

Customizing Mass Housing:  
A Discursive Grammar for Siza's Malagueira Houses

by

José Pinto Duarte

Bachelor in Architecture, 1987  
Lisbon Technical University, Portugal

Master of Science in Architecture, 1993  
Massachusetts Institute of Technology

Submitted to the Department of Architecture in  
Partial Fulfillment of the Requirements for the Degree of  
Doctor of Philosophy in Architecture: Design and Computation

at the

Massachusetts Institute of Technology

September 2001

© 2001 José Pinto Duarte. All rights reserved.

The author hereby grants to MIT permission to reproduce and to distribute publicly paper and  
electronic copies of this thesis document in whole or in part.

Signature of Author.....

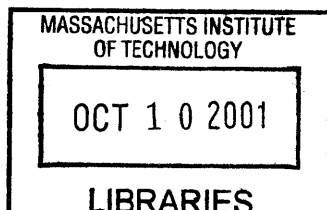
Department of Architecture  
August 13, 2001

Certified by.....

William J. Mitchell  
Professor of Architecture and Media Arts and Science  
Dean, School of Architecture and Planning  
Thesis Supervisor

Accepted by.....

Stanford Anderson  
Chairman, Committee on Graduate Students  
Head, Department of Architecture



ROTC





## **Dissertation Committee**

Terry Knight  
Associate Professor of Design and Computation

William J. Mitchell  
Professor of Architecture and Media Arts and Science

George Stiny  
Professor of Design and Computation

Patrick Winston  
Ford Professor of Artificial Intelligence and Computer Science



CUSTOMIZING MASS HOUSING:  
A DISCURSIVE GRAMMAR FOR SIZA'S MALAGUEIRA HOUSES

by

José Pinto Duarte

Submitted to the Department of Architecture  
on August 13, 2001 in partial fulfillment of the  
requirements for the Degree of Doctor of Philosophy  
in Architecture: Design and Computation

ABSTRACT

This thesis proposes a process of providing mass-customized housing based on computer-aided design and production systems. It focuses on the design part, which mainly consists of an *interactive system* for the generation of design solutions based on a mathematical model called *discursive grammar*. A discursive grammar includes a shape grammar, a description grammar, and a set of heuristics. The shape grammar provides the rules of formal composition, whereas the description grammar describes the design from other relevant viewpoints. The set of heuristics is used to guide the generation of designs by comparing the description of the evolving design with the description of the desired house. The generation of a design proceeds first by producing a design brief from the user-prompted requirements and then by finding a solution that satisfies this brief. Search is largely deterministic, which decreases the amount of time required to find a solution, thereby making it reasonable to develop Web-based implementations. The proposed model enables an enduring designer's dream, that of the mass customization of housing.

The model is illustrated with a case study that includes a shape grammar developed for the houses designed by the architect Álvaro Siza at Malagueira, a description grammar based on the Portuguese housing regulations, and a set of heuristics inferred after a set of experiments. In these experiments, designers were asked to generate houses based on the Malagueira grammar for specific clients. It is argued that this discursive grammar provides a rigorous method for understanding and teaching Siza's design process and that similar grammars could be developed for other styles. A Web page for explaining the grammar and generating new designs on-line was developed as a prototype.

Thesis Supervisor: William J. Mitchell

Title: Professor of Media Arts and Science, Dean, School of Architecture and Planning



## Acknowledgements

I am deeply thankful to my advisors Terry Knight, William J. Mitchell, George Stiny and Patrick Winston for their support, guidance, and friendship. This thesis owes much to Terry, Bill, and George's work on shape grammars. Patrick provided me with complementary viewpoints from artificial intelligence.

Thank you Álvaro Siza for his support and enthusiasm and for giving me unrestricted access to his archives.

Thank you António Reis Cabrita and João Bento who gave me the conditions to develop part of this research first at the *Laboratório Nacional de Engenharia Civil (LNEC)*, and then at the *Instituto Superior Técnico (IST)* in Lisbon. João also was helpful in guiding me throughout my course of study. I thank Teresa Heitor for understanding my urge to finish my writing, and for helping me to make a smooth transition to IST.

I thank William Porter and Edith Ackerman for providing me with the opportunity to undertake and discuss part of the experiments described in Chapter 6 within the context of the MIT Design Research Seminar. I also am thankful to Ryusuke Naka, from Miyagi University, for creating the opportunity to develop the remaining experiments. Thank you also to the students who volunteered to participate in both experiments.

I thank João Pedro for sharing his work on the Portuguese housing guidelines, used as a basis for the development of the programming grammar described in Chapter 7. I also thank the valuable support of Ana Runa, Filipe Santana, and Sara Eloy, who assisted in the development of the Web site described in Chapter 8.

This work had the financial support of *Fundação para A Ciência e a Tecnologia*, the Luso-American Foundation, the Calouste Gulbenkian Foundation, Portugal; and the Massachusetts Institute of Technology, U.S.A. It was partially developed within the scope of the Praxis XXI project Praxis/2/2.1/CSH/844/95 with the institutional support of LNEC.

I want to thank my friends, Anne, Susan, Stephen, Paul, Birgul, Andrew, João, and Luisa for contributing to make my stay at MIT so pleasant, and for many interesting discussions.

I dedicate this thesis to my parents and my wife.



# Table of contents

1. Introduction .....	13
1.1 Preview: problem and solution .....	13
1.2 Areas of study .....	18
1.2.1 From performance criteria to a description grammar .....	18
1.2.2 From typology to shape grammar .....	19
1.2.3 From optimization to deterministic heuristic search .....	20
1.2.4 Discursive grammar: simulating, generating, and "optimizing" .....	25
1.3 From mass production to mass customization .....	26
1.4 Siza's Malagueira .....	28
1.5 Contributions .....	31
1.6 Organization of the thesis .....	33
References .....	34
2. Methodology .....	37
2.1 Concepts .....	37
2.2 Preparatory phase .....	41
2.3 Descriptive phase .....	42
2.4 Analytic Phase .....	43
2.5 Synthetic Phase .....	44
2.6 Goal Phase .....	45
2.7 Discursive Phase .....	48
2.8 Implementation Phase .....	49
2.9 Summary .....	49
References .....	50
3. Precedents .....	51
3.1 Introduction .....	51
3.2 Performance criteria .....	51
3.2.1 Programming method .....	53
3.2.2 Evaluation method .....	58
3.3 Grammars .....	61
3.3.1 Shape grammars .....	61
3.3.2 Parametric shape grammars .....	61
3.3.3 Shape grammars: analytical and original .....	63
3.3.4 Computer implementation .....	64
3.3.5 Proposed grammar: analytical and original .....	65
3.3.6 Description grammars .....	66
3.3.7 Parallel grammars .....	66
3.4 Conclusion .....	67
References .....	67
4. Corpus .....	71
4.1 Introduction .....	71
4.2 Plan .....	71
4.2.1 Context .....	71
4.2.2 The plan .....	74
4.2.3 Housing tissue .....	85
4.2.4 Promotion .....	92
4.2.5 Construction phases .....	92
4.3 Housing types .....	95
4.3.1 Design schemes .....	95
4.3.2 Structural and building systems .....	132

4.4 Summary.....	135
References .....	136
5. The Malagueira Shape Grammar .....	137
5.1 Introduction.....	137
5.2 Algebras .....	137
5.3 Parallel grammars .....	139
5.4 Context .....	143
5.5 Composition: dissecting rectangles.....	144
5.6 Function.....	148
5.6.1 First floor functional organization .....	148
5.6.2 Articulation between floors .....	152
5.7 Structure.....	153
5.8 The universe of design solutions.....	154
5.8.1 Designing the grammar.....	154
5.8.2 Estimating the number of solutions .....	156
References .....	167
5.9 Rules .....	167
5.9.1 Stage 0: Introduce initial shape.....	167
5.9.2 Stage 1: Define the first floor.....	168
5.9.3 Stage 2: Define the second floor.....	175
5.9.4 Stage 3: Define the terrace .....	177
5.10 Derivation of an existing design.....	216
5.11 Summary.....	233
References .....	233
6. Experiments.....	235
6.1 Introduction .....	235
6.2 Experiment 1: existing design (analytic test) .....	235
6.2.1 Goal.....	235
6.2.2 Subjects.....	236
6.2.3 Setting .....	236
6.2.4 Task.....	236
6.2.5 Procedure .....	236
6.2.6 Results.....	237
6.2.7 Discussion and conclusions .....	237
6.3 Experiment 2: random design (synthetic test) .....	256
6.3.1 Goal.....	256
6.3.2 Subjects.....	256
6.3.3 Setting .....	256
6.3.4 Task.....	256
6.3.5 Procedure .....	257
6.3.6 Results.....	257
6.3.7 Discussion and conclusion.....	257
6.4 Experiment 3: goal-oriented design I (goal test) .....	268
6.4.1 Goal.....	268
6.4.2 Subjects.....	268
6.4.3 Setting .....	269
6.4.4 Task.....	270
6.4.5 Procedure .....	270
6.4.6 Results.....	271
6.4.7 Discussion .....	281
6.5 Experiment 4: goal-oriented design II (goal test) / collaborative design.....	303
6.5.2 Subjects.....	304
6.5.3 Setting .....	304
6.5.4 Tasks.....	305



6.5.5 Procedure .....	307
6.5.6 Results.....	308
6.5.7 Discussion .....	323
6.6 Summary.....	338
References .....	340
<b>7. Discursive grammar.....</b>	<b>341</b>
7.1 Introduction .....	341
7.2 Definitions .....	342
7.2.1 Technical definition.....	343
7.2.2 Operative definition .....	343
7.2.3 Mathematical definition.....	344
7.3 The PAHPA-Malagueira discursive grammar .....	345
7.4 The PAHPA programming grammar.....	347
7.4.1 Description (features): constraints, quality, and cost .....	348
7.4.2 User/programmer interface .....	353
7.4.3 Dependency among features .....	353
7.4.4 Rules .....	356
7.5 The revised Malagueira grammar.....	440
7.5.1 Algebras and parallel grammars: viewpoints and features .....	441
7.5.2 Rule types .....	443
7.5.3 Rules .....	444
7.6 Summary.....	507
References .....	507
<b>8. Implementation .....</b>	<b>509</b>
8.1 Introduction .....	509
8.2 The interpreter.....	509
8.2 The Web site.....	514
8.4 The envisioned housing design framework .....	521
8.5 Summary.....	525
References .....	525
<b>9. Conclusion .....</b>	<b>527</b>
9.1 Summary.....	527
9.2 Contributions.....	528
9.2.1 Findings.....	528
9.2.2 Minor contributions .....	530
9.2.3 Major contributions .....	531
9.3 Future work .....	533
9.3.1 Improvements.....	533
9.3.2 Major steps.....	534



# 1. Introduction

## 1.1 Preview: problem and solution

The ultimate goal of this work is a computational framework for the design of mass-customized houses that includes a design and a production system. The current focus is on the development of the design system. The purpose of mass-customization is to provide high-quality housing at an affordable cost. The definition of quality is threefold. First, it implies the satisfaction of functional requirements defined by building regulations. Second, it requires the satisfaction of aesthetic requirements established within a particular design style. This style can be historical or based on the work of an existing architect. Third, it aims at satisfying requirements, which can be functional, aesthetic, or cost-based, and are specified by the client in addition to the other two. In summary, quality is defined as the satisfaction of the user needs. A high degree of customization leads to high user-satisfaction and prevents costs associated with post-construction changes. Cost-control also is guaranteed with recourse to production techniques that rely not on exhaustive repetition, as in traditional mass-production, but rather on computer-aided manufacturing processes.

Traditionally, when a designer is faced with the design of a large development, the usual solution is to design a limited number of housetypes and then to repeat them based on market analysis. The reason for such a procedure is twofold. First, the designer is not capable of designing each house individually due the large amount of information that would be required to process. Second, traditional manufacturing techniques require

repetition to lower the costs using economies of scale. The envisaged process aims at overcoming such limitations by using computer-aided design and manufacturing processes. The idea is to give mass-produced houses some of the qualities associated with individually designed homes. The use of such a process gives the average client access to the work of highly skilled architects, thereby making architecture more democratic. The biggest market is not the traditional custom home client, but the majority of clients who do not use architects anyway.

The framework consists of computer-aided design and production systems (Fig. 1.1). The design system includes an interactive program for generating housing solutions (A), rapid prototyping and virtual reality techniques for visualizing these solutions (B), and computer-aided production to materialize them (C). The user accesses the program on the Web. The program guides the user through questions that an architect would normally ask during an initial meeting, such as the family members' profile, their living habits, the rooms they want, the cost that they can afford, and so on, to gather enough site (1) and user data (2). When the interview is over, the program generates the design brief or housing program (4), taking into account existing housing regulations (3). The user can then make changes to the initial requirements and the program will update the design brief (5). Once the brief is approved (6), the program generates a housing solution that satisfies the requirements (8) within a given design language (7). The solution takes the form of a 3D digital model. This model can be visualized on a flat computer screen using simple 3D viewers, or taken into a sophisticated, virtual reality environment (9) in which the user can walk-through the house (10). Alternatively, the model can be used to produce a scaled physical model (12) using rapid prototyping techniques (11). After assessment and visualization (13), the user might want to change the initial requirements and proceed through another iteration of the design process (14).

Once a solution is accepted (15), an order can be automatically issued to the housing factory. This order will include a detailed list of parts, and digital information to manufacture the parts (17) using computer-aided manufacturing techniques (16). At the end of the manufacturing process, these parts are transported to the site and assembled (18). The house is finally ready to inhabit (19).

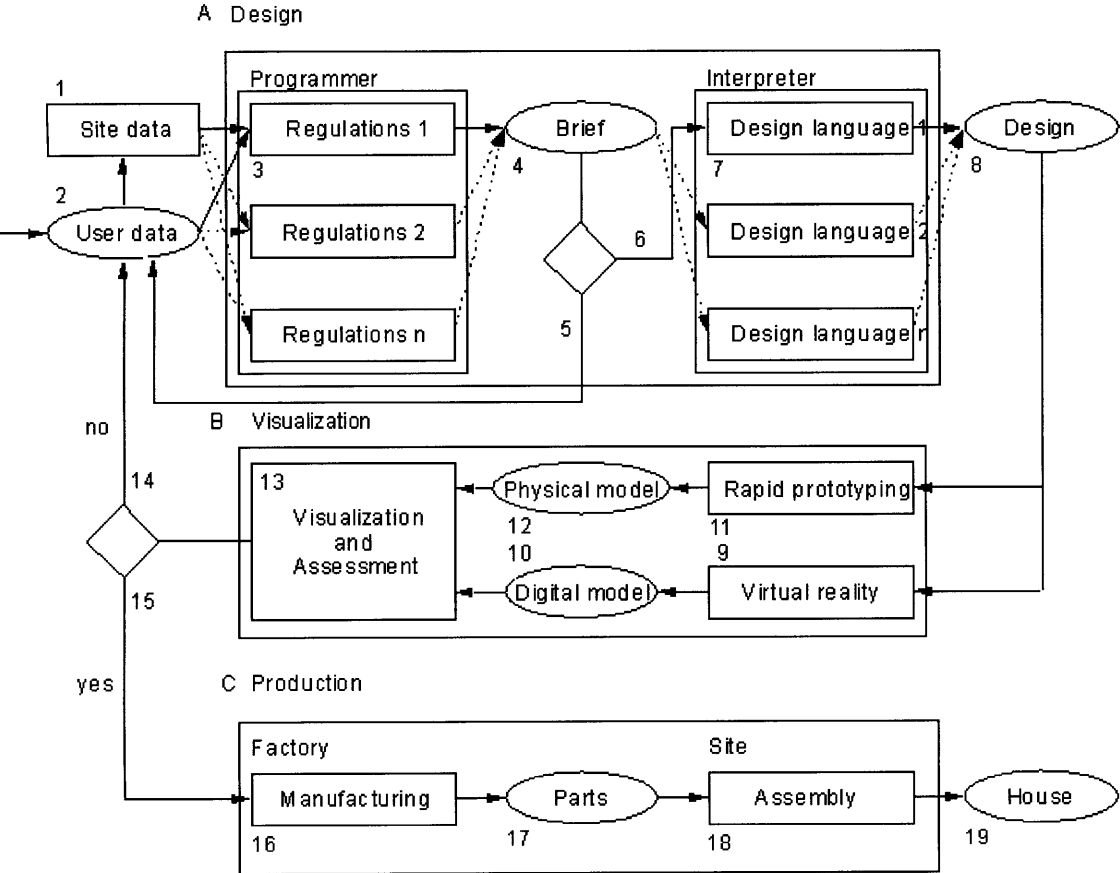
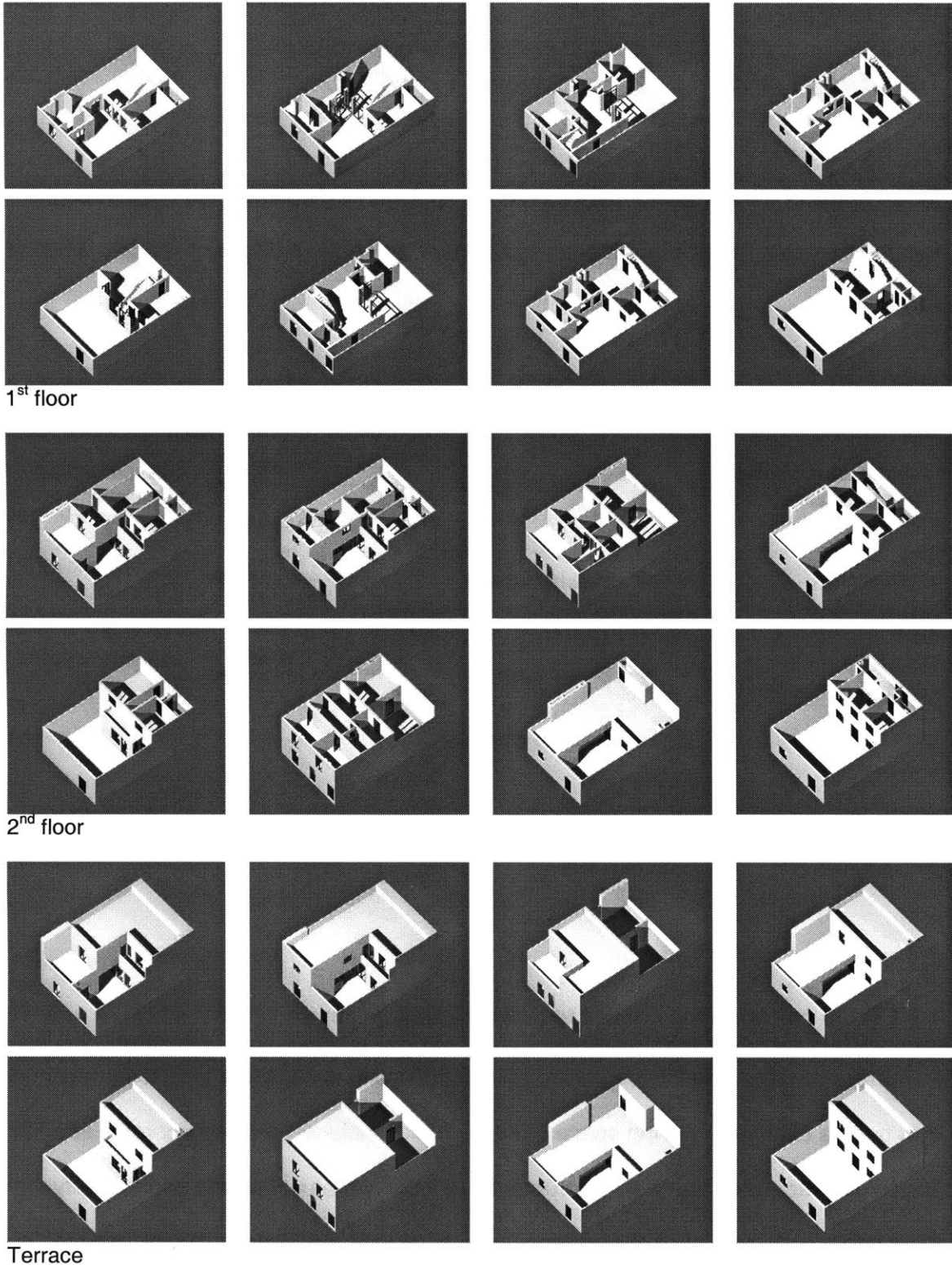


Figure 1.1 - Diagram of the framework envisioned for the mass customization of housing

This work aims at the development of a mathematical model for the interactive program just described. This program can be used by the designer or by the client, but in both cases, it enhances the designer's creativity by rapidly providing alternative design solutions. Such model needs to overcome three problems. These problems correspond to the three different types of computer-based methods identified by Radford and Gero

(1988). First, it needs to provide a way of translating client data into design requirements, and to verify whether a design satisfies these requirements -- *the simulation problem*. In simulation, the computer manipulates a mathematical model that describes the design to evaluate the performance of a given design configuration against the design requirements. Second, it has to codify the rules of formal composition to design a house in a given style -- *the generation problem*. In generation, the computer is used to produce design configurations according to a set of rules. And third, it needs a mechanism to translate the design requirements into a housing solution -- *the optimization problem*. In optimization, the computer is used to generate design configurations that meet a performance goal. The use of the term optimization to refer to design is controversial, but it is possible to overcome the controversy, as it will be explained in Section 1.2.3. The number of solutions that satisfy multiple requirements is potentially very large. Thus, an important part of the model is a computational strategy capable of searching a potentially large design space, and providing insight into function-form relations for multi-criteria housing design. In the next section, we will see how different areas of study are merged to overcome the simulation, generation, and optimization problems and create a model with the desired features, called discursive grammar.

The model is illustrated with a case study that includes a shape grammar developed for the houses designed by the architect Álvaro Siza at Malagueira, a description grammar based on the Portuguese housing guidelines, and a set of heuristics inferred after a set of experiments. In these experiments, the designers were asked to generate houses based on the Malagueira grammar for specific clients. One of the designs was placed among Siza's designs and shown to Siza who did not distinguish it from his own designs thereby validating the grammar. (Figure 1.2)



1<sup>st</sup> floor

2<sup>nd</sup> floor

Terrace

Figure 1.2 – The model for the design system of the framework proposed for customizing mass housing is illustrated with a discursive grammar for Siza’s Malagueira houses. A design by the author of the grammar after its rules was shown to Siza amidst several of his own designs. Siza did not distinguish the new design from his own. Do you? (See solution on page 19.)

## 1.2 Areas of study

Three areas of study -- performance criteria, typology, and optimization -- are brought together in this work by a fourth one -- grammars, as shown in Figure 1.3 and explained below.

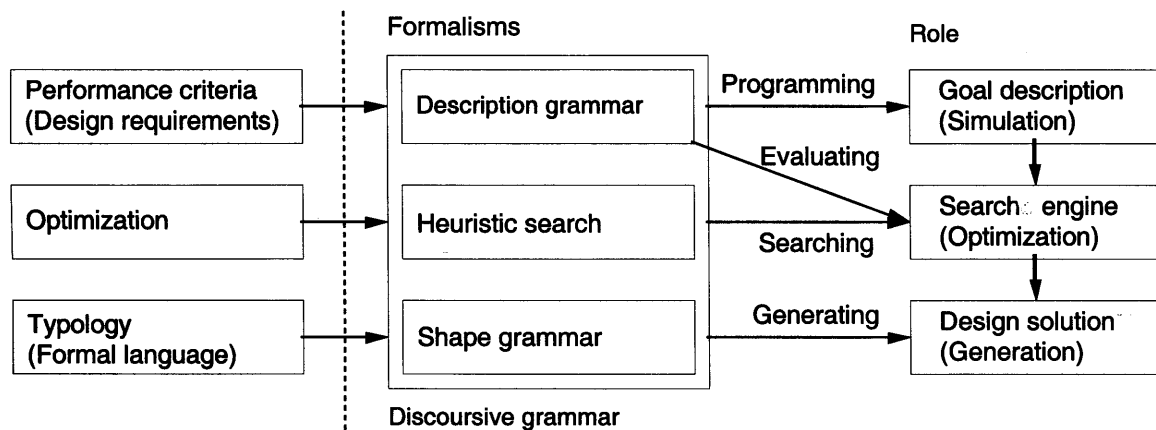


Figure 1.3 - Merging multiple areas of study

### 1.2.1 From performance criteria to a description grammar

*Performance criteria* had its beginning in the 1960s with studies that tried to understand how people used the dwelling space. These studies included anthropomorphic and sociological analyses that were instrumental to identify the user needs and to codify them into a coherent set of design requirements (Portas and Gomes 1964a, 1964b). A practical result of these studies was the development of design guidelines that helped the designer to establish the housing program, such as the IPHPE (MHOP e LNEC



1978), the RTHS (MES 1984), and the NTPEH (Duarte e Paiva 1994). Such studies then led to the development of quality evaluation studies to determine whether existing dwellings satisfied design requirements (Portas 1969, Cabrita 1987, Coelho 1993). The outcome of these studies was the development of several quality evaluation methods to measure the performance of dwellings and designs against requirements such as the SEL (Aellen et al 1979) and QUALITEL (Association Qualitel 1989) methods. The field of study concerned with the definition and evaluation of performance became known as performance criteria. Recently, Pedro summarized the previous studies into a housing programming and evaluation method adjusted to the contemporary Portuguese reality (Pedro 2000). The current study encodes a modified version of this method into a coherent set of rules forming a description grammar (Stiny 1981). This grammar has a dual role. First, it transforms the user data into the set of design requirements that form the design brief or housing program. The brief takes the form of a goal description. Second, it provides a way of evaluating the evolving design by comparing its description with the goal description. Therefore, the description grammar solves the simulation problem, the first problem mentioned at the end of Section 1.1.

### **1.2.2 From typology to shape grammar**

Typology<sup>2</sup> also was a consequence of the 1960s effort to understand the use of space. To study the variety of dwelling forms it was necessary a classification into categories. This required the definition of classification criteria, then to group the dwellings into categories with the same features according to such criteria, called typologies, and

---

<sup>1</sup> Figure 1.2: the design by the author of the grammar is the second on the second row.

finally, to illustrate these groups with concrete examples, called types (Blachère 1972, Vidler 1977, Habraken 1988). To define the classification criteria, however, was problematic. First, it was necessary to define the viewpoint, which could be formal (e.g. number of floors: single-floor), structural (e.g. building material: adobe), social (e.g. a clear separation between the private and the public quarters), or stylistic (e.g. Art Deco). Then, within each viewpoint, there could be different criteria (e.g. number of floors and type of covering: “single-floor with roof”). The result was an extensive list of descriptions, often contradictory, that made classification difficult and did not say how to design new members of the typology. For this reason, the illustrative type was often taken too literally, and used as a model. Shape grammars (Stiny and Gips 1972) overcame these problems by merging the different viewpoints into a set of instructions that specify how to generate new instances of the typology. Therefore, shape grammar is the formalism used in the proposed design system for constructing new designs, thereby solving the generation problem, the second problem mentioned at the end of Section 1.1.

### **1.2.3 From optimization to deterministic heuristic search**

Once, we have a goal description and a process to generate solutions, the problem becomes one of finding the solution whose description most closely matches the goal. This can be viewed as an optimization problem. Viewing design as optimization is controversial (Radford and Gero 1988), but part of the controversy is because optimization has two meanings. In the general meaning, it means to improve a solution

---

<sup>2</sup> In the literature, the term typology has a double meaning. It might be used to refer to the studies concerned with the classification of buildings into categories, or to refer to the categories themselves.

to a problem. In the technical meaning, it refers to a set of techniques used in operations research. Authors who think that optimization is not a good paradigm for design usually indicate four difficulties. The first is the subjectivity of some criteria used in the generation and evaluation of designs. An example of such criteria is the symbolic value of buildings. The second is the difficulty to model aspects of design that are difficult to abstract, such as the society's pressure. The third difficulty is that the design problem is often ill-defined, that is, the user might not be completely aware of what the needs are. The fourth difficulty is that the problem might be over-constrained, as there might be conflicting requirements. The first two difficulties are common to the modeling of complex problems in general. In these cases, the model is an abstraction and it can be accepted as long as it proves useful. Therefore, there is no reason why design problems should be treated differently. Once the model is developed and implemented, it will be possible to test its usefulness. The ill-definition difficulty is overcome in the proposed model by including rules to provide default values for data not supplied by the user. Such values are based on the typological studies referred to in Section 1.2.2. The user can then change these values, to increase customization. The over-constrained problem is solved by including rules to deconstrain the problem. In practice, this corresponds to finding the feasible solution that is closest to the goal. Similar procedures are already used by existing optimization techniques. In conclusion, it is possible to represent the design of customized housing as an optimization problem.

The problem, however, is complex. There are many tasks involved in the design of a house (e.g. layout, openings), and there is a multitude of criteria to satisfy (e.g. space, topology, comfort). *Task complexity* is tackled by decomposing the problem into smaller problems, then concentrating on essential features of these sub-problems, and finally, selecting an appropriate technique to solve each of them. For instance, there is a

separation between the generation of the layout, and the design of the opening system. These procedures, known in AI as sub-goaling, abstraction, and means-end analysis, mean that solution optimality cannot be guaranteed (Winston 1993). Even if the different steps are optimal locally, there is no guarantee that their combination will be globally optimal. *Criteria complexity* is tackled by representing the problem as an optimization problem and then by turning the multicriteria problem into a single criterion one. There are two basic forms of optimization methods: single criterion and multi-criterion. In single-criterion, there is only one viewpoint that is being optimized, whereas in multi-criteria there are several. The solution to a multi-criteria problem is a set of solutions with performances such that no other solution exists that will yield an improvement in one criterion without degrading at least one other criterion. These solutions are known as Pareto optimal. In multicriteria, choice implies tradeoffs, that is the amount of quality in one criterion that the user can give up in order to gain more of another. Such a choice requires the user to rank or to assign weights to the criteria. Ranking is a special, less specific case of weighting. There are three-basic types of approaches to solve multi-criteria problems: non-preference methods, preference methods, and interactive methods.

The *preference approach* requires the user to assign weights to each criterion a priori. It solves the problem by turning it into a multicriteria problem whose fitness function is the weighted sum of the fitness functions for the single criterion. The result is a solution that represents the best compromise according to these weights. The *non-preference approach* generates several of the Pareto solutions, enough to give an indication of the tradeoffs involved. It works by varying the weights assigned to each criterion and generating the corresponding solutions. Thus, in this approach the problem also is transformed into a single criterion problem. Different non-preference methods differ in

the strategy used to vary the weights so that the set of generated solutions is representative of the Pareto set.

The *interactive approach* combines the processing power of the computer with the assessing abilities of the user, through interaction between the two. There are two kinds of interactive methods: interactive search, and interactive choice. In *interactive search*, the user sets the weights a priori, as in the preference method, and the computer generates the corresponding solution. Then the user changes the weights and the computer generates the new solution. The process is repeated until the user is satisfied with a solution. In *interactive choice*, the computer generates a set of Pareto solutions, as in the non-preference method, and then the user imposes restrictions on the performances. Then computer generates a new set of Pareto solutions. The process is repeated until choice is narrowed down to a single solution. The method used in the proposed model is an interactive search method, mainly because the lesser burden on the computer implementation was considered more appropriate, for a Web-based implementation, in which response time is crucial. The drawback is the greater burden on the user to articulate the preferences.

Once complexity is reduced, the problem becomes a simple search problem. Search is an important topic in artificial intelligence (AI) that emerged with the first AI computer programs in 1950s. There are two opposite approaches to AI, called weak and strong AI. The first defends the possibility of developing general problem-solving programs, whereas the second advocates the development of programs with intensive domain knowledge. The first AI researchers believed in the weak approach, but a move towards intensive knowledge-domain has characterized the field since then. Search is a very

general problem-solving technique that tends to be close to the knowledge-poor end of the spectrum, but its exact position depends on the type of search method (Korf 1995).

There are different kinds of search, blind, heuristic, optimal, and stochastic search. Blind search proceeds without any assessment of the intermediate states relatively to the goal. They are knowledge-poor algorithms and their major drawback is the time that they might take to find a solution, especially if the width and depth of the search tree are very large, as in the design of a house. In heuristic search, a function is used to estimate the value or distance of intermediate states relatively to the goal. The conversion of the problem from multicriteria into single criterion yields a single heuristic function that might be used in heuristic search. Thus, some domain knowledge is used to decrease search time. The disadvantage of heuristic methods, however, is that search might get stuck in a local maximum, if the search space contains peaks, ridges, or plateaus. Optimal search algorithms take into account not only the distance to the goal of a given intermediate state, but also the distance traversed so far. They are, thus, able to find the best path to the goal. Stochastic algorithms try to avoid getting stuck in local maxima by giving intermediate states with lower performances some probability of being chosen for further development. The drawback is that such algorithms are non-deterministic and might yield different results in different runs with the same criteria. Therefore, they do not guarantee that a global maximum is found, and it might take a long time before the algorithms converge to a solution.

On the opposite side of search methods are pure grammars. If there is enough knowledge of the domain, it is possible to know exactly which rule to choose at each step. This process is deterministic and fast. The problem, however, is that it might be difficult to acquire such domain knowledge, without generating all the possible solutions.

Generating all the possible solutions might take too long or be unfeasible if their number is very high, as in the Malagueira grammar, our case study. For this reason, experiments with designers using the grammar were undertaken to find the heuristics used to choose rules leading to good solutions. The experiments permitted to find some of such heuristics, but these were not enough to develop a pure grammar. Therefore, it was necessary to couple the grammar with a search mechanism.

In summary, a choice between developing a pure grammar or using search depends on the number of solutions and it involves a tradeoff between the time required to acquire enough domain knowledge to develop a pure grammar, and the time needed to find a solution running a grammar with search.

#### **1.2.4 Discursive grammar: simulating, generating, and "optimizing"**

In Section 1.1, it was outlined three problems -- simulation, generation, and optimization -- that needed to be solved in the development of a design system for customizing mass housing, and in sections 1.2.1 through 1.2.3 it was shown how it could be effectively done. The simulation and generating problems can be solved with the use of a description grammar (Stiny, 1981) and a shape grammar (Stiny and Gips, 1972), respectively. A grammar consists of a set of substitution rules that apply recursively to an initial assertion to produce a final statement. In description grammars, the assertions are symbolic descriptions, whereas in shape grammars, they consist of shape descriptions. In addition, description grammars deal with semantics, and shape grammars address form. The third constraint is satisfied with a set of heuristics. Heuristics are used to choose a rule for application at each step of the design generation

or to constraint choice to a small number of rules. Other heuristics assess the designs that would result from the application of each of the available rules, and then choose the one that takes the evolving design closer to the goal description in the design brief. This process is deterministic. At a micro-scale, a specific design context will lead to the application of a specific rule; at a macro-scale, a given context will lead to a given housing solution. I call this mathematical model a discursive grammar because it allows the generation of formally and semantically correct designs. Each house is like a piece of speech in the language that is appropriate for the context. In the following section, it will be shown how the approach proposed in this study fits into the context of approaches to mass housing.

### **1.3 From mass production to mass customization**

Three great technological revolutions have changed the course of human history: the agricultural, industrial, and computer revolutions. For Toffler (1984), these revolutions are like waves of change that spread across space and time. The impacts of the first two revolutions in architecture are well known, but the impact of the third one, which started in the 1950s, is still being acknowledged. In fact, the impacts of these waves of change seem to take some time to reach architecture, which consequently often lags behind other fields.

The framework for the customization of mass housing proposed in this work represents an effort to take advantage of the benefits brought by the computer revolution to solve the problem of customizing mass housing. The design of mass housing has been an important theme in architecture since industrial revolution in the 19th century caused a



rural exodus towards the cities. Not surprisingly, an important part of architectural production in the 20th century was focused on this design problem and it is possible to identify three different approaches.

In the first half of the century, designers attempted to solve the problem by introducing in architecture a production process based on the assembly line. The assembly line was initially developed for the automotive industry by Henry Ford, but soon became a paradigm for the whole industry. It enforced the production of standard parts and identical products by a single company. This approach was extensively used in the reconstruction of Europe after World War II. In Eastern Europe, centralized governments and egalitarian societies made it easier to introduce and accept until the fall of the Berlin wall. In Western Europe, once the housing shortage caused by the war was overcome, the implied degree of repetition became unacceptable by a society increasingly focused on individual freedom and choice. Consequently, this approach was progressively abandoned.

In the United States, market forces led to the development of a new paradigm for mass housing in the 1960s, called the kit-of-parts. In the previous approach, there was no interchangeability of parts. The new approach was an open process that required heavy coordination to integrate standard parts from different companies. Production costs were reduced because each company, being focused on the production of a single part, could optimize the process. Nevertheless, difficulties in communication and in guaranteeing perfect interchangeability of parts constituted barriers to integration and cost-reduction.

None of the industrial approaches referred to above were able to solve the housing problems, especially in those parts of the World with incipient degrees of industrialization. Therefore, there was a progressive shift towards a new approach concerned with the human and social aspects of housing and their impact on design. Eventually, this approach became mainstream in the industrialized countries as well. In Portugal, it found a favorable ground, leading to the development of the theoretical studies mentioned in Section 1.2.1, and then to design proposals that attempted to put them into practice. The Malagueira development by Álvaro Siza, mentioned in Section 1.4, and used as case study in this work, is an example of such proposals.

In the meanwhile, the development of the computer revolution has already prompted the shift towards mass customization. In this new industrial model, the assembly line creates thousands of variations of the same product, each one different. There is a shift from the current focus of one-size fits all to a new focus of a customized product. This is already happening in the computer and clothing industry. The housing industry has been slow to adopt a similar model, although it has long been proposed (Duarte 1989, 1995). This work attempts to give a step towards that model.

#### **1.4 Siza's Malagueira**

Álvaro Siza (1933-) is one of the most influential contemporary architects. His work has been the focus of numerous studies, but there has not been any analysis that tries to understand in depth Siza's work at Malagueira (1977-). Perhaps no other work of Siza is more conceptually meaningful in the context of contemporary architecture than the Malagueira housing development. The neglect of this project is somewhat surprising if

one considers that an important part of architectural practice and theory in the 20th century was concerned with the design of mass housing.

All the great masters of the 20<sup>th</sup> century have addressed this design problem in their work. To recall an important few one can mention Walter Gropius' housing development at Törten (1926-1928), Le Corbusier's Domino Houses at Pessac (1926-1929), and Frank Lloyd Wright's Usonian houses (1946-1954). What is common among these projects is the desire to devise a scheme that could be used to generate affordable mass housing, using industrialization as the means for lowering the costs. Gropius used repetition at its extreme. He designed three housetypes for a total of 316 dwellings (Bonelli et al 1983). Le Corbusier, despite the use of repetition, was concerned with variation. He designed four types for a projected 200 dwellings, although only 51 were built. His concern was essentially placed at the urban scale, as he did not foresee variations within each type. Yet, "if we compare the various interiors evolved by the occupants with Le Corbusier's original design it is immediately apparent that his conception lent itself to subsequent modification." (Boudon, 1979) Wright was probably the most concerned with adjusting the design to the households in line with his "concept of houses being as different as their owners" (Sergeant, 1976) and this was clearly expressed in the design of 47 different homes at Usonia.

Siza's Malagueira is placed at the core of this discussion about housing and it represents a logical development of the previous approaches. This development has its roots in the experiments undertaken in Portugal after the 1974 revolution under the Ambulatory Support to Local Residents program<sup>3</sup>, known as SAAL. The program had as

---

<sup>3</sup> *Serviço Ambulatório de Apoio Local* in Portuguese

one of its desired outcomes the direct participation of future dwellers in the design of their homes. In the spirit of the program, designers were expected to work with the future dwellers in order to produce customized dwellings. At Malagueira, as in many other SAAL developments, cooperatives of future dwellers were responsible for promoting the development. The designer was supposed to meet with an assembly of cooperants to discuss housing types, and then with each individual household to customize its house. Later, assemblies and individual meetings became less frequent because they were time-consuming. A similar phenomenon happened at Malagueira.

Malagueira was planned as an extension of the city of Évora and it is a large development that encompasses 1,200 dwellings. Although the first house was designed in 1977 and built in 1978, design and construction still proceeds today. Siza devised a scheme that allowed for the generation of different houses. In fact, over 35 different layouts were designed, ranging from one bedroom to five bedroom houses. He used this scheme to incorporate into the design process the users desire for a unique house. The scheme was composed of a set of design rules that were used by Siza or his collaborators to design customized houses. However, despite the potential of Siza's design system, three limitations could be identified. First, it was difficult to convey the rules to other designers because they were never laid down in an explicit way. Second, there were obvious difficulties in representing the universe of solutions using traditional design media and thus difficulties in conveying them to prospective dwellers. Third, the system's potential to customize the dwellings was not fully used, despite the ability for generating diverse designs.

This study is an attempt to overcome such limitations and it is based on three arguments: (1) shape grammars can provide the technical apparatus to make Siza's

design rules at Malagueira explicit; (2) a computer program encoding the grammar would allow one to use Siza's design system more effectively; and (3) shape grammars and computer programs, coupled with rapid prototyping and virtual reality techniques, can provide a digital framework for customizing the design of mass housing. With such a framework, designers could work with dwellers in the design of their houses and reestablish the dialogue envisaged by the SAAL program.

## 1.5 Contributions

The dissertation makes four major contributions:

1. The outline of a system for the mass customization of housing. The provision of housing still follows a process in which the prospective household has to choose a house among pre-defined types. The opportunity for customization with such a process is very little. The envisaged process attempts at taking advantage of new technologies to improve the degree of customization. This system will include a design system, which will be elaborated in the dissertation, and a production system, which will be the subject of future research. The design system encompasses a Web-based interactive system for the exploration of design solutions, and virtual reality and rapid prototyping techniques for visualization purposes. The production system will include computer aided manufacturing and assembly processes.
2. A mathematical model for the interactive design system. Traditional shape grammars permit the random generation of design solutions within the languages they codify. These designs are syntactically correct but there is no way to guarantee

that they match given requirements, that is, that they are semantically correct too. Existing approaches to this problem propose the use of stochastic processes, which use trial and error to guide the generation of designs towards the goal. Such processes might take several hours before finding a solution and, therefore, they are not suitable for web-based implementations intended for use by prospective clients. The proposed model takes a deterministic approach as a way of reducing the generation time. Such model is an extension of the grammar formalism and it includes a shape grammar, a description grammar, and a set of heuristics that allows the generation of a design solution that matches requirements given a priori. This model is called a discursive grammar.

3. The prototype of a Web page for explaining the grammar in a visually understandable way. Traditional documents on shape grammars tend to be very technical and mathematically oriented texts. Such features make it difficult for non-technically oriented designers to use grammars in practice thereby preventing a greater interest in the development of further research. An interactive Web page in which the process of developing and using the grammar is described using interactive gadgets is, therefore, proposed as a way to overcome such difficulties. The Web page is a major part of the mass customization system mentioned above and it includes a catalog of existing designs to be browsed by prospective clients, a tool for teaching the designers how to design houses in the language defined by the grammar and, ultimately, a mechanism for generating new houses on line.
  
4. A rigorous method for understanding and teaching Siza's style at Malagueira. Traditional studies on the history of architecture tend to describe a particular style without saying how to actually generate new designs in the style. The dissertation

uses as a case study the Malagueira houses designed by Álvaro Siza and a grammar is proposed for this style. The study goes one step beyond previous grammars as it uses the designer as a source of information and proposes a model for the designer's decision-making process while designing. By describing how to generate designs in this style with mathematical rigor, the study gives one step towards the explanation of architectural qualities in a rigorous way. Similar methods might be applied to other styles.

## **1.6 Organization of the thesis**

The dissertation is organized into nine chapters, including this introductory chapter.

Chapter 2 presents the methodology used in the development of the discursive grammar, which included the development of a shape grammar followed by a set of experiments.

Chapter 3 describes precedent studies in performance criteria and grammars, explaining how the proposed discursive grammar fits in the context.

Chapter 4 describes the corpus of Malagueira houses, the main source of information utilized for developing the shape grammar, using photos, drawings, and digital models.

Chapter 5 presents the initial shape grammar developed for the Malagueira houses. First, it explains the structure of the grammar, and then it explains the rules. The generation of a house in the corpus is included to illustrate rule application.

Chapter 6 describes the set of four experiments undertaken to test the grammar. The first experiment checked whether the grammar could account for a new house designed by Siza after the grammar had been finished. The second experiment was the generation of a random new house by the author of the grammar. The third and fourth experiments consisted of the generation of new houses for specific clients by other designers.

Chapter 7 introduces the discursive grammar. After describing its structure, including the programming and the designing grammar, it explains how the rules are different from the initial grammar, and how the different components interact to guide design generation towards a solution that meets the goal.

Chapter 8 describes three steps towards the implementation of the proposed framework, including a computer implementation of the discursive grammar, a Web site that permits the generation of new designs on-line, and the combined use of rapid prototyping and virtual reality to visualize such designs.

Chapter 9 summarizes the conclusions and outlines paths for future work.

## **References**

AELLEN, Kurt; THOMAS, Keller; MEYER, Paul; WIRGAND, Jürgen – *Système d'évaluation de logements (SEL)*. Berne, Ed. Office Fédéral du Logement, 1979.

ASSOCIATION QUALITEL – *Guide Qualitel*. Paris, Ed. Association Qualitel, 1989.



- BLACHÈRE, Gérard. – *Un langage pour la description (typologique) d'une habitation*. Paris, Cahiers du Centre Scientifique et Technique du Batiment, n.º 135, 1972.
- BONELLI, R., MENNA, F., MOLTKE, W. – *Walter Gropius e l'habitat del Novecento*. Effelle, Roma, 1983.
- BOUDON, P. – *Lived-in Architecture: Le Corbusier's Pessac revisited*. MIT Press, Cambridge, Massachusetts, 1979.
- CABRITA, A. R. - *Boa Habitação: Do conhecimento à Gestão da Qualidade*. Lisboa, LNEC, 1987.
- COELHO, A. B. – *Análise e Avaliação da Qualidade Arquitectónica Residencial*. Volume I. Lisboa, LNEC, 1993.
- DUARTE, J. Pinto; PAIVA, J. Vasconcelos - *Normas Técnicas para Projecto de Edifícios de Habitação*. Lisboa, Ed. LNEC, 1994.
- HABRAKEN, N. J. - *Type as a Social Agreement*. Asian Congress of Architects, Seoul, 1988.
- KORF, Richard E. - *Space-Efficient Algorithms*, in *ACM Computing Surveys*, Vol. 27, No. 3 September 1995.
- MINISTÉRIO DA HABITAÇÃO E OBRAS PÚBLICAS e LABORATÓRIO NACIONAL DE ENGENHARIA CIVIL (MHOP e LNEC) – *Instruções para projectos de habitação promovida pelo estado*. Lisboa, Ed. LNEC, 1978.
- MINISTÉRIO DO EQUIPAMENTO SOCIAL (MES) – *Recomendações Técnicas para Habitação Social (Portaria n.º 580/83, de 17 de Maio)*. Lisboa, MES e LNEC, 1985.
- PEDRO, J. Branco - *Indicadores de Qualidade Arquitectónica Habitacional*. Dissertação apresentada em cumprimento das exigências de provas de doutoramento na Faculdade de Arquitectura da Universidade do Porto. Porto, Ed. FAUP, 2000.
- PORTAS, Nuno, GOMES, Ruy – *Estudo das funções e da exigência de áreas da habitação - Necessidades familiares e áreas da habitação. Análise de exigências por funções da habitação. Volume I*. Lisboa, Ed. LNEC, 1964a.
- PORTAS, Nuno, GOMES, Ruy – *Estudo das funções e da exigência de áreas da habitação - Necessidades familiares e áreas da habitação. Análise de exigências por função da habitação. Volume II*. Lisboa, Ed. LNEC, 1964b.
- PORTAS, Nuno – *Funções e Exigências de áreas de Habitação*. Colecção Informação Técnica de Edifícios, nº 4. Lisboa, Ed. LNEC, 1969.
- RADFORD, A. and GERO, John - *Design by Optimization in Architecture, Building, and Construction*, Van Nostrand Reinhold, New York, 1988.
- SERGEANT, J. – *Frank Lloyd Wright's Usonian Houses: the case for organic architecture*. Whitney Library of Design, New York, 1976.
- STINY, G., GIPS, J. - *Shape Grammars and the Generative Specification of painting and Sculpture*. In C V Freiman (ed) *Information Processing 71*, pages 1460-1465, Amsterdam, North-Holland, 1972.

STINY G, 1981, "A note on the description of designs" *Environment and Planning B: Planning and Design* **8** 257-267.

TOFFLER, A. - A Terceira Vaga, Livros do Brasil, Lisboa e Rio de Janeiro, 1984.

VIDLER, Anthony - The Production of Types. *Oppositions* 8, 1977.

WINSTON, Patrick - Artificial Intelligence, Addison-Wesley, New York, 1993.

## 2. Methodology

### 2.1 Concepts

Stiny and Mitchell (1978) listed three tests to confirm if a grammar has any explanatory or predictive value. First, it should reveal the common, underlying features of designs in the corpus -- the descriptive test. Second, it should provide the criteria to determine whether a building is a design in the language -- the analytic test. And third, it should specify how to generate new designs in the language -- the synthetic test. Given the goal of generating customized designs, we propose to apply an additional test to a discursive grammar. This new test states that a discursive should possess the means to generate designs that match given criteria. We call this test goal test. These four tests were used to structure the methodology followed in the development of proposed Malagueira grammar as shown in Table I and diagrammed in Figure 3.

The tests validate the grammar, but researchers have carried them out by inquiring its ability to generate certain type of designs. Stiny and Mitchell initiated such a strategy while demonstrating the descriptive value of their Palladian Grammar through the generation of the plan for Villa Malcontenta, a design in the corpus. Koning and Eizenberg (1981) continued this strategy by proposing three new houses after their grammar for Frank Lloyd Wright's Prairie houses to show its syntactic correctness. In our study, we extended the strategy by showing the generation of a house designed by Siza after the grammar was developed, and by showing the generation of criteria-matching designs. Thus, the tests can be more accurately reformulated as follows:

Descriptive test: can the grammar generate designs in the corpus?  
Analytic test: can the grammar generate existing designs not in the original corpus?  
Synthetic test: can the grammar generate new designs in the style?  
Goal test: can the grammar generate designs that match criteria?

In addition to the generation of designs, performing the tests requires testers to determine whether such designs are in accordance with the goals "in the corpus," "in the style," and "match criteria" required by the tests. The ideal tester for the Malagueira grammar would be Siza, the original author, and his clients. Unfortunately, although Siza was very supportive, he was very busy and, therefore, it was necessary to limit the use of his time. We overcame this constrain by submitting designs to Siza only when they had been approved by other testers. The use of multiple testers also permitted to crosscheck the grammar. These additional testers were for the Siza's collaborator (2<sup>nd</sup> author), the author of the grammar (3<sup>rd</sup> author) and his collaborators (4<sup>th</sup> authors), and other authors non-familiar with the grammar (5<sup>th</sup> authors) in the descriptive, analytic, and synthetic tests, and fictitious clients in the goal test.

The methodology comprised the following seven phases: preparatory, descriptive, analytic, synthetic, goal, discursive, and implementing phases, which roughly correspond to Chapters 3 through 8 of this document. With few exceptions, each phase was targeted at the performance of the test with the same name, although it included the performance of other tests, as well. There are explicit and implicit reasons to perform several tests in each phase. The explicit reason was to subject the grammar to tighter tests in subsequent phases to improve its accuracy. For instance, in the description phase both Siza and the author of the grammar had confirmed that grammar revealed the underlying common features of the designs in the corpus. Nevertheless, this did not prove that it also succeeded in revealing such features to designers who did not have

previous knowledge of the Malagueira work. Thus, in the goal phase another descriptive test was carried out with designers in such conditions. The explicit reason was that the performance of some tests indirectly requires the performance of other tests. For instance, to "generate a design in the style that matches given criteria" requires one to check whether the grammar can generate a design at all (synthetic test,) whether the design matches the criteria (goal test), and whether it is in the style (analytical test.)

The flow diagram of the proposed methodology is represented in Figure 3 and the sequence of steps is diagrammed in Figure 4. Please compare both diagrams for further information. A brief description of the procedures involved in each methodological phase is provided below.

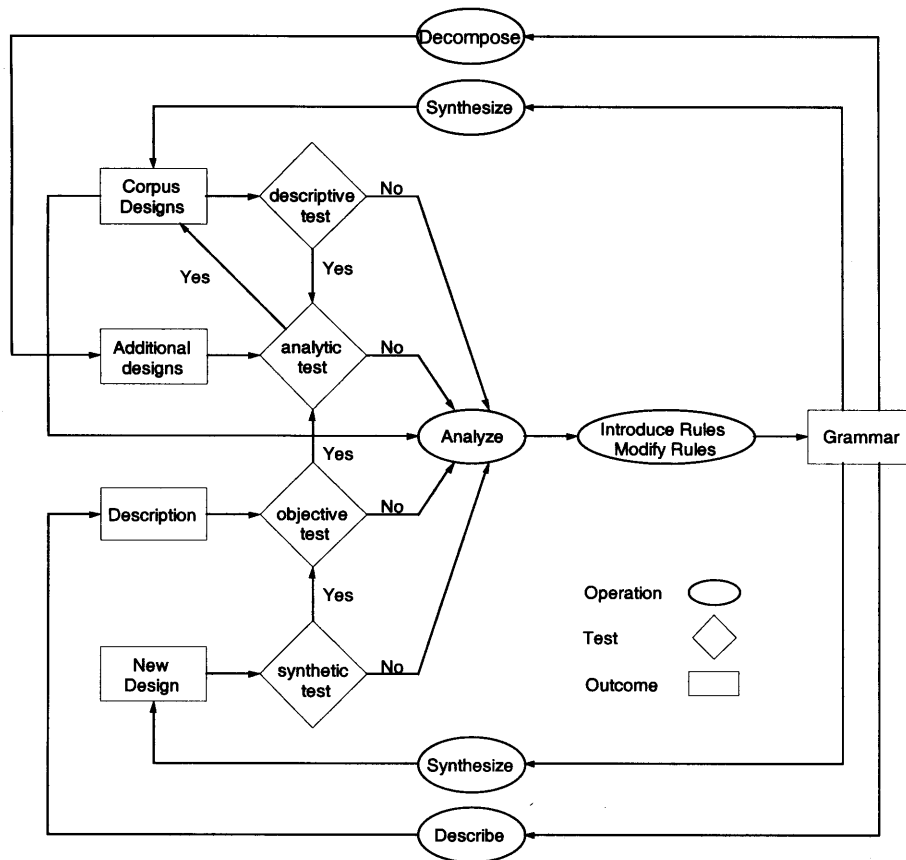


Figure 3 - Flow diagram of the proposed methodology

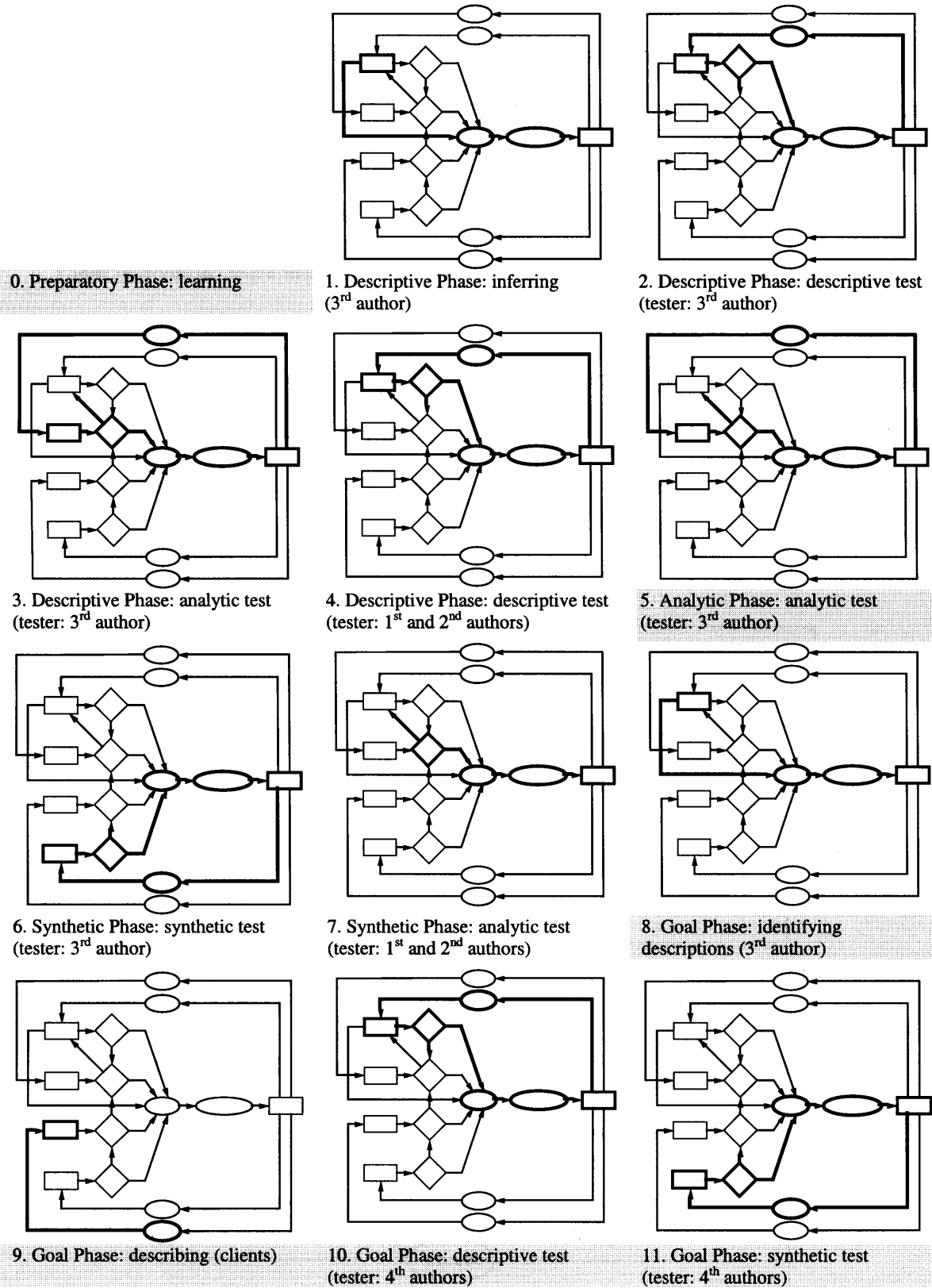


Figure 4 - Sequence of steps in the methodology. For an accurate analysis, please match each individual diagram with the diagram in Figure 3. Bold lines represent the steps followed in each phase.

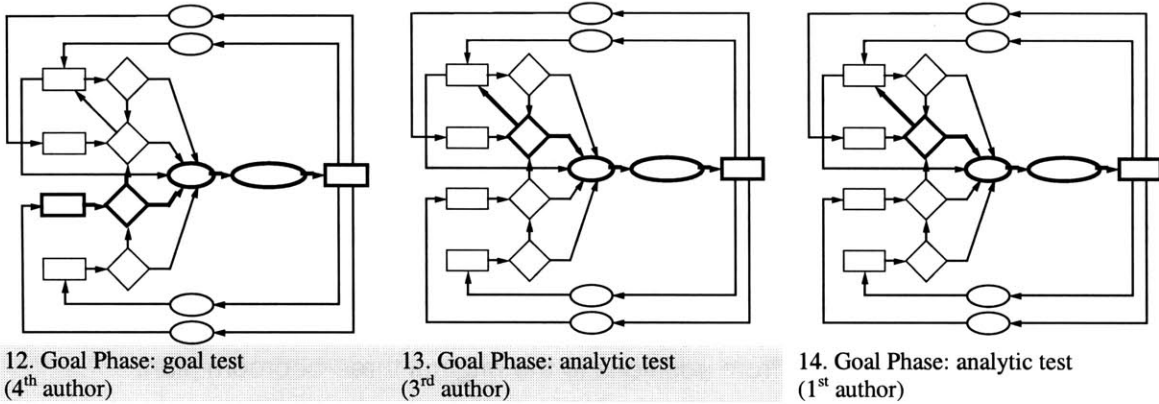


Figure 4 (continued) - The flow of tests in the methodology. For an accurate analysis, please match each individual diagram with the diagram in Figure 3. Bold lines represent the steps followed in each phase.

## 2.2 Preparatory phase

The preparatory phase was aimed at gaining a basic understanding of the fields of study -- performance criteria, grammars, and optimization -- and the specific problem used as a case study -- the Malagueira development. It consisted of a literature review, and the gathering of information on Malagueira. The results of the literature review are presented in Chapter 3. The sources of information included drawings, interviews with the designers, and field trips to Malagueira. The drawings were collected from the archives of Siza's main office in Porto and in the local office he established in Évora to support the project and included drawings at 1/100, 1/50, and 1/20 and 1/1 scales. The set of interviews included interviews with Siza and interviews with his main collaborator in the Malagueira project, Nuno Lopes, who was in charge of the Évora office. The interviews were led in an informal way and permitted the gathering of information about the history of the project and the process of designing the houses. The material collected at Malagueira included slides from the interior and exterior of the houses, as well as notes from conversations with local residents. This material was used to

complete the information provided by the drawings. For instance, the four and five bedroom variations of one of the housetypes (Type D, please see Chapter 4) were not found among the drawings collected but they were identified and visited in person. Conversations with residents helped to understand the genesis of the urban development and the houses. For instance, we learned that these four and five bedroom variations resulted from additions to the original three-bedroom plan.

### **2.3 Descriptive phase**

The descriptive phase was targeted at the development of a shape grammar for the Malagueira houses. It included four steps: the analysis of the collected drawings, the sketching of the grammar, the generation of an existing design, and new interviews with the 1<sup>st</sup> and 2<sup>nd</sup> authors (Siza and his collaborator.) This process corresponds to steps 1 through 4 of the diagram in Figure 4.

The first step required the analysis of the designs in the corpus to infer the shape grammar rules. This analysis included functional, topological, and dimensional analyses. These analyses revealed the formal structure behind the prototypical designs.

In the second step, the generations of the prototypical designs were reconstructed following the rules inferred in the previous step. The goal was to highlight the commonality of formal structure for the two designs, thereby assuring that the grammar fulfilled the requirements of the descriptive test.



In the third step, the remaining designs in the corpus were considered one by one. Each type another design was considered, an analytic test was implicitly performed. In an analytical test, the rules of the grammar are applied in reverse order starting with the final design and terminating with the initial shape. The goal is to decompose the design and determine whether the grammar can account for its generation, thereby confirming it as an instance of the style.

In the step 4, the last in the descriptive phase, we undertook an interview with Siza and Nuno Lopes to perform a final descriptive test. The grammar was explained to the designers while showing the rules being applied in the generation of the prototypical designs. Then, they were asked whether the grammar succeeded in revealing the underlying structure of the Malagueira houses.

The strategy followed in the analysis of the designs and in the sketching of the grammar followed, to a certain extent, the process used by Siza at Malagueira, who after having designed the first two housetypes -- a frontyard and a backyard house -- designed the remaining as variations of the first. Similarly, the grammar was sketched after the prototypical designs, and then successively refined to account for the generation of the remaining designs.

## **2.4 Analytic Phase**

The goal of the analytic phase was to perform a final analytic test to check whether the grammar could account for the generation of a new patio house designed by Siza after

the grammar had been developed. As explained in the previous section, the rules of the grammar were applied in reverse order to decompose the design into the initial shape. The test was carried out with some limitations because the size and shape of the lot was different from the standard Malagueira lot used in the other designs. The result showed that the rules of the grammar could successfully decompose the new design, with the exception of design details originated by such a difference (please see Section 7.1.) To the extent of our knowledge, it was the first time that such a test was carried out. The test can only be applied to a grammar for the work of a living architect and the other grammar in these circumstances, the grammar for Glen Murcutt's country houses (Hanson and Radford 1986) was tested in a different way. In this case, both the authors of the grammar and the original author designed a house for the same client. The former following the rules of their grammar, and the latter following his traditional design process. At the end, both designs were compared. This test is a goal test similar to the ones performed in the goal phase, as described further below.

## **2.5 Synthetic Phase**

The synthetic phase was targeted at the generation of a new design in the Malagueira style and it included a synthetic and an analytic test. The synthetic test checked the capability of the grammar to generate a new design and the analytic test confirmed it as an instance of the style. The descriptive phase had shown that recent designs were variations from the prototypical designs obtained by different rule applications, like in the rule for locating the staircase mentioned in Section 2.3. The same strategy was followed in the generation of an entirely new design. Siza had designed only one backyard

housetype because there was no demand for such a type. The new design was a backyard housetype, which differed from the one Siza had designed in the location of the staircase (please see Section 6.2 for further detail.) The new design was shown to Siza amidst several frontyard designs he had designed. The backyard house was purposefully omitted. Interestingly enough, Siza did not distinguish the new design from those that he had designed. Namely, he did not notice that the shown backyard design was not his. At one point, he seemed confused because of the different location of the staircase, but he acknowledged its validity, and dismissed his doubts. Not even when the generation of the new design was shown in detail did he notice that it was not his design, thereby confirming the design as an instance of the style, and validating the grammar.

## **2.6 Goal Phase**

In the previous phases, the focus was on the development and refinement of the Malagueira shape grammar. With these goals in mind, descriptive, analytic and synthetic tests were performed. These tests can be viewed as experiments in which the author of the grammar (3<sup>rd</sup> author) was the sole subject and the original designers (1<sup>st</sup> and 2<sup>nd</sup> authors) were the control group. The general goal was to check whether the 3<sup>rd</sup> author by using the grammar could achieve a performance similar to the 1<sup>st</sup> and 2<sup>nd</sup> authors. One of the aims of the goal phase was to continue refining the grammar by testing its ability to convey the rules to designers who were less or non-familiar with the design language (4<sup>th</sup> and 5<sup>th</sup> authors.) Therefore, additional tests were performed with such designers.

However, the focus in this phase shifted to the development of the description grammar. Stiny mentioned two problems that one had to solve in undertaking such an enterprise. (Stiny 1981) The first was fixing the contents of the description, that is, to decide which features of the designs are considered relevant and then to describe them in an appropriate way, through verbal or numerical descriptions. The second was to define the description rules. Given that the ultimate goal is the generation of designs that match descriptions, we have identified a third problem: how to arrive at a design given its description. Solving these problems is crucial, as it will determine what questions to ask the client and how to derive the design. We approached the three problems in four different ways. First, we considered the categories included in Siza's documents. Second, we came out with categories after our own description of the existing designs. Third, we asked fictitious, prospective clients to describe the houses that they needed. And fourth, we monitored designers in their attempt to generate designs that matched descriptions. Together, these approaches led to the simulation of a goal test. The test was simulated because the grammar did not include a mechanism to guide the derivation of a design towards a specified goal. In fact, the idea behind the simulation was to develop such a mechanism.

To simulate the test, two series of experiments were undertaken (Table II.) In the first series, all the subjects were asked to design a house for the same client and it included two groups of subjects. The first included subjects who had collaborated in the development of the grammar (4<sup>th</sup> authors) and who functioned as the control group. The second group included designers who did not have previous knowledge of either Siza's work at Malagueira or the grammar (5<sup>th</sup> authors.) The client was familiar with the cultural

context of the Malagueira houses. In the second series of experiments, none of the experimental subjects was knowledgeable of the Malagueira work, and none of the clients was familiar with its cultural context. This series included two sets of experiments. In the first set, the subjects were not allowed to change the grammar, whereas in the second set they were. Results showed that designers could use the grammar to generate criteria-matching designs, and provided important clues on how to incorporate such a mechanism into the grammar. Both the experimental setting and the results are described in detail in Chapter 5. Each of the series of experiments iterated through steps 8-14 of the diagram in Figure 4.

**Table II - Experiments of the goal phase**

Phase	Series	Set	Subjects	Client	Steps
Goal	1 <sup>st</sup>	1 <sup>st</sup>	4 <sup>th</sup> authors (control group)	Familiar with cultural context	9-14
		2 <sup>nd</sup>	5 <sup>th</sup> authors	Familiar	9-14
	2 <sup>nd</sup>	1 <sup>st</sup>	5 <sup>th</sup> authors	Non-familiar	9-14
		2 <sup>nd</sup>	5 <sup>th</sup> authors	Non-familiar	9-14

In step 8, the designs were analyzed to identify explicit and implicit descriptions. In step 9, clients were asked to describe the house that they needed. First, they were given a Web-based catalog of Malagueira houses, and shown a few houses in detail to make them aware of the functional and aesthetic possibilities provided by the style. Then, they were asked to describe their needs. Their description was video-recorded and later given to the experimental subjects. In step 10, the grammar was explained to the experimental subjects by showing them the generation of some existing designs. Then, they were asked to reconstitute the generation of other existing designs. Finally, they were asked to comment on the ability of the grammar to describe the style (descriptive test.) In step 11, each subject was given the video with the client interview and then

asked to generate a house for the client out of the grammar rules. At the end, they were asked to assess the capability of the grammar in specifying how to generate a Malagueira house (synthetic test.) In step 12, the houses were shown to the clients who were asked whether the design satisfied the needs described at the outset (goal test.). In step 13, the author of the grammar certified the designs' compliance with the style (analytic test.) Steps 11 and 12 were repeated until the goal and the analytic tests were successful. Finally, in step 14, all the designs were shown to Siza who was asked whether he considered them Malagueira designs, for a final analytic test.

## **2.7 Discursive Phase**

The goal of the discursive phase was to develop the description grammar and to incorporate into the shape grammar the mechanism that permitted the generation of criteria-matching designs, thereby obtaining the discursive grammar. The experimental results of the goal phase permitted the identification of relevant descriptions and heuristics used by designers in their efforts to generate designs that matched criteria. These results, however, were not sufficient and other descriptions were introduced after studies on performance criteria. Once the problem of which categories to include was solved, the next step was the development of the description rules by constructing the descriptions of the left and right sides of the rules. The next step was to perform a descriptive test consisting in the generation of the description, followed by the generation of the corresponding existing design. This test showed that a discursive grammar constituted a valid model. Fine-tuning the grammar will require the performance of synthetic, goal, and analytic tests. Given the complexity of the descriptions, it is

recommended to do this after a computer implementation is completed. In this circumstances, the tests will require: (1) the automatic generation of new designs to satisfy the needs of clients; (2) to ask the clients if the designs matched their needs, and (3) to ask the original designer if the designs were in the style.

## **2.8 Implementation Phase**

The implementation phase was aimed at the development of the interpreter for the Malagueira discursive grammar. An interpreter is a computer program that implements the rules of the grammar and, therefore, is capable of generating designs in the language. Chronologically, this phase was parallel to the discursive phase so that the structure and contents of the descriptions in the grammar could be developed in a way that would make its computer implementation easier.

## **2.9 Summary**

In the previous sections, we described the methodology followed in the development of the Malagueira grammar and interpreter. This methodology was structured upon three tests proposed by Stiny and Mitchell (descriptive, analytic, synthetic tests) and a fourth tests developed in this study (goal test.) The methodology comprises six phases, excluding an initial, preparatory phase. These phases can be grouped into two parts. The first part is concerned with the shape grammar and it is steered towards the development of the grammar (descriptive phase), and the refinement of its performance

(analytic and synthetic phases). The second part is mainly concerned with the description grammar and it is directed towards the definition of the categories in the description (goal phase), the development of the description grammar (discursive phase), and the development of a computer interpreter. We propose this methodology as general methodology for the development of discursive grammars.

## **References**

STINY G - A note on the description of designs. *Environment and Planning B: Planning and Design* 8 257-267, 1981.

STINY, G and MITCHELL, W J - The Palladian Grammar. *Environment and Planning B*, Vol. 5, pages 5-18, 1978.

KONING, H, and EIZENBERG, Julie - The Language of the Prairie: Frank Lloyd Wright Prairie Houses. *Environment and Planning B*, Vol. 8, pages 295-323, 1981.

HANSON, Neil, RADFORD, Anthony - On modeling the work of the architect Glen Murcutt. *Design Computing*, Vol. 1, pages 189-203.



## **3. Precedents**

### **3.1 Introduction**

Related work can be divided into three categories: performance criteria, grammar, and optimization studies. Performance criteria studies are concerned with the identification of the requirements for the design of housing, and with the evaluation of both designs and houses. Grammar studies focus on the development of shape and description rules that encode languages of designs. Optimization is concerned with finding the best solution for a stated problem. This chapter introduces additional background information on performance criteria and shape grammar studies. For the purposes of this work, optimization was sufficiently covered in the introductory chapter.

### **3.2 Performance criteria**

Performance criteria studies started with studies aimed at understanding how people used the dwelling space. These studies were important to identify the user needs and to codify them into a coherent set of design requirements. Among such studies developed for the Portuguese context is the Study of the Housing Functions and Area Requirements (Portas and Gomes 1964a and b).

A practical result of these studies was the development of guidelines aimed at helping the designer to establish the housing program. In Portugal, these guidelines were the Design Guidelines for State Promoted Housing-IPHPE (MHOP and LNEC 1978), the

Technical Recommendations for Social Housing-RTHS (MES and LNEC 1985), and the Technical Guidelines for the Design of Housing Buildings-NTPEH (Duarte and Paiva, 1994). These normative documents were sequentially in use at the time of Siza's work at Malagueira, and therefore, were taken into consideration in his designs.

Design guidelines led to the development of quality evaluation studies to determine whether designs and dwellings satisfied design requirements. The outcome of these studies was the development of several quality evaluation methods to measure the performance of designs and dwellings against such requirements, such as the Dwelling Evaluation System-SEL (Aellen et al 1979), and the Qualitel Guide (Association Qualitel 1989).

Recently, Pedro updated and summarized previous studies into housing programming and evaluation methods adjusted to the contemporary Portuguese reality called Housing Program (Pedro 1999a and b) and Architectural Housing Quality Indicators (Pedro 2000). These documents are used as a basis for the development of the discursive grammar proposed in this work for the following reasons. On the one hand, these methods are in line with the documents that regulated the design of housing when Siza designed the Malagueira houses, they are recommended by the two major Portuguese institutes that regulate housing issues—the National Housing Institute<sup>1</sup> and the National Laboratory for Civil Engineering<sup>2</sup>, and they take into account the Portuguese contemporary reality.

On the other, the goal of the discursive grammar is to generate not only the houses that Siza has already designed, but also new houses in the style. Therefore, the use of such

---

<sup>1</sup> *Instituto Nacional da Habitação (INH)*

<sup>2</sup> *Laboratório Nacional de Engenharia Civil (LNEC)*

two documents is likely compatible with Siza's rules, and it permits the generation of new contemporary designs.

For a better understanding of the proposed discursive grammar, the programming and evaluation methods developed by Pedro are summarized below.

### **3.2.1 Programming method**

In the studies concerned with analysis of housing, the main motivation to define a housing program is to make it possible to analyze and evaluate the dwellings by verifying the extension to which the program is satisfied. Another motivation is to use the program to define the requirements that will lead to the satisfaction of a high number of users. This second motivation is important because in traditional mass housing provision processes, the designer conceives the houses for anonymous users, whose individual expectations are unknown. The idea is to come up with models of dwellings that satisfy typical programs, which are then used as references by the designer in the design process, or by the analyst in the evaluation process.

The definition of a housing program represents the first step in the design process, in which the problem data and the requirements to be satisfied by the solution are identified. Pedro considers that a housing program consists of three parts: the *program data*, the *quality requirements*, and the *reference models*. The *program data* classify the dwelling functions and rooms, characterize the users, and identify the most common dwelling types. The *quality requirements* define the level of performances of the dwelling rooms that satisfies the user needs. The *reference models* are sample solutions for housing programs. The goal of the discursive grammar is to avoid the use of reference models by encoding the rules for generating specific solutions for given

problems. In the application of his method, Pedro considers the existence of four physical levels: spaces and rooms, dwelling, building, and neighborhood. However, given the scope of the discursive grammar, only the first two are considered in the present work. A brief explanation of the concepts of program data, and quality requirements at the spaces and dwelling levels is provided below.

### Program data

The program data includes the identification of the *spaces and rooms* that form a dwelling (e.g. kitchen, double bedroom, etc.) and the identification of *functions* (e.g. washing clothes) *and activities* (e.g. hand and mechanic washing) that the dwelling will shelter. It also includes the assignment of functions and activities to spaces and rooms (e.g. washing clothes in the kitchen, bathroom, or laundry), the characterization of the schedule of the different activities in the spaces (e.g. washing clothes is done weekly, during the day, and lasts for several hours), and the characterization of the different dwellers in terms of age and relationship to the household (e.g. washing clothes is done by individual household members who are older than 14 years, or by an employee).

### Design or quality requirements

The design requirements define the performance level of the rooms and construction units to ensure the satisfaction of the user needs. The formulation of requirements defines the required performance quality, but it does not provide solutions. Performance quality is defined in terms of levels. A quality level is a set of requirements that define a degree of user requirements satisfaction. There are three quality levels: minimum, medium, and maximum (or high). The minimum level is defined by a set of requirements that satisfy the elemental necessities of daily life at a level of performance that does not constrain the household's way of living in any significant way. The medium level is

defined by a set of requirements that supports daily life better than the minimum level requirements by taking into consideration different ways of living and the expectable evolution of household needs. The *maximum* level is the one, beyond which, the performance of the dwelling does not improve significantly and some investment problems might indeed arise.

**Table 3.1 - Tree of qualities applicable to the dwelling level.**

Quality	Groups of qualities	Qualities	Elemental Qualities				
	Comfort	21	acoustic	27	acoustic	100	
			visual	45	solar orientation	31	
					natural lighting	26	
					shading	10	
					views/hurdles	20	
					views/monitoring	13	
						100	
			air flow	28	air flow	100	
						100	
		Security	11	use	48	use	100
				fire	26	fire	100
				intrusion	26	intrusion	100
						100	
		Spatial adequacy	30	capacity	32	type and number of rooms	42
						equipment	28
						furnishable wall extension	30
							100
				spaciousness	43	area	49
						useful dimension (w, l)	33
						height	19
							100
			functionality	25	functionality	100	
						100	
	Spatial articulation	22	privacy	59	internal	44	
					external	56	
						100	
			accessibility	41	among rooms	50	
					more than one floor	28	
					handicapped	22	
						100	
	Personalization	16	adaptability	37	adaptability (among rooms)	25	
		100			expandability (perimeter)	75	
			appropriation	63	appropriation	100	
						100	

## COMFORT

**Acoustic comfort:** The dwelling must provide an adequate level of acoustic insulation among its rooms and between these and the surroundings. Ex.: The sleeping and the living zones are separated by a door or staircase. Nominal

**Solar orientation:** The dwelling must provide direct sunlight of its rooms during different periods of the day, and it should provide rooms with an adequate solar orientation for the functions that they host. Ex.: The solar orientation of a bedroom is optimal between east and southwest.

**Natural lighting:** The dwelling must provide rooms with an adequate solar orientation for the functions that they host. Ex.: the ration between the area of a room's windows and its area is between 5 and 22%.

**Shading:** The dwellings must have openings with devices that permit their total or partial shading. Ex.: The bedrooms have shading devices that enable total darkness.

**Views:** The dwelling must provide an adequate visual contact with the exterior through a vision field free of obstacles, and through the size and details of windows. Ex.: The ration between the area of a room's windows and its area is between 5 and 22%.

**Visual monitoring:** The dwelling must allow the user to monitor activities taking place in the adjacent external spaces, and views over pleasant scenes. Ex.: The openings have views over children's leisure places.

**Air quality:** The dwellings must provide for ventilation that allows air renewal, pollution substances removal, enough air for combustion equipment to work, and the extraction of smoke and gases that these produce. Ex.: There exist two facades on opposite sides that permit cross-ventilation.

## SECURITY

**Use security:** The dwelling must protect the dwellers from aggressive agents, circulation accidents (hits or falls), and falls from elevated places, during the normal use of its spaces and equipment. Ex.: The stove is not close to an operable window.

**Fire security:** The dwelling must minimize the risk of fire initiation and propagation, facilitate user-evacuation, rescue and fire-fighting operations, and protect users from smoke and high temperatures. Ex.: The inhabited rooms have access to the exterior through one or more circulation spaces separated from other rooms, alternative, or emergency exits.

**Intrusion security:** The dwellings must ensure the protection of users and their property from the intrusion of people, animals, and objects. Ex.: One can clearly see the space adjacent to the entrance door.

## SPATIAL ADEQUACY

**Type and number of rooms:** The dwelling must have rooms that enable their adequate use by the number of its users. Ex.: The dwelling has two living-rooms.

**Equipment:** The dwelling must have equipment installed during construction that enables its adequate use by the number of its users. Ex.: The extension of the kitchen's counter of a two-bedroom dwelling is between 2.5 and 3.4m.

Figure 3.1 – The qualities considered in Pedro's programming method,

### **SPATIAL ADEQUACY (continued)**

**Furnishable wall extension:** The dwelling must have walls with extensions that enable the placement of furniture adequate to the number of its users. Ex.: The extension of bedroom furnishable walls of a two-bedroom dwelling is between 18 and 26m.

**Useful area:** The dwelling must contain rooms with areas that can accommodate the equipment, furniture, and circulation space required for their adequate use by the number of its users. Ex.: The area of bedroom area of a two-bedroom dwelling is between 19 and 30m<sup>2</sup>.

**Height:** The dwelling must contain rooms with a height that is adequate to their use for housing. Ex.: The height of inhabitable rooms is between 2.3 and 2.70m.

**Functionality:** The dwelling must provide adequate conditions for users to perform the dwelling functions. Ex.: It must be possible in all the bedrooms to place the beds away from lateral objects, with the top against the wall, and with a distance between the bottom and the opposite wall no smaller than 0.5m.

### **SPATIAL ARTICULATION**

**External privacy:** The dwelling must permit privacy at the personal and the household level through the way it relates to the exterior. Ex.: There are no spaces adjacent to the dwelling with direct views towards the sleeping, living, or water closet zones.

**Internal privacy:** The dwelling must permit privacy at the personal and the household level through the way the rooms are related to each other. Ex.: The kitchen and the living room have access to a water closet through circulation spaces separated from the sleeping zone.

**Accessibility among rooms:** The dwelling must provide users with easy physical links among rooms strongly related. Ex.: The average extension of the path between the bedrooms and the corresponding water closet should be lower than 12.5m.

**Accessibility between rooms on more than one floor:** The dwelling must provide users with easy physical links among rooms strongly related. Ex.: There is at least a bedroom at the entrance level.

**Handicapped users accessibility:** The dwellings must permit the use of its rooms by handicapped users. Ex.: The useful dimension of the kitchen is between 1 and 1.50m.

### **PERSONALIZATION**

**Appropriation:** The dwelling must allow users to make changes to the dwelling to personalize it. Ex.: there are spaces in which the user can have plants.

**Expandability:** The dwelling must permit changes to its perimeter to adapt it to the user's life style. Ex.: the dwelling is expandable.

**Adaptability among rooms:** The dwelling must permit changes to the relations among rooms to adapt it to the user's life style. Ex.: The kitchen and the living can be merged/separated through a mobile device.

Figure 3.1 (continued) – The qualities considered in Pedro's programming method,

The design requirements are a list of qualities that need to be satisfied. The main qualities are comfort, security, spatial adequacy, spatial articulation, personalization, aesthetics, and cost. Each of these main qualities in turn includes several qualities. For instance, comfort includes acoustic and visual comforts. These qualities also include elemental qualities. For instance, visual comfort includes solar orientation, natural lighting, shading, and views. The tree of qualities is shown in Table 3.1 and the definitions are presented in Figure 3.1.

### **3.2.2 Evaluation method**

The goal of the evaluation method is to support the decision-making process. The evaluation method measures the satisfaction of design requirements by a housing solution. Given the complexity of such a solution, to proceed with evaluation it is necessary 1) to decompose the general goal into subgoals, 2) to establish the degree of importance of each subgoal, 3) to measure the satisfaction of each subgoal, and 4) to calculate the final result. The housing evaluation method is a multi-criteria method, in which, the tree of qualities becomes a tree of viewpoints. This tree includes main viewpoints as its branches (groups of requirements or qualities), and elemental viewpoints as its leaves (basic requirements or qualities). The tree was defined taking into account the following criteria: 1) to include all the elemental viewpoints considered relevant; 2) to include only commensurable viewpoints; 3) to ensure that each main viewpoint does not include too many sub-viewpoints; and 4) to ensure that the depth of the tree is as small as possible.



**Table 3.2 - From groups of qualities and qualities to evaluation criteria and requirements**

	Abstract concept	Concrete example	Definition
1	Physical level	Dwelling	
2	Groups of qualities	Environmental comfort	
3	Quality	Acoustic comfort	The dwelling should be conceived as to ensure an adequate acoustic insulation among the different rooms of the dwelling and between the dwelling and its surroundings
4	Quality indicators	<i>Acoustic comfort among rooms</i>	
5	Evaluation criterion	Separation of functional zones	
6	List of requirements		1) There should be a separation by door or staircase between the sleeping and the service and living zones; 2) The separation between the sleeping and the service and living areas should be increased by introducing closets or intermediate rooms with functions that are not sensitive to noise, or by reducing the contact surface among such areas.

The elemental viewpoints are quantifiable through the use of descriptors. A *descriptor* is a set of values that permits to quantify in a numerical scale the performance of the design from an elemental viewpoint. *Evaluation criteria* are the means through which one can relate a feature of the design from an elemental viewpoint to a value on the scale of the descriptor. A *list of requirements* consists of a sequence of requirements that once satisfied lead to values on the descriptor scale. (Table 3.2) A *Transformation function* relates the satisfaction of viewpoints to the descriptor scale. (Figure 3.2)

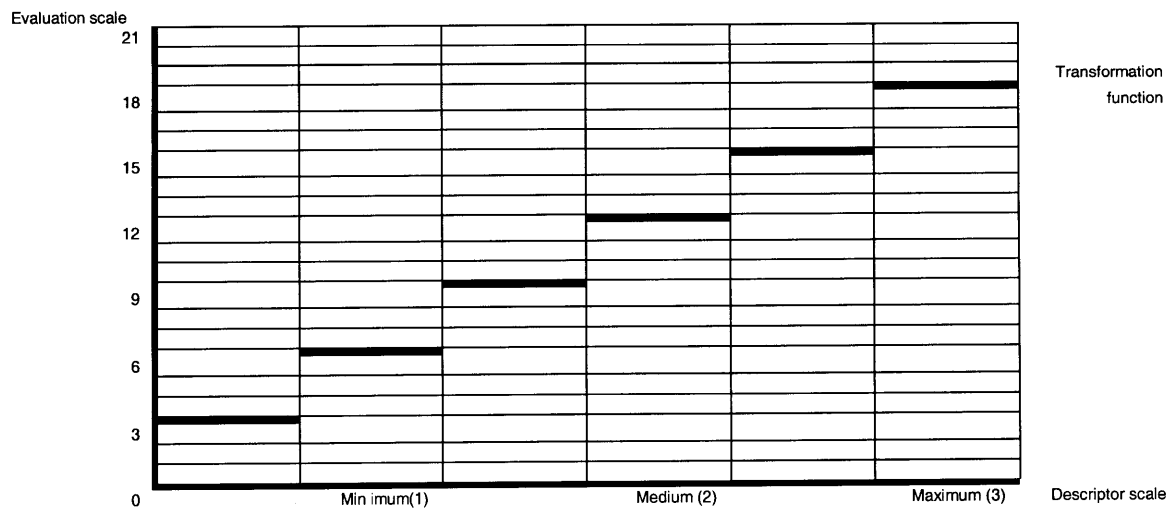


Figure 3.2 – Example of transformation function articulating a descriptor and an evaluation scale.

Each of the viewpoints is assigned a *weighing factor* that expresses its importance for the designer or the evaluator. To define the weighing factors, a method called swing weights is used. This method comprises three steps: 1) ordering of the viewpoints, considering their relative importance; 2) assignment of the value 100% to the most important viewpoint, followed by the assignment of values to the remaining ones by comparing their importance with that of the most important one (for instance, 90%, 80% etc.); and 3) standardization of the weights onto a scale that adds up to 100. The result of such a process also is illustrated by the weights in Table 3.1.

After evaluating the design from the different elemental viewpoints, one can obtain a summarized result for the corresponding main viewpoint by calculating the weighted average of the performance of the design from the different elemental viewpoints:

$$V_{vp} = \frac{\sum (V_{svp_i} \times P_{svp_i})}{\sum P_{svp_i}}$$

$V_{svp_i}$  - Value of the design solution performance from sub-viewpoint  $i$   
 $P_{svp_i}$  - Ponderation factor of sub-viewpoint  $i$   
 $V_{vp}$  - Value of the design solution performance from a given viewpoint

The evaluation method is subjective because only some viewpoints are selected, and because the weighing factors are determined by the evaluator. The set of viewpoints were chosen according to the following criteria: to have a significant impact on architectural housing quality, to be relevant for the evaluation context defined, to be likely satisfied by current dwellings, and to be amenable to an objective evaluation. Similar criteria were used in the selection of the qualities included in the discursive grammar, which do not exactly match those in Table 3.1. The subjectivity problem was not an issue because choosing weights is part of the exploration of solutions, as it will be shown in Chapter 7. The discursive grammar adapts the housing programming and evaluation methods just described.

### **3.3 Grammars**

Grammatical design studies had their beginning in a seminal paper by Stiny and Gips (1972), in which they laid the foundation of what was to become the most important algorithmic approach to design. Since their invention, the field grew to encompass a number of technical devices and research issues. Those that are relevant to the current work are briefly presented below.

#### **3.3.1 Shape grammars**

A shape grammar specifies how designs can be generated from an initial shape through the recursive application of shape rules. Shape grammars can be divided into two categories, depending on whether they support, or not, shape emergence. Emergence is the ability to recognize shapes that were not predefined but emerge in the computation. The grammar presented in this work does not support shape emergence it is therefore, a set grammar.

#### **3.3.2 Parametric shape grammars**

A parametric shape grammar is a shape grammar in which rules are parameterized so that each rule represents a set of rules. The grammar proposed in this work is a parametric grammar. A parametric shape grammar can be described by an ordered sequence of five elements, which is called a five-tuple. As a way of illustration, consider the grammar defined by the five-tuple (S, L, T, G, I). S is a set of parametric shape rules of the form  $A \rightarrow B$  that specifies that whenever a shape A is found in the design, it can be substituted by a shape B. In our illustrative grammar, this set is composed of two parametric rules R1 and R2 (Figure 3.3). L is a set of labels that are used to control computations. T is the set of similarity transformations (rotation, translation, scaling,

reflection or any composition of these) under which rules apply.  $G$  is a set of functions that assigns values to parameters in rules, thereby defining specific rules. Both the similarity transformation and the assignment function determine the conditions under which the left-hand side of rules can be matched to a shape in the design during rule application. Finally,  $I$  is the initial shape to which the first rule applies to start a computation. Other rules then apply recursively to continue the derivation of a design within the language defined by the grammar. (Figure 3.4)

The shape grammar formalism can be summarized in the equation:  $C_{n+1} = [C_n - t(g(A))] + t(g(B))$ ,  $n > 0$ , in which  $C_n$  is the shape in the design at step  $n$ . The equation states that for a rule to apply,  $A$ , the shape in the left-hand side of the rule, must be a part of  $C$  under some assignment of values and transformation, in which case it is deleted and substituted by  $B$ , the shape on the right-hand side. For a detailed description of shape grammars and parametric shape grammars see Stiny (1980).

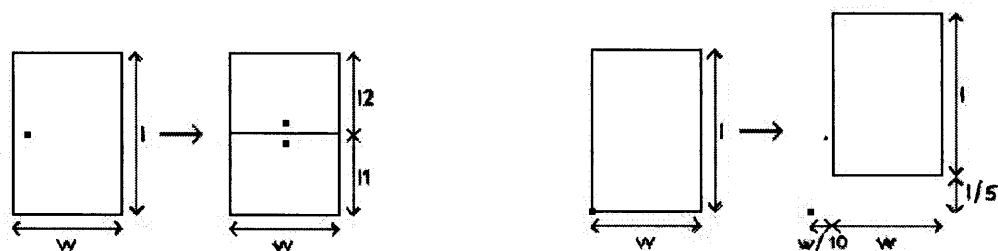


Figure 3.3 – A simple parametric shape grammar consisting of two rules. R1 (left) dissects a rectangle; and R2 (right) translates a rectangle.

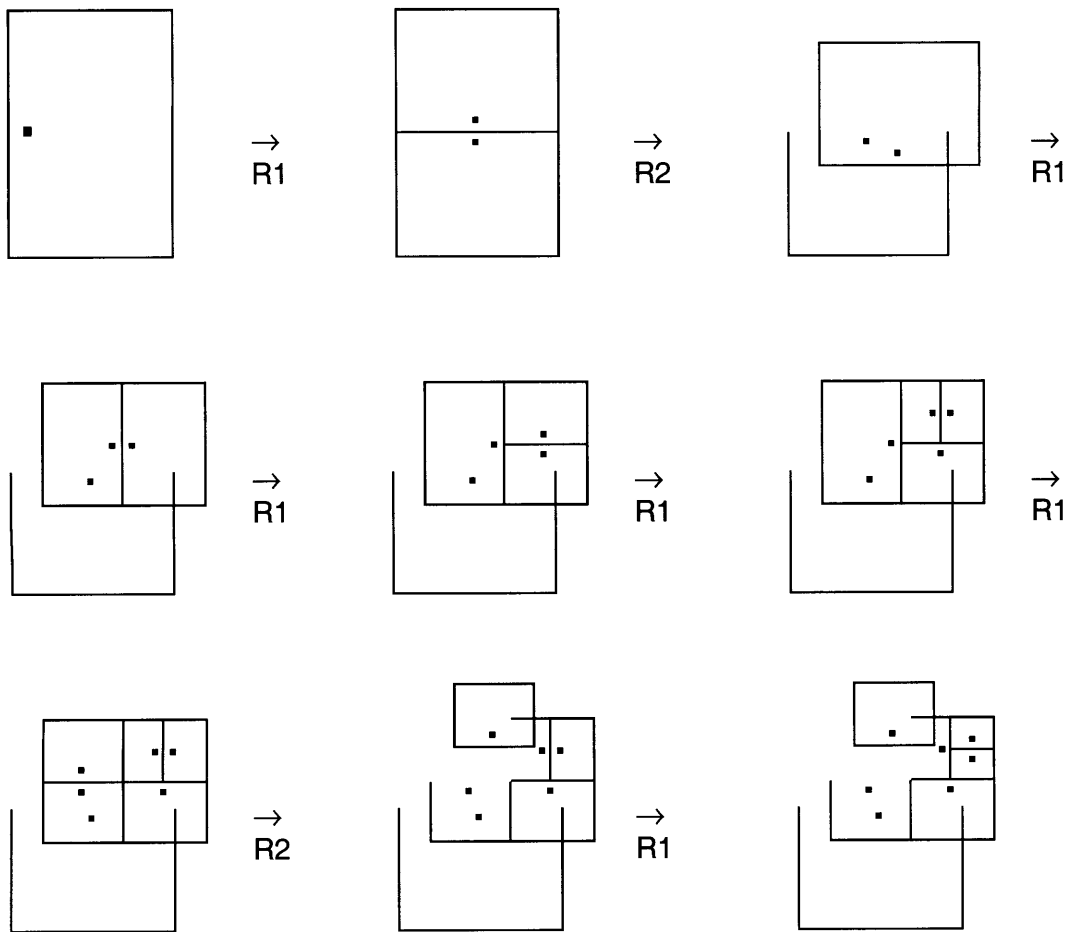


Figure 3.4 – Derivation of a design in the language defined by rules R1 and R2.

### 3.3.3 Shape grammars: analytical and original

Shape grammar studies can be grouped into two different categories: analytical and original. Analytical grammars have been developed to describe and analyze historical styles or languages of designs by architects no longer living. In fact, after the first grammar was developed to explain a corpus of architectural artifacts, the one for Palladian villas (Stiny and Mitchell, 1978), others have been developed with the same purpose over the past twenty years. Among them are Wright's Prairie Houses (Koning and Eizenberg 1981), Buffalo bungalows (Downing and Flemming 1981), Japanese

tearooms (Knight 1981), Queen Anne houses (Flemming 1987), Wren's city churches (Buelinckx 1993), and Taiwanese vernacular dwellings (Chiou and Krishnamurti 1995), to name an important few. Analytical studies use a set of existing designs to represent the language—the corpus—and to infer the rules of the grammar. The grammar is, then, tested by using the rules to generate designs in the corpus, as well as new designs in the language.

Original grammars are concerned with the creation of new and original styles of designs “from scratch.” The use of grammars for creative design has not been explored as deeply as the use of grammars for analytical studies. Although implicit in Stiny and Gips (1972), such use of grammars was only explicitly addressed in Stiny (1980) where he proposes a programme for developing new grammars that is illustrated using Frederick Froebel's kindergarten method of education. Stiny's programme was implemented by Knight who introduced grammars in the design studio. From this experience, Knight highlighted some of the difficulties in using grammars for creative design, which are connected to the “translation of abstract, experimental forms into architectural designs that fit particular design contexts or programmes” (Knight, 1992). Solving this difficulty is central to the current work, which is focused on the design of goal-matching designs.

### **3.3.4 Computer implementation**

Two approaches have been proposed to solve the goal-matching difficulty. The first is to predict what rules will do so that a designer can decide which rules to apply to obtain designs with specific properties. (Knight 1999, 2000) The second approach is to use computer implementations of shape grammars to rapidly generate the results of rules, allowing for a faster search through the space of design possibilities. Such an alternative leads to two paths. The first uses the computer to generate designs, but

requires the designer to assess the generated designs; in short, the computer is used only to accelerate and facilitate the derivation of designs (Tapia 1996; Heisserman 1991; Piazzalunga and Fitzhorn 1998; Wang and Duarte 2000). The second approach requires the computer to perform assessment in addition to generation (Cagan and Mitchell 1993; Shea and Cagan 1996, 1998). In this approach, the computer is explicitly given criteria for a suitable design, which it uses to control generation and traverse the space of design solutions in search of a design that matches the criteria. The search processes that have been proposed so far are stochastic. The process proposed in this work is heuristic and, to a considerable extent, deterministic.

### **3.3.5 Proposed grammar: analytical and original**

The grammar for Siza's houses at Malagueira is in the footsteps of the analytical studies mentioned above. Nevertheless, it is a grammar developed for an evolving project by a living architect. To the extent of author's knowledge, there has been only one other grammar of this kind: the one on the work of the architect Glen Murcutt (Hanson and Radford, 1986). However, unlike the Murcutt grammar, the Malagueira grammar was developed with Siza's support, and therefore, it can be seen as a natural extension of Siza's work at Malagueira. The impact of such a novelty is twofold. First, it is possible to use the architect and the dwellers in addition to existing designs as sources of information to derive the rules of the grammar. Second, it is possible to use the grammar to generate and build new houses in the language. Therefore, the grammar is more than a mere analytical grammar aimed at describing a family of designs. But it is not a full grammar developed from scratch to generate entirely new designs. It is reasonable to consider that it spans between analytical and original grammars.

### **3.3.6 Description grammars**

The concept of description function was developed by Stiny (1981) to account for features of designs not covered by shape grammars. A shape grammar specifies how designs can be generated. A description function describes the design in terms of other features considered relevant according to some criteria of interest. The relation between the shape grammars and description grammars is such that for each shape rule there is one or more corresponding description rules, plus an additional starting description corresponding to the initial shape. As the grammar rules are applied to the evolving design, the corresponding description rules are applied to the evolving description. Thus, as the generation of the design evolves, the description of the design is constructed. Mitchell (1989) suggested a similar approach by proposing the use of first order logic to describe and evaluate designs.

### **3.3.7 Parallel grammars**

Parallel grammars separate different representations or aspects of designs into different computations that interact with each other. This separation facilitates the manipulation of complex design problems, by breaking them into smaller ones. The representations can be visual (elevation, plan, etc.) or symbolic (thermal performance, number of bedrooms, etc.). The joint use of a shape grammar and a description grammar is an example of a parallel grammar. The use of parallel grammars permits to address and solve different aspects separately, for instance, the generation of floors with different programs. In this work, description grammars are used as a way of generating multiple representations of designs, but also, as a way of dealing with the interdependency of design parameters. For more information on parallel grammars see Stiny (1992).



### 3.4 Conclusion

Performance criteria studies have codified the programming and evaluation of housing. The discursive grammar proposed in this work adapts the most recent Portuguese housing programming and evaluation methods because it is compatible with Siza's Malagueira design rules, while permitting the generation of new designs that are adjusted to the current reality. Design grammars have been developed as a way of capturing the algorithmic nature of design. The discursive grammar proposed in this work draws on this formalism by proposing a grammar to capture both such housing methods and Siza's Malagueira design rules and extends it by proposing a heuristic method to search for criteria-matching designs. The discursive grammar is a parametric set grammar that uses parallel shapes and descriptions, and is amenable to computer implementation.

### References

- ASSOCIATION QUALITEL – *Guide Qualitel*. Paris, Ed. Association Qualitel, 1989.
- AELLEN, Kurt; THOMAS, Keller; MEYER, Paul; WIRGAND, Jürgen – *Système d'évaluation de logements (SEL)*. Berne, Ed. Office Fédéral du Logement, 1979.
- BUELINCKX H, 1993, "Wren language of City church designs - a formal generative classification" *Environment and Planning B: Planning and Design* 20 645-676.
- CAGAN, Jon, and William J. MITCHELL, "Optimally Directed Shape Generation by Shape Annealing," *Environment and Planning B: Planning and Design* 20 (1993): 5-12.
- CHIOU S-C, KRISHNAMURTI R, 1995, "The grammar of Taiwanese traditional vernacular dwellings" *Environment and Planning B: Planning and Design: Planning and Design* 22 689-720.
- DOWNING F and FLEMMING U, 1981, "The bungalows of Buffalo" *Environment and Planning B: Planning and Design* 8 269-293.
- DUARTE, J. Pinto; PAIVA, J. Vasconcelos - *Normas Técnicas para Projecto de Edifícios de Habitação*. Lisboa, Ed. LNEC, 1994.

- FLEMMING U, 1987, "More than the sum of parts: the grammar of Queen Anne houses" *Environment and Planning B: Planning and Design* 14 323-350.
- HANSON N L R and RADFORD A D, 1986, "On Modeling the Work of the Architect Glenn Murcutt" *Design Computing*, 1, 189-203.
- HEISSERMAN J A, 1991, *Generative Geometric Design and Boundary Solid Grammars* Ph.D. Dissertation, Department of Architecture, Carnegie Mellon University, Pittsburgh.
- KNIGHT T W, 1981, "The Forty-one Steps: the languages of Japanese tea-room designs" *Environment and Planning B: Planning and Design* 8 97-114.
- KNIGHT T W, 1992, "Designing with grammars" in Schmitt G N (ed) *Computer-Aided Architectural Design* (Wiesbaden: Verlag Viewag) 33-48.
- KNIGHT T W, 1989, "Color grammars: designing with lines and colors" *Environment and Planning B: Planning and Design* 16 417-449.
- KNIGHT T W, 1999a, "Shape grammars: six types," *Environment and Planning B: Planning and Design* 26 15-31.
- KNIGHT T W, 2000, "Shape grammars: five questions," *Environment and Planning B: Planning and Design*.
- KONING H and EIZENBERG J, 1981, "The language of the prairie: Frank Lloyd Wright's prairie houses" *Environment and Planning B: Planning and Design*, 8 295-323.
- MINISTÉRIO DA HABITAÇÃO E OBRAS PÚBLICAS e LABORATÓRIO NACIONAL DE ENGENHARIA CIVIL (MHOP e LNEC) – *Instruções para projectos de habitação promovida pelo estado*. Lisboa, Ed. LNEC, 1978.
- MINISTÉRIO DO EQUIPAMENTO SOCIAL (MES) – *Recomendações Técnicas para Habitação Social (Portaria n.º 580/83, de 17 de Maio)*. Lisboa, MES e LNEC, 1985.
- MITCHELL W J, 1989, *The Logic of Architecture*, (MIT Press, Cambridge, MA).
- PEDRO, J. Branco - Programa Habitacional. Espaços e Compartimentos. Coleção Informação Técnica e Arquitectura, nº4, Lisboa, LNEC, 1999a.
- PEDRO, J B - Programa Habitacional. Habitação. Coleção Informação Técnica e Arquitectura, nº5, Lisboa, LNEC, 1999b.
- PEDRO, J. B - Indicadores de Qualidade Arquitectónica Habitacional. Dissertação apresentada em cumprimento das exigências de provas de doutoramento na Faculdade de Arquitectura da Universidade do Porto. Porto, Ed. FAUP, 2000.
- PIAZZALUNGA U, FITZHORN P I, 1998 "Note on a three-dimensional shape grammar interpreter" *Environment and Planning B: Planning and Design* 25 11-33.
- PORTAS, Nuno and GOMES, Ruy – *Estudo das funções e da exigência de áreas da habitação - Necessidades familiares e áreas da habitação. Análise de exigências por funções da habitação. Volume I*. Lisboa, Ed. LNEC, 1964a.

- PORTAS, Nuno and GOMES, Ruy – *Estudo das funções e da exigência de áreas da habitação - Necessidades familiares e áreas da habitação. Análise de exigências por função da habitação. Volume II*. Lisboa, Ed. LNEC, 1964b.
- SHEA K and CAGAN J, 1996, "Innovative Dome Design: Applying Geodesic Patterns with Shape Annealing", *Artificial Intelligence for Engineering Design Analysis, and Manufacturing*
- SHEA K and CAGAN J, 1998, "Generating structural essays from languages of discrete structures" in *Artificial Intelligence in Design '98* Ed J S Gero, F Sudweeks (Kluwer Academic Publishers, Dordrecht) 365-384.
- STINY G, and GIPS J, 1972, "Shape Grammars and the Generative Specification of Painting and Sculpture" in C V Freiman (ed) *Information Processing 71* (Amsterdam: North-Holland) 1460-1465.
- STINY G, 1980a, "Introduction to shape and shape grammars" *Environment and Planning B: Planning and Design* 7 343-351.
- STINY G, 1981, "A note on the description of designs" *Environment and Planning B: Planning and Design* 8 257-267.
- STINY G, 1992, "Weights" *Environment and Planning B: Planning and Design* 19 413-430.
- STINY G, Mitchell W J, 1978a, "The Palladian Grammar" *Environment and Planning B: Planning and Design* 5 5-18.
- TAPIA M A, 1996, *From Shape to Style Shape Grammars: Issues in Representation and Computation, Presentation and Selection*, Ph.D. Dissertation, Department of Computer Science, University of Toronto, Toronto 1996.
- WANG Y, DUARTE J P, (forthcoming) "Automatic fabrication and generation of designs" in T Kvan (ed) *Automation in Construction*, Special issue on Rapid Prototyping, Elsevier.



## 4. Corpus

### 4.1 Introduction

This chapter describes the Malagueira development, namely the corpus of houses used in the development of the shape grammar. Although this work is focused on the development of a grammar for generating Malagueira houses, we also include a brief description of the Malagueira plan. By doing this, we hope to hint that Siza followed systemic approaches to both problems, that these approaches share similarities and, therefore, that it would be possible to enlarge the current grammar to encode the rules of the plan. With this in mind, the description below highlights the design principles followed by Siza in the designs of the plan and the houses.

### 4.2 Plan

#### 4.2.1 Context

The Malagueira Plan was commissioned to Siza by the Évora Town Hall in 1977 to substitute a former plan developed by DGSU (*Direcção Geral da Sistemização Urbanística* – National Commission for the Urban Systematization) in 1975 (Figure 4.1) for a zone which had been classified as Priority Expansion Zone in 1974. The 27 hectares addressed in the Malagueira plan are only part of the area addressed in the DGSU plan mainly because part of this plan had already been implemented. Thus, the DGSU plan considered the settlement of 12,000 people, whereas the Malagueira plan

foresaw the settlement of only 4,120 people with the construction of 1,200 houses. The DGSU plan, in line with the Athens Chart, used functional zoning to create sectors with multifamily housing, and sectors with single-family homes. Siza had to respect the urban indexes of this plan but avoided functional zoning. The DGSU plan had started to be implemented by a developer from Lisbon. The Malagueira plan foresaw a different type of development strongly based on user-participation. Siza resumed a process that had started within the SAAL Program with a group of one hundred families, which had been jeopardized when SAAL was cancelled by the central government. The differences between the two plans mirrored the political conflict between the central and the local governments, which belonged to different sides of the political spectrum, and are a consequence of the power struggle that followed the Portuguese revolution in 1974.

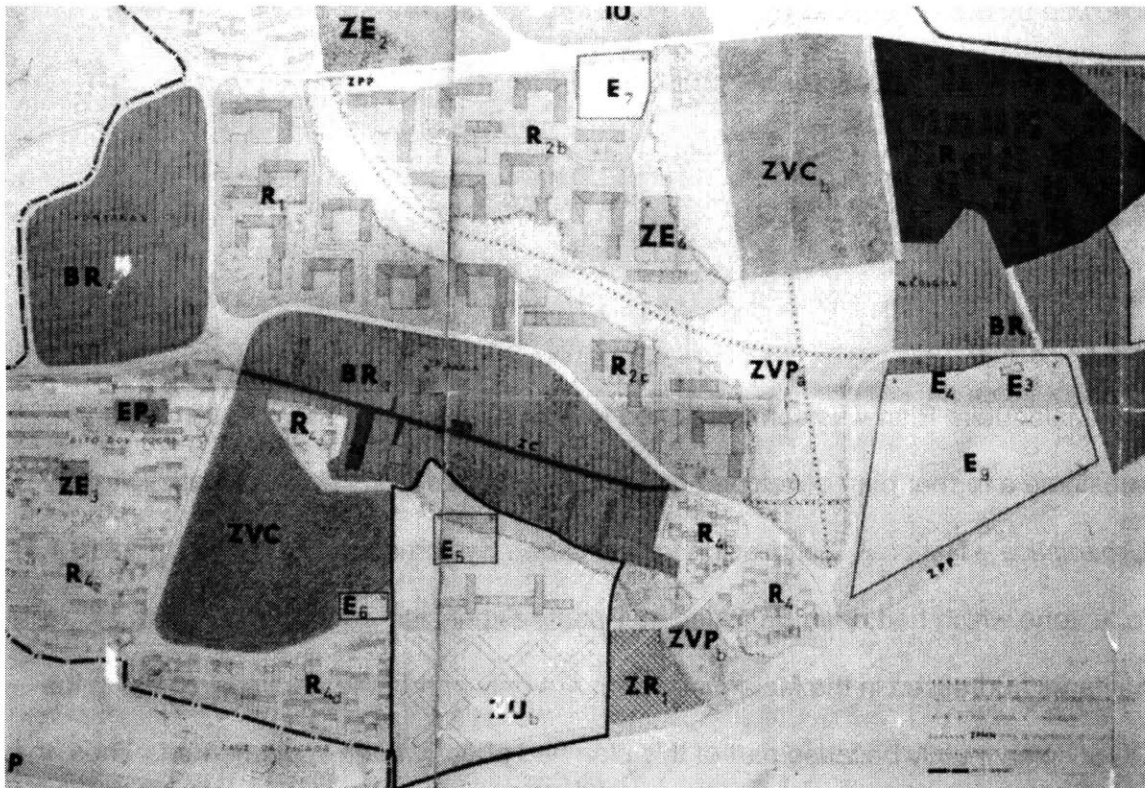
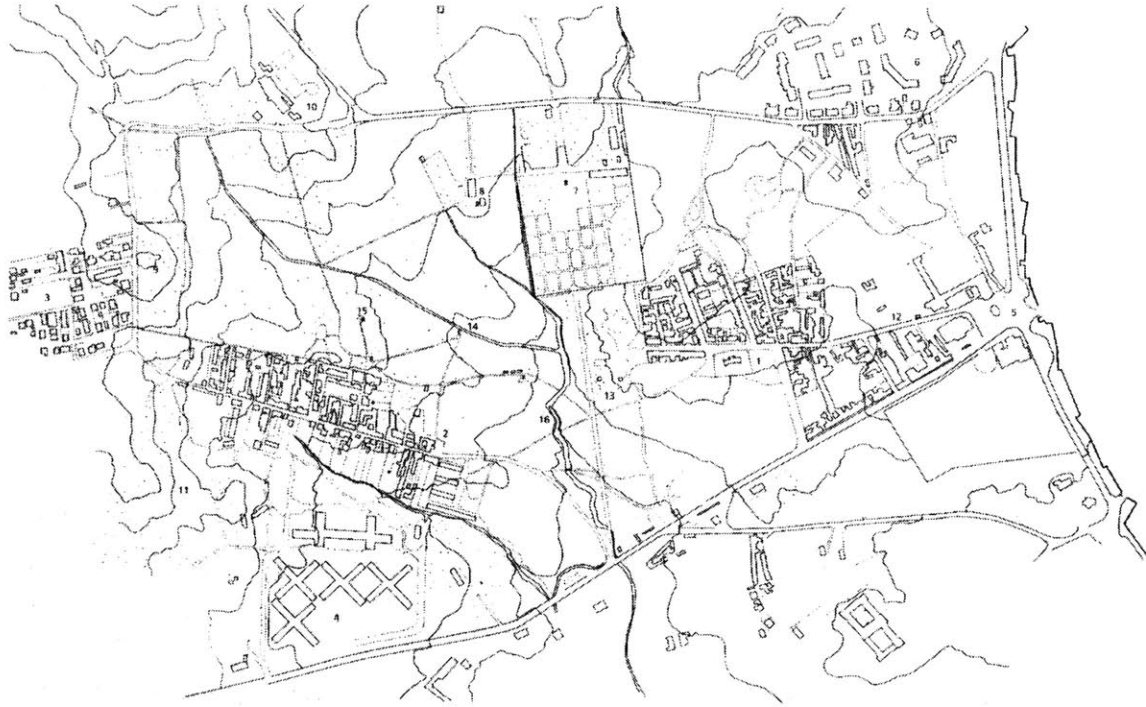


Figure 4.1 – The Évora West Expansion Plan developed by DGSU (Siza's archive)

Évora is a 50,000 people town (35,000 in 1977) located 140km to the west of Lisbon, and it is the most important city of the Alentejo region. Alentejo is a flat, agricultural, and scarcely populated region in the South of Portugal. The climate is characterized by a hot and dry summer (up to 42°C in August.) The city, founded by the Romans on the top of a hill, has a labyrinthine urban grid and its architectural richness led to its classification as World Heritage by UNESCO in 1986. The Malagueira neighborhood is located to the west of the city, on the site of former farms Malagueira and Malagueirinha, which were expropriated in 1974. When Siza started his work, the site was surrounded by a private development of middle-class single-family homes, by a development of prefab houses built for Colonial War veterans, by a social housing development financed by the National Housing Development Fund (FFH) and by three illegal developments (Fontanas, Nossa Senhora da Glória, and Santa Maria.) It was delimited by the national road to Lisbon at the south, and by a local road at the north. On the site, there were only agricultural fields, and rural pathways. (Figure 4.2)



1 - Nossa Senhora da Glória; 2 - Santa Maria; 3 - Fontanas; 4 - Cruz da Picada, FFH; 5 - Portas D'Alconchel; 6 - Vista Alegre; 7 - Malagueira farm; 8 - Malagueirinha farm; 9 - Senhora de Aires farm; 10 - Swimming pool; 11 - Ecurinho forest; 12 - Salesianos street; 13 - Windmills; 14 - Water well; 15 - Alberca; 16 - the Turgela creek

Figure 4.2 – Plan of Malagueira before development (Molteni 1997)

#### 4.2.2 The plan

Two basic principles were at the base of the development of the plan. The first principle was the use of local references as a basis to create an enduring structure:

“Estates limits, small pathways, trees, some rocks, served as a reference to our intervention... [It] departed from the idea recorded in our first visit because I think that the idea is on the site, more than in anyone’s head, for those who care to see.”

(Siza, quoted in L’ Architecture d’Aujourd’hui 1980)

The second principle was an understanding of the city as an organism that grows supported on that structure:

“What is interesting to me in the construction of a city is its capacity of transformation which, to a certain extent, is similar to the growth of a human being. It is born with certain characteristics and a degree of autonomy, a basic structure that can integrate or oppose itself to the changes of life.”

(Siza, quoted in L’ Architecture d’Aujourd’hui 1991)



The plan consisted of a low-rise, dense and continuous residential tissue formed by single-family homes, which permitted to create large green areas, while respecting the urban indexes of the DGSU plan. In addition, it made possible a strong integration with the public buildings, the historic city, the illegal settlements, and the landscape.

Compositionally, the plan is based on a series of functional elements, such as the illegal settlements, two axes, a service duct, three housing sectors, garages, public facilities, and green areas. (Figure 4.3)

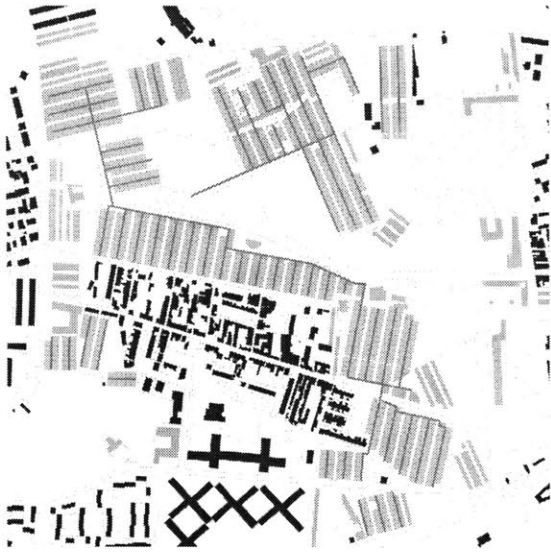
The illegal settlements (Figure 4.4a) serve as a reference for the scale, urban pattern, and building types of the residential tissue. The two axes constitute the main ordering elements of the development. (Fig. 4.4b) The first is an East-west axis that continues the preexisting Salesianos Street, and the second is a North-south axis based on a previous rural pathway that was transformed into a road up to the east-west axis and, then, kept as a pedestrian walkway through the Malagueira farm.

The service duct, a tree-like structure that adjusts itself to the topography, constitutes the backbone of the development. (Figure 4.4c) The duct was inspired in the aqueduct that crosses Évora originating a peculiar morphological system recreated by Siza at Malagueira. It concentrates all the urban infrastructures (water distribution, electricity, gas, telephone, and a collective TV antenna).

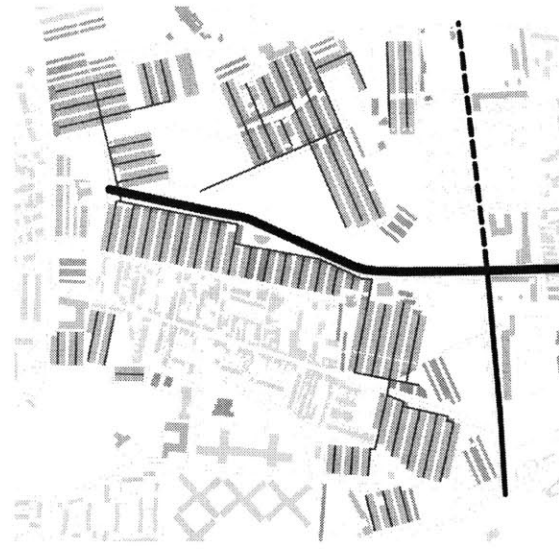
Three main sectors (Figure 4.4d) form the residential tissue. The first sector, at the southwest, extends the spatial pattern of the Santa Maria illegal settlement. The second sector, at the north, was structured upon a diagonal ridge first used as a rural pathway leading to Malagueirinha, and then converted into a shopping arcade. The third sector, to the west, articulates the Fontanas illegal settlement and follows the topography.



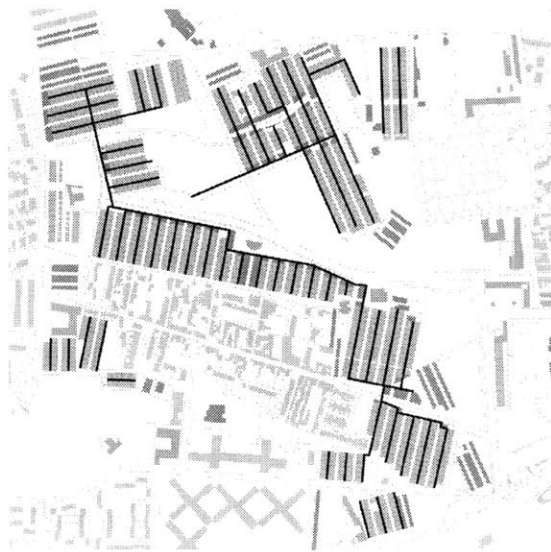
Figure 4.3 - The Malagueira plan.



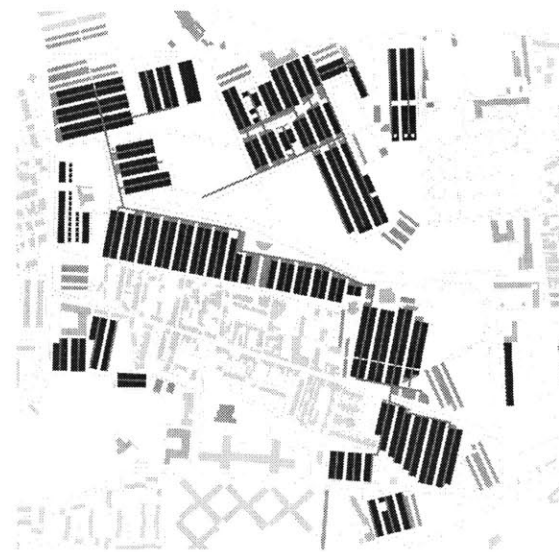
(a) Preexisting housing



(b) Compositional axes

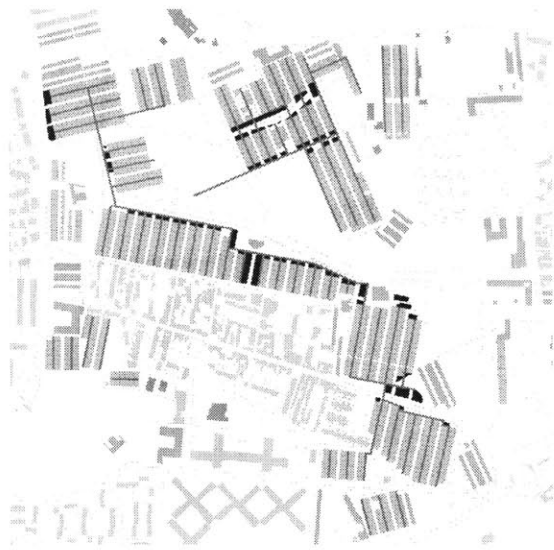


(c) Service duct

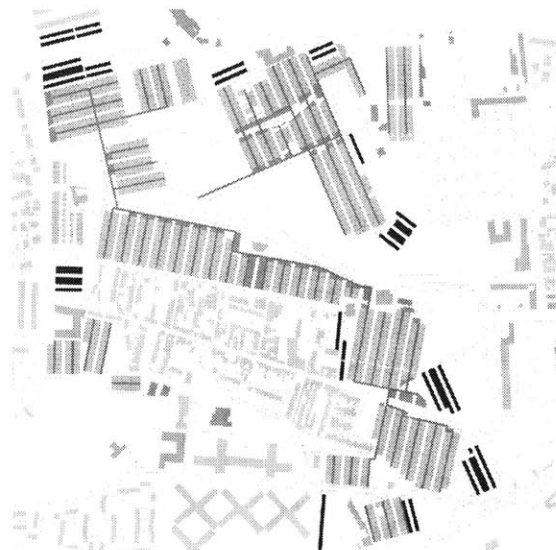


(d) Housing

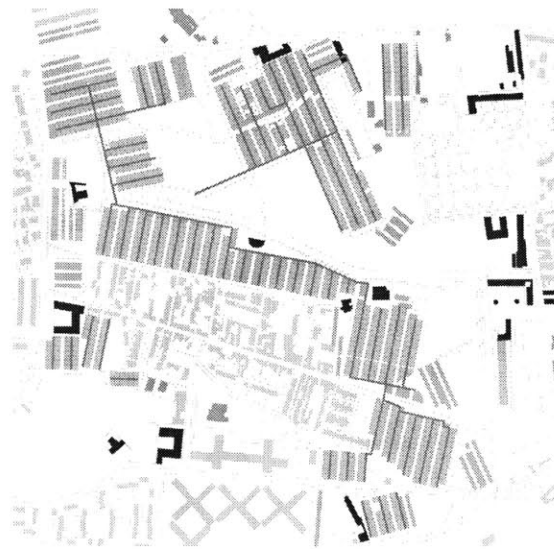
Figure 4.4 - Functional elements of the Malagueira plan.



(f) Commerce



(e) Garages



(g) Public facilities



(h) Green areas

Figure 4.4 (continued) - Functional elements of the Malagueira plan.

The remaining buildings are either in continuity or in contrast with the residential tissue. The shops (Figure 4.4e) are distributed along the service duct, and located on the extremes of the housing blocks. The garages (Figure 4.4f) are segregated from the houses and constitute lineal, compact volumes that differ from the cubist massing of the houses. The public facilities (Figure 4.4g) stand out due to their forms, without disrupting the residential tissue. A dome whose location was determined by the dominating views over the city and the area is the civic and physical center of the neighborhood. The physical center of the neighborhood is a dome designed to be a civic center. The remaining public facilities include buildings designed by Siza (apartment-hotel, language institute, religious center, restaurant, open-air auditorium, and the headquarters of the Boa-vontade cooperative,) buildings designed by Siza's collaborator, Nuno Lopes (orchestra, and the headquarters of the Geraldo-sem-pavor cooperative,) and buildings designed by other authors (post office and supermarket.) Other elements designed by Siza punctuate the urban landscape (bridges, gardens, fountains, stairways, dwells, and benches.)

The high-density of the residential tissue permitted to free large green areas (Figure 4.4h) which articulate the different orientations of the residential sectors, and establish continuity with the rural landscape. The streets are delimited by the continuous surface of the house facades and they are free from the elements that usually obstruct city streets (hoses, sidewalks, stands, etc.) Sheltered from the hot summer light, they are mainly conceived as pedestrian streets, constituting a place to socialize in accordance with the local lifestyle. The design of the street pattern also followed functional constraints:

“The transversal orientation of the streets follows a logic dictated by the water draining problem. There are no pipes, but all the streets follow natural slopes so as to drain the water to a local creek and then to the dam that forms an artificial pond.”

(Siza, quoted in L'Architecture d'Aujourd'hui, 1991)

The variety and unity of the urban landscape is documented in Figures 4.6-19. The spots from which the photos were taken are shown in Figure 4.5.

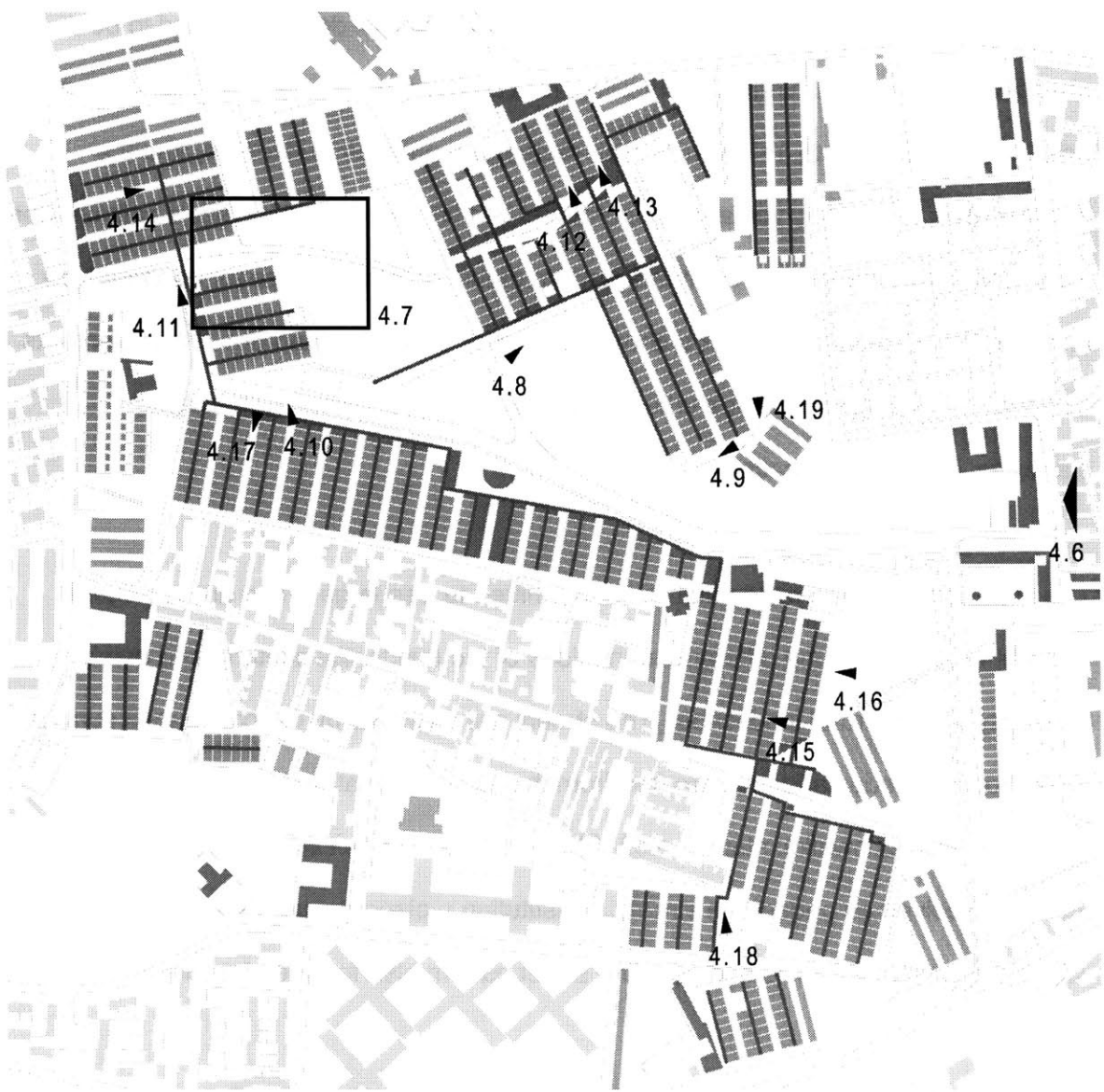


Figure 4.5 - Viewpoints of the photos in Figures 4.6 - 4.19.





Figure 4.6 - Aerial view of the whole development



Figure 4.7 - Aerial view showing courtyard houses in the foreground.



Figure 4.8 - Central green area and northern shopping gallery.

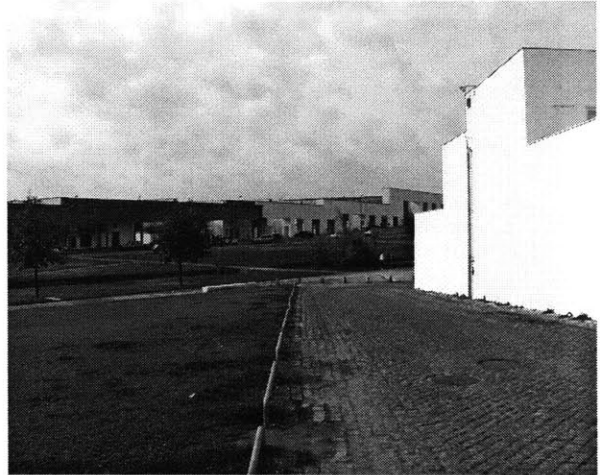


Figure 4.9 - Central green area and southern shopping gallery.

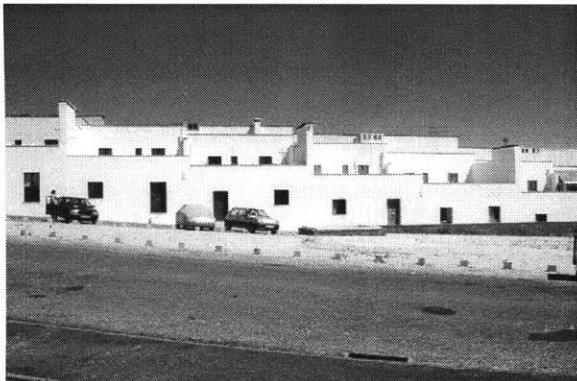


Figure 4.10 - Public housing.



Figure 4.11 - Public housing and the service duct.





Figure 4.12 - Cooperative housing.



Figure 4.13 - Private housing.



Figure 4.14 - The service duct passing over a pedestrian street.

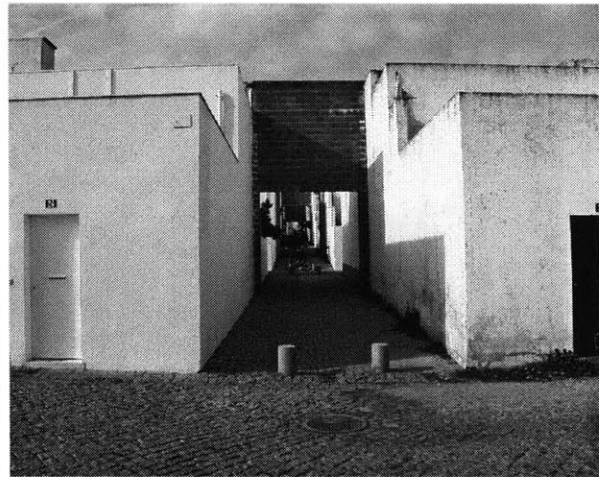


Figure 4.15 - The service duct passing over a pedestrian walkway.



Figure 4.16 - Cooperative housing.



Figure 4.17 - Public housing.



Figure 4.18 - Back alley.

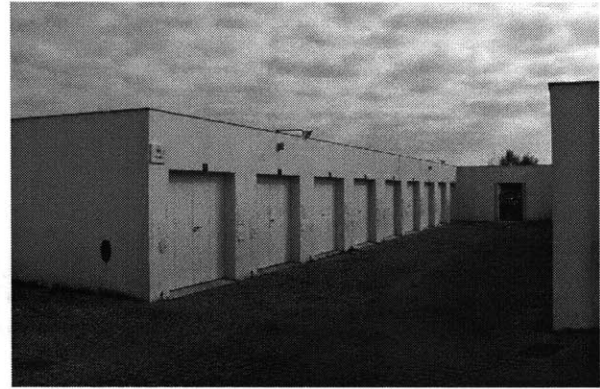


Figure 4.19 - Private garages bordering the residential area.

### **4.2.3 Housing tissue**

The housing tissue's structure and growth is supported on the tree of infrastructures formed by the service duct. The main service duct branches off to create secondary ducts. From both sides of these ducts grow load-bearing walls, forming linear grids of 8 by 12 m lots that constitute the city blocks. Parallel, 6m wide streets separate the blocks with varying length depending on the roads, preexisting elements, or public buildings. The houses can be built independently and expanded to adjust themselves to the needs of their users, permitting the housing tissue to grow by expanding the block or the houses, as documented in Figure 4.20.

Although the lots have the same size in most cases, the housing tissue is formed by a patchwork of different houses. Siza devised a series of housetypes, each of which can have different variations ranging from one up to five bedrooms. These variations can be combined to form varied housing blocks, as illustrated in Figures 4.21-24 for two hypothetical blocks based on Siza's sketches. The variety of the housing tissue in terms of types and variations is represented in Figures 4.25 and 4.26. Such a variety is lower than the one that could be potentially achieved because there was little mixing of types in the same blocks. The decrease in variety was necessary to control complexity with the available design, promoting, and construction means. Siza designed types as time went by for specific promoters who were in charge of building a group of adjacent lots, as explained over the next sections.

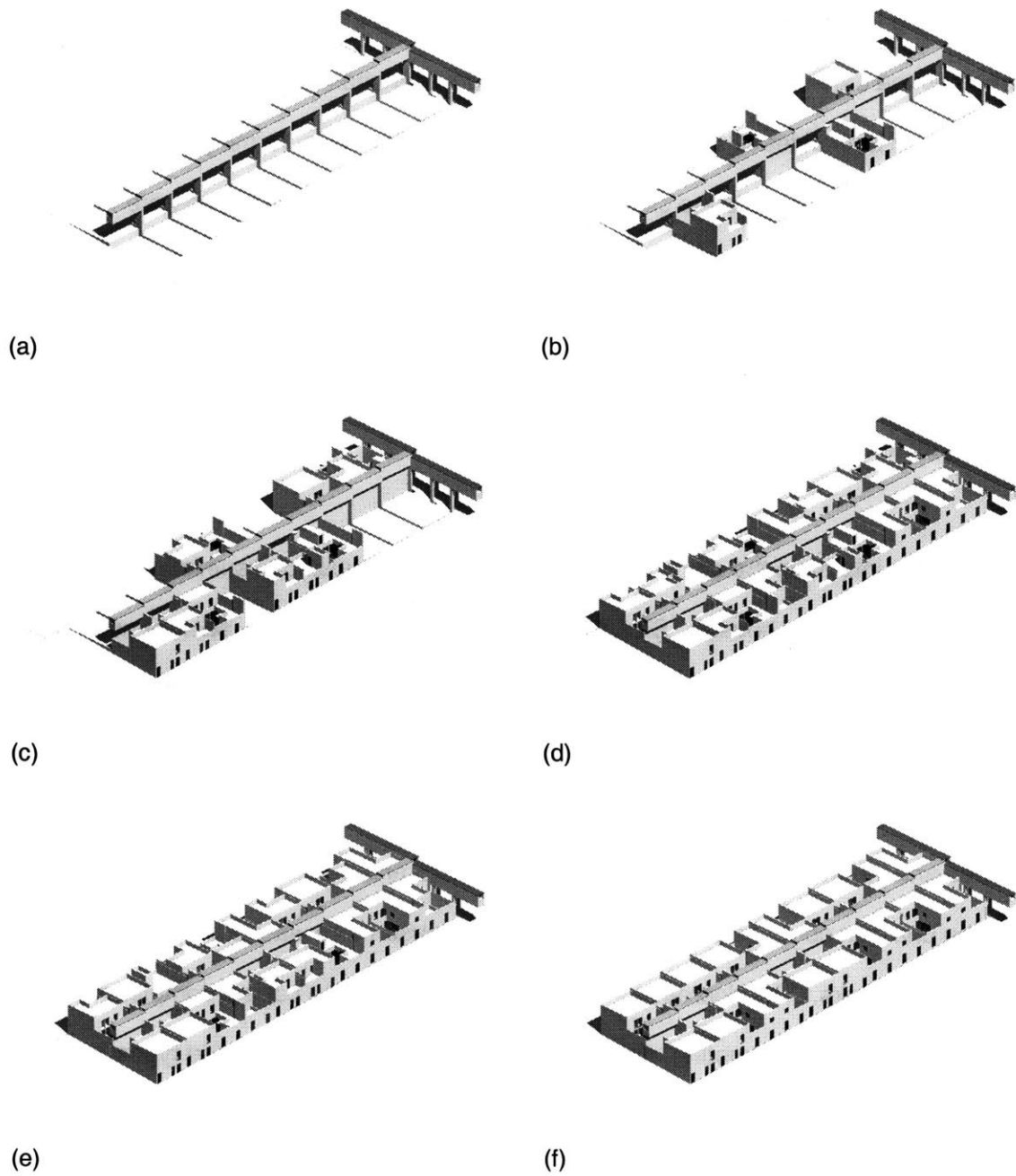


Figure 4.20 - Supported on the tree-like structure of the service duct, the urban tissue can grow first by expanding the block (a-d) and then by expanding the houses (d-f.)

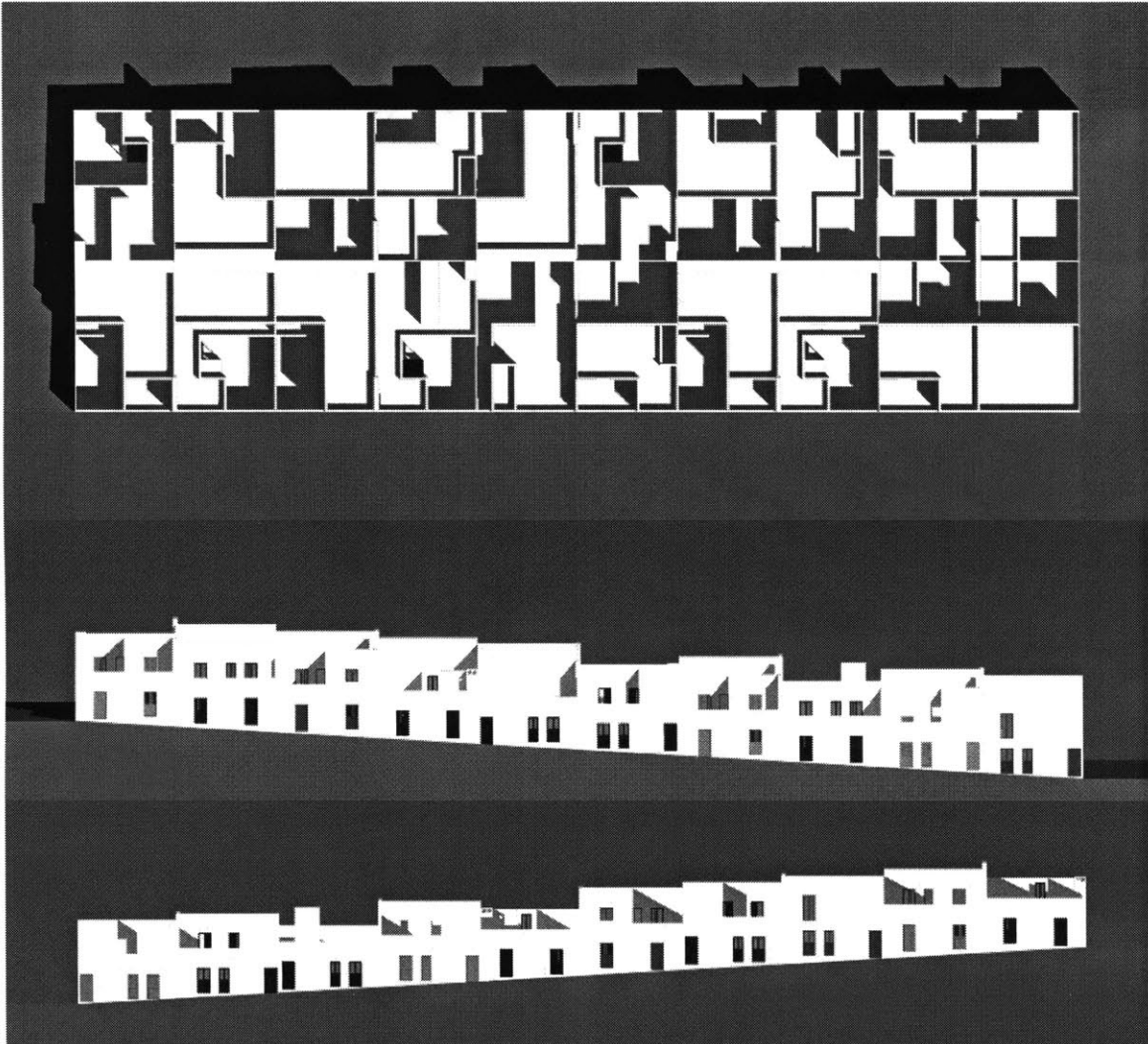


Figure 4.21 - Configuration of a hypothetical city block, in a certain moment in time: plan and elevations.

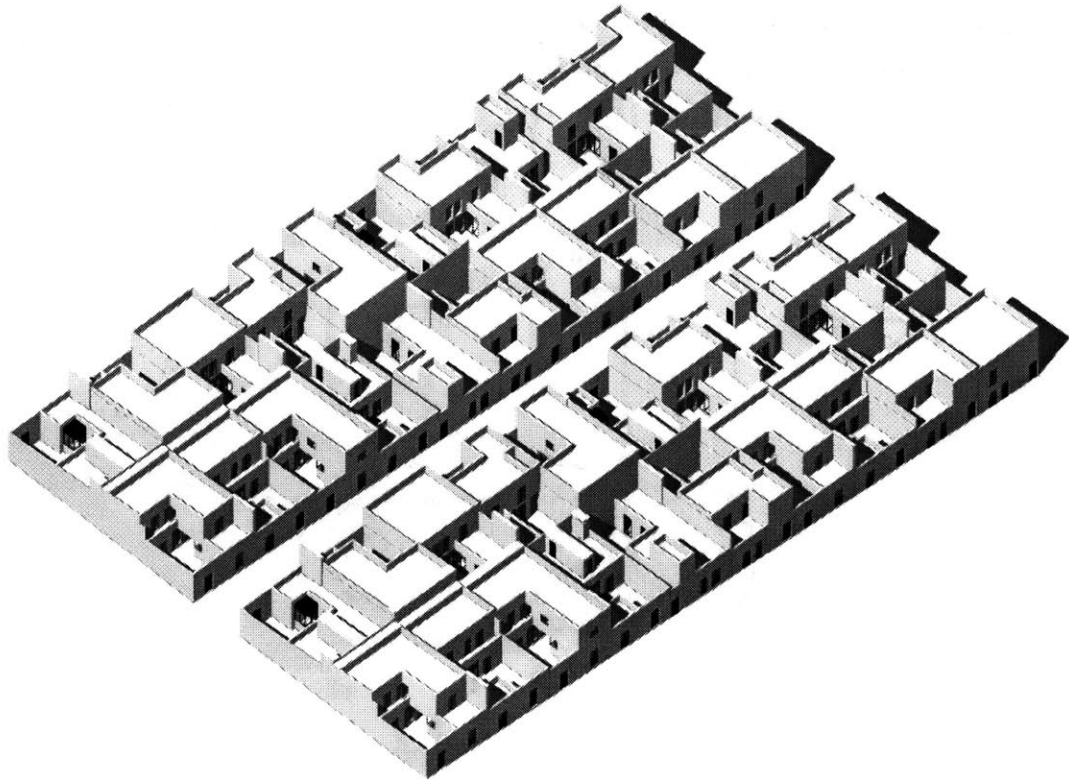


Figure 4.22 - Configuration of a hypothetical city block, in a certain moment in time: aerial view.

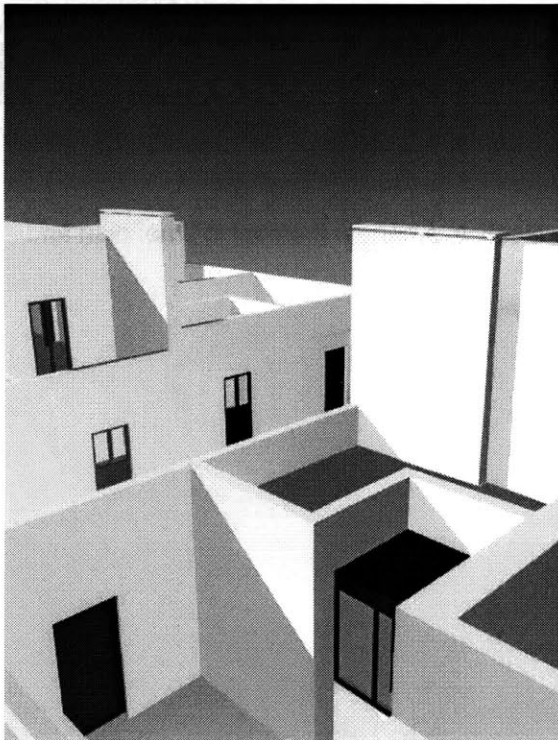


Figure 4.23 - Configuration of a hypothetical city block, in a certain moment in time: terrace view.

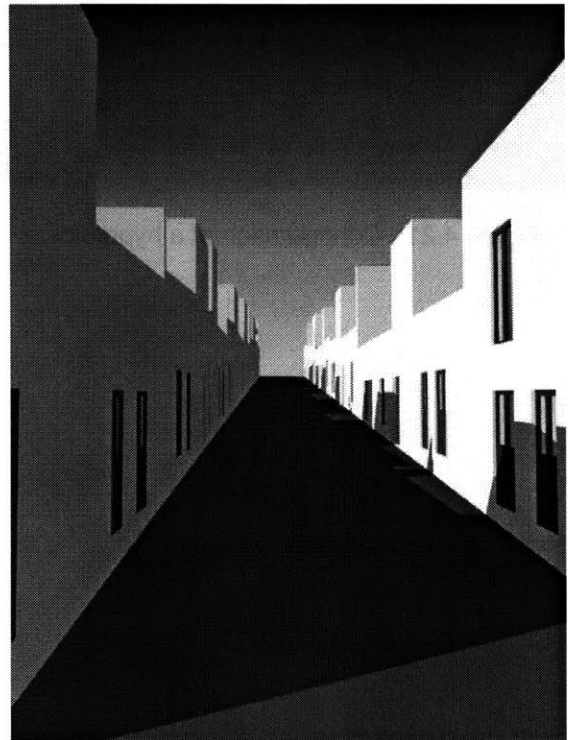
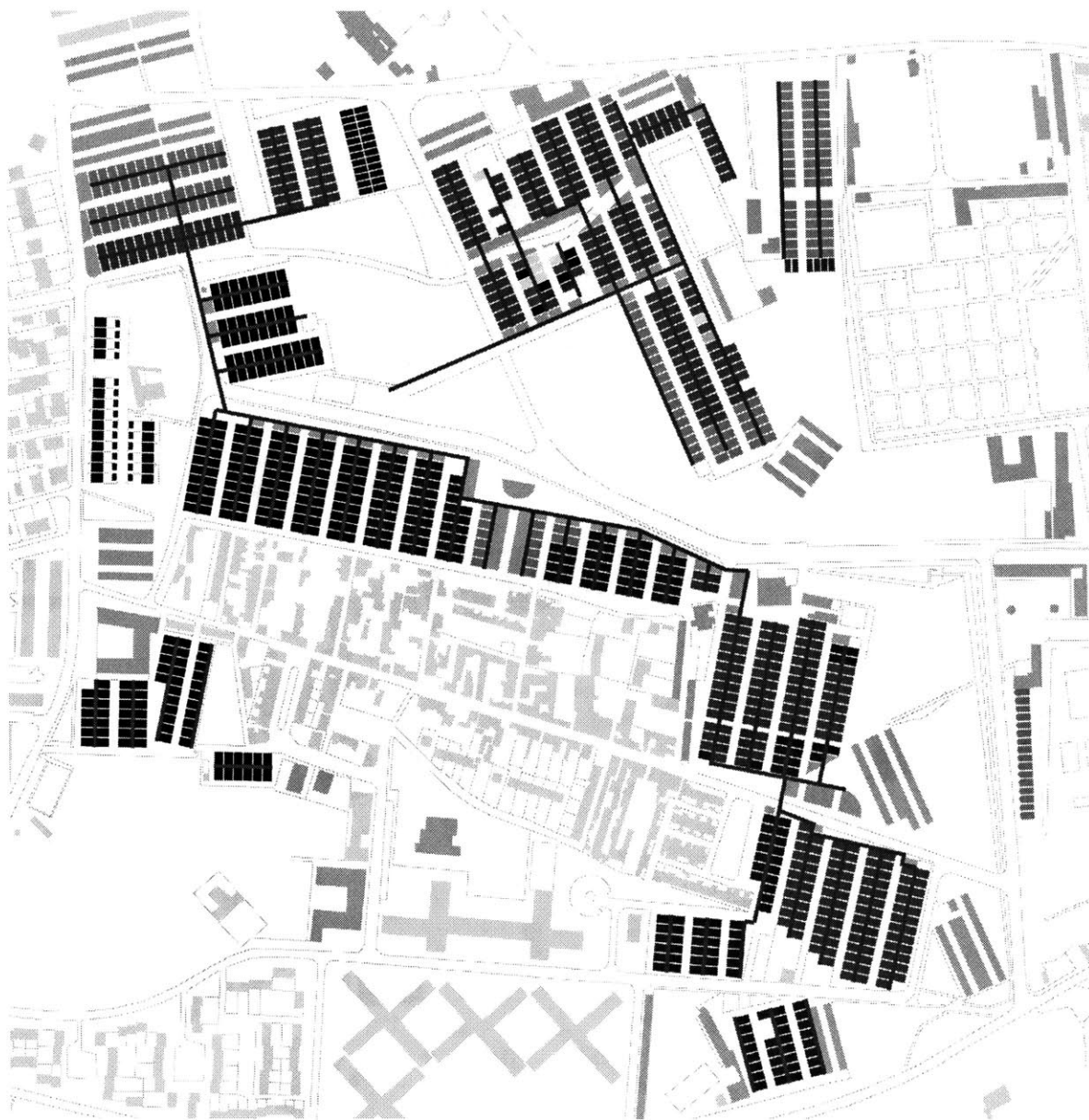


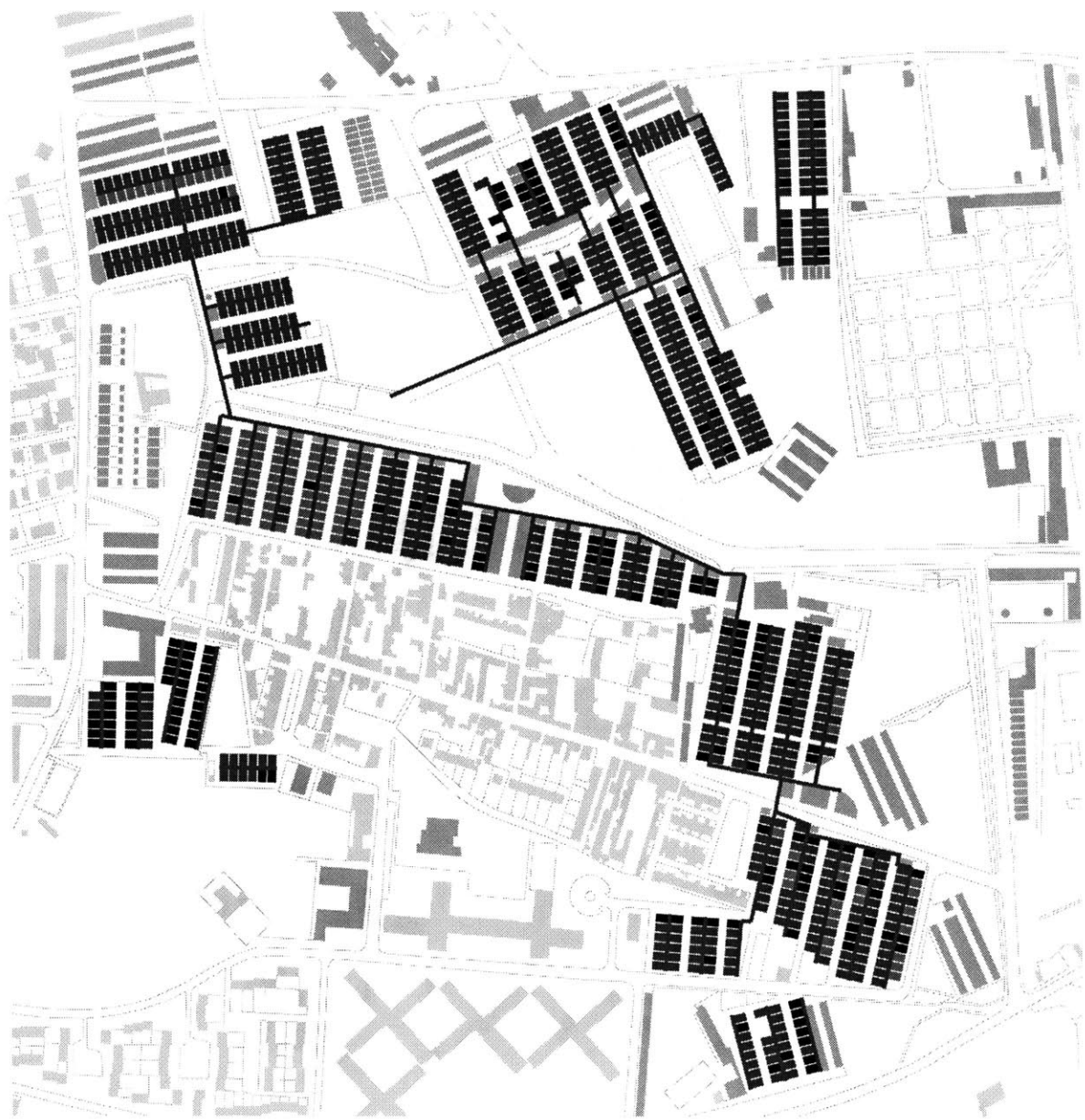
Figure 4.24 - Configuration of a hypothetical city block, in a certain moment in time: street view.





- Ab (8 x 12m) 1977
- Ac (8 x 12m) 1980
- Bb (8 x 12m) 1978
- Ca (8 x 12m) 1985
- Cb (8 x 12m) 198(?)
- D (8 x 12m) 1988
- E (8 x 12m) 1984
- Customized - Siza's office (8 x 12m)
- Customized - Other offices (8 x 12m)
- X (7 x 12m) 1988
- Y (6 x 12m) 1993
- Z (8,5 x 22,3m) 1994
- W (6 x 15m) 1995

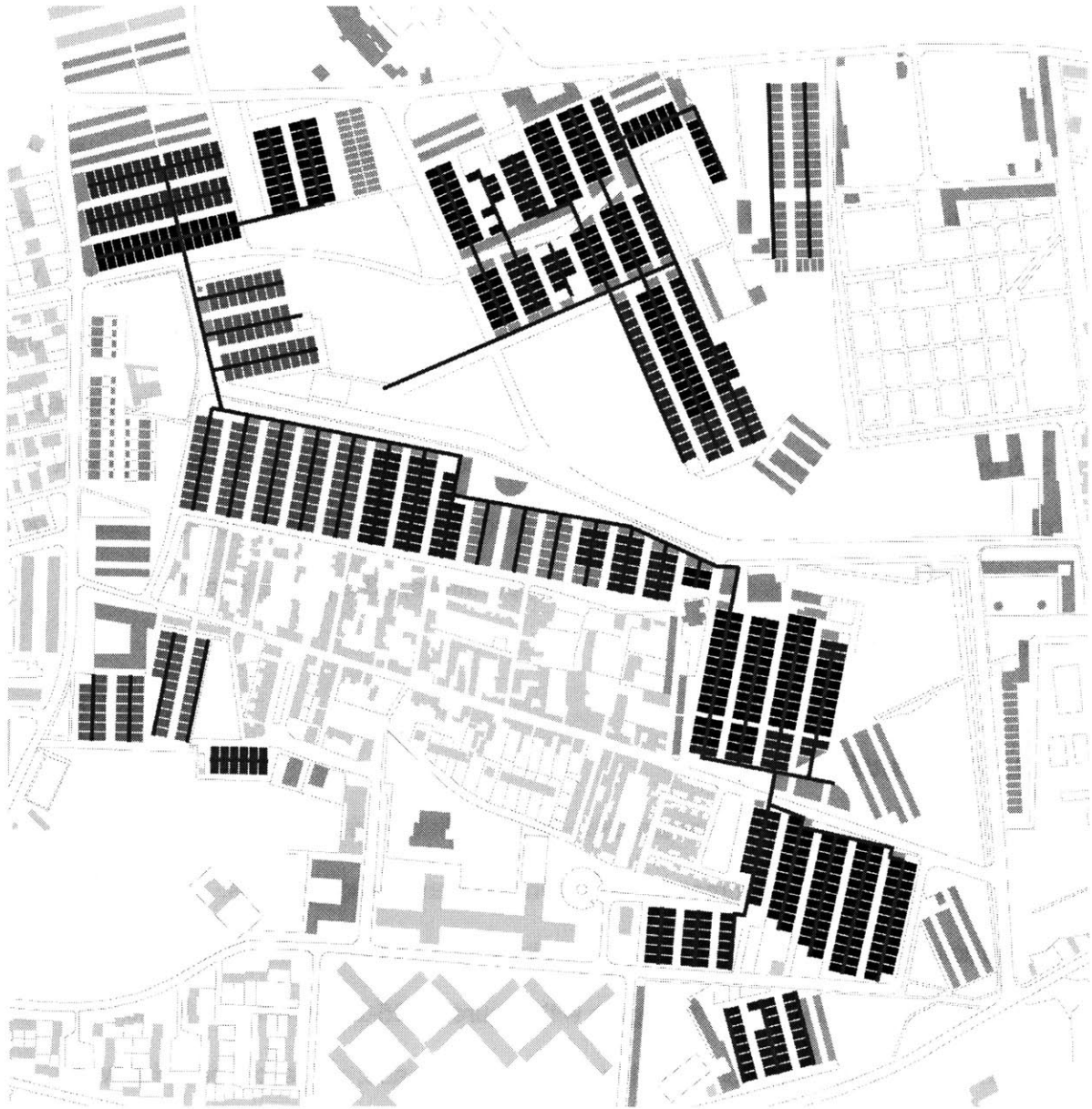
Figure 4.25 – The breakdown of the housing tissue into types. (Please, see Section 4.3 for detailed descriptions of the Malagueira types.)



- T1 and T1+
- T2 and T2+
- T3 and T3+
- T4 and T4+
- T5 and T5+

Figure 4.26 – The breakdown of the housing tissue into variations. (Please, see Section 4.3 for detailed descriptions of the Malagueira types variations.)





- Cooperative Boa Vontade - 1st phase (1978)
- Cooperative Boa Vontade - 2nd phase (1985)
- Cooperative Boa Vontade - 3rd phase (1988)
- Cooperative Giraldo - 1st phase (1979)
- Cooperative Giraldo - 2nd phase (1980/86)
- Cooperative Giraldo - other phases (1987/98)
- Public - FFH / IGAPHE - 1st phase (1980)
- Public - FFH / IGAPHE - 2nd phase (1981)
- Private (1978/1998)

Figure 4.23 - Breakdown of the housing tissue into promotion schemes.

#### **4.2.4 Promotion**

The variety of the housing tissue at the physical level is mirrored in the kinds of promotion that were used to build Malagueira. (Figure 4.27) Siza started to develop the plan with a commission of one hundred future inhabitants (Associação de Vizinhos São Sebastião) which had been constituted in the spirit of the SAAL program (1974-1976.) These people were supposed to build their houses by self-construction.<sup>1</sup> In addition, the initial plan considered the following types of promotion: 407 dwellings by cooperatives, 300 by the public institute (FFH,) 300 by private companies, and 93 by development contracts.<sup>2</sup> (Siza, 1979) The cooperatives were, by far, the most successful group and, the final breakdown was as follows: 60% cooperatives, 33% public, and 7% private. (Molteni, 1997)

#### **4.2.5 Construction phases**

The strategy followed in the construction was to split the development into sectors, assign them to different promoters, and then proceed through phases. The pace of construction, reconstructed after the analysis of aerial photographs, is depicted in Figure 4.28. A close comparison of this figure with Figures 4.25-27 permits the identification of the housetypes and the promoters involved in each phase, as described in following. In 1978, the Boa Vontade cooperative constructed the northern part of the south sector and a small block near the Santa Maria illegal development (1<sup>st</sup> phase.) In 1979, the Giraldo cooperative built the middle part of the south sector (1<sup>st</sup> phase.) And in 1980, the FFH built in the north and west sectors (1<sup>st</sup> phase.) The houses in these first phases were type Ab houses, except for a few type B houses. Still in 1980, the Giraldo cooperative built the first type C houses in the south sector (2<sup>nd</sup> phase,) and in the following year, the

---

<sup>1</sup> This commission later became the Geraldo-sem-pavor cooperative.

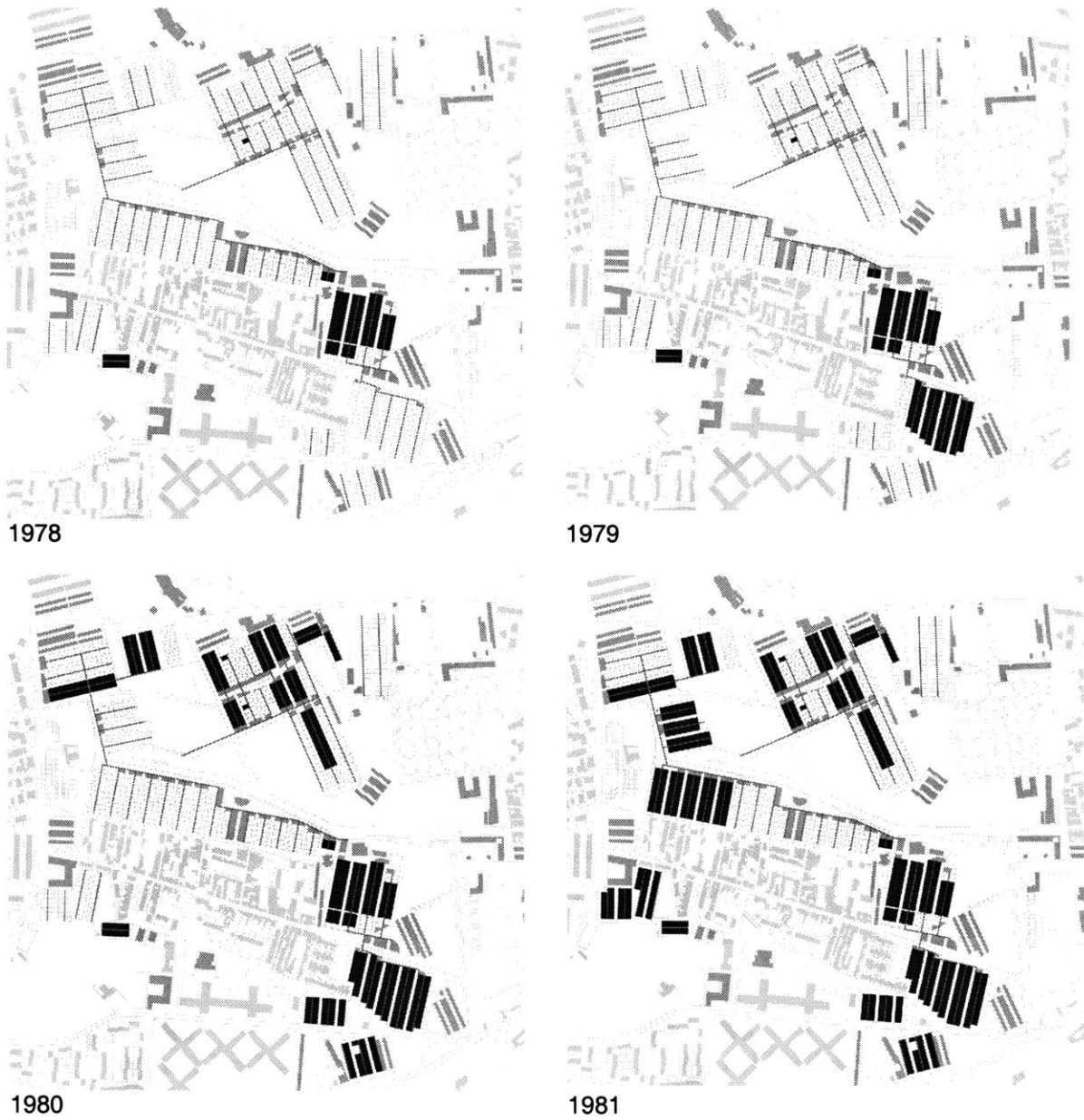
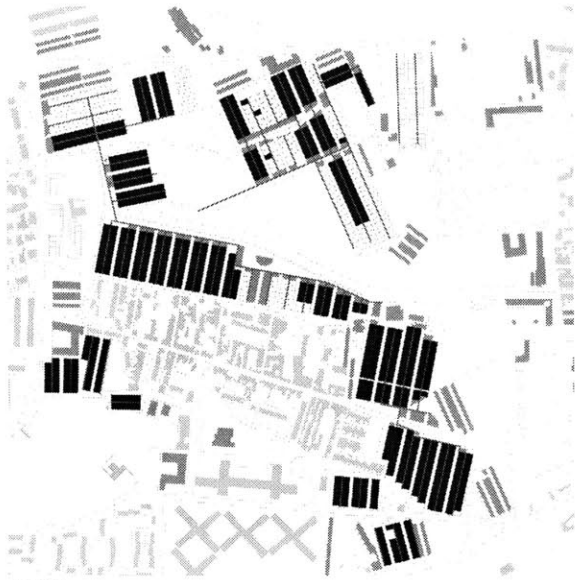
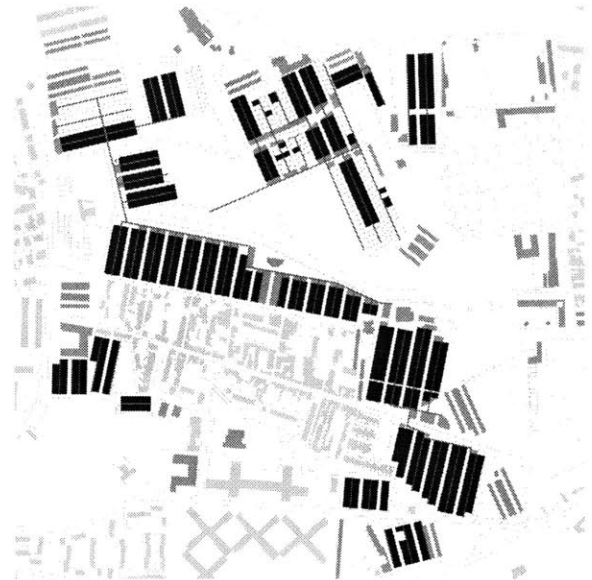


Figure 4.28 - Housing construction phases.

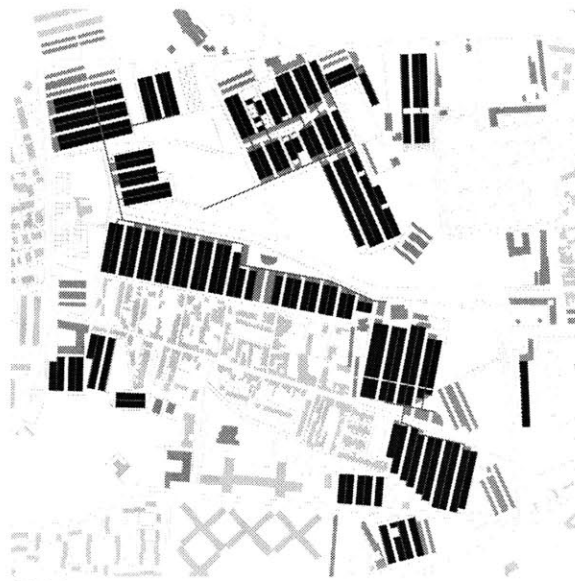
<sup>2</sup> The urban development contract was an agreement that allowed the town hall to promote the development of housing with central government funds.



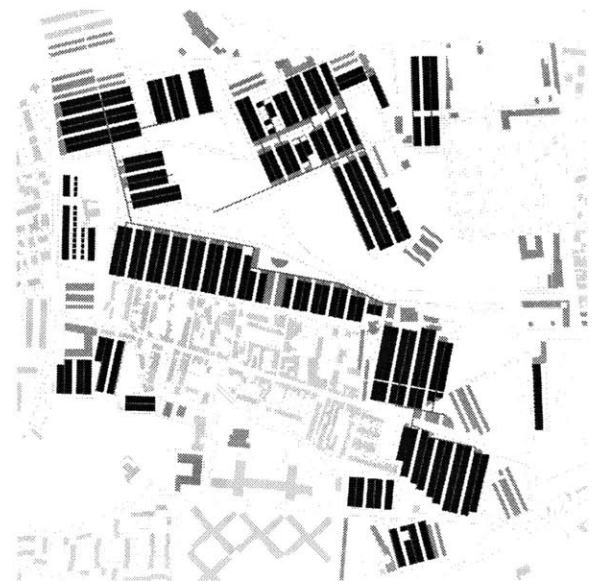
1985



1988



1996



1998

Figure 4.28 (continued) - Housing construction phases.

FFH built type Ac houses in the west and south sectors (2<sup>nd</sup> phase.) In 1982-1985, the Boa Vontade cooperative built the remaining type C houses in the south and north sectors (2<sup>nd</sup> phase,) and the first type D houses in the south sector in 1986-1988 (3<sup>rd</sup> phase.) In 1989-1996, the Giraldo cooperative built other type D houses in the west and north sectors (3<sup>rd</sup> phase.) Since then, both cooperatives have built in a much smaller scale, other types designed by Siza. In the meanwhile, the construction of houses by private promoters in the north sector has occurred since 1978. In 2000, the development was near completion as only type Z houses and a few of the private houses, as well as some shops, the dome, the language institute, and the apart-hotel remained unbuilt.



Figure 4.29 - Illegal housing at Malagueira.



Figure 4.30 - Vernacular housing of Alentejo.

## 4.3 Housing types

### 4.3.1 Design schemes

Several authors have pointed out that Siza looked at the housing types of the illegal developments (Figure 4.29) and the vernacular types of Alentejo (Figure 4.30) in the

design of the Malagueira houses. (Testa, 1984; Fleck, 1992; Molteni, 1997) Siza explains that this influence was appropriate because such types are the result of a long evolution process of adjustment to the environment. The influence is both formal and functional. The domestic space is organized into small interior rooms around a large patio, denoting a lifestyle centered in the outside. Small openings and whitewashed surfaces protect from the strong sunlight, and big chimneys create a powerful plastic effect. However, Siza interpreted the illegal and vernacular types to create a set of rules that permitted to adjust the designs to the needs of modern life and to different lifestyles. We have classified the variety of designs into schemes, general types, types, subtypes, and variations, depending on the degree of generality. Scheme is the most general category, and general type, type, subtype, and variation are increasingly more specific.

Siza developed five basic housing schemes. The first scheme was devised as early as 1977, and it includes several housetypes designed since then. It accounts for the majority of houses built at Malagueira and it is the basic construction unit of the urban tissue. The remaining schemes (Figures 4.31-34) can be considered special cases. They were designed much later, they include a single housetype each, and few houses were built. We named them as X, Y, Z, and W, and they were designed respectively in 1988, 1993, 1994, and 1998. Preliminary analysis revealed important differences in the layout between the first and the latter schemes, such as the lot size, the formal structure, and the functional organization. The size of the lot in the first scheme is 8 x 12 m, whereas in the latter schemes it is 7 x 12, 6 x 12, 8.5 x 22.30; and 6 x 15 m, respectively. Consequently, the houses were placed transversally on the wider lot of the first scheme, and longitudinally on the others. Then, in the first scheme the patio determines the functional organization -- the "L" shape of the house surrounds the patio, whereas in the latter schemes, it is either absent, smaller, or it seems to be less

determinant -- the patio is simply added to the house. Because the first scheme accounts for the majority of the Malagueira houses, and because there are significant differences in the layout relatively to the latter schemes, the shape grammar was developed taking into account only the first scheme. We believe, nevertheless, that it is possible to extend the shape grammar to incorporate the latter schemes.

In the first scheme, the houses are expandable and can have up to two floors and five bedrooms. The ground floor remains constant within each type, and it has a patio around which three functional zones, living, sleeping, and services are laid out to form an "L" shape. The patio increases the available lighting surface and secures a certain degree of comfort due to the microclimate created by walls, trees, and pergolas. The inner distribution of rooms is rationalized in response to circulation and lighting demands. The bathrooms are overlaid on the top of each other to facilitate water distribution and drainage. The openings, especially those facing the street, are placed regularly on the facades. The richness of street facades is due to adjustments prompted by the topography, or to different configurations of the houses.

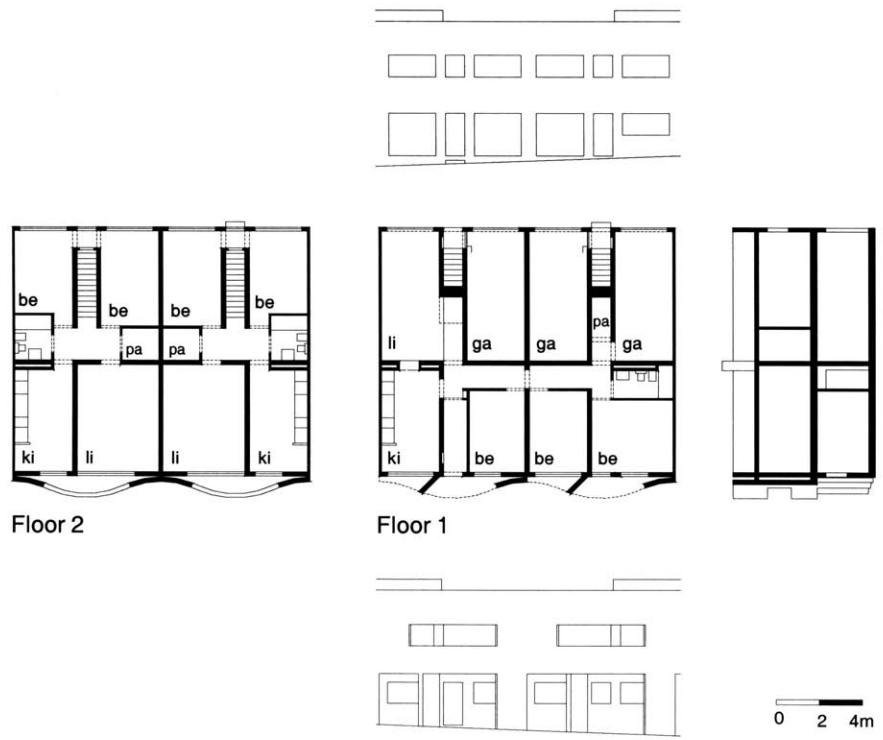


Figure 4.31 - Type X, 1988. (Not considered in the corpus.)

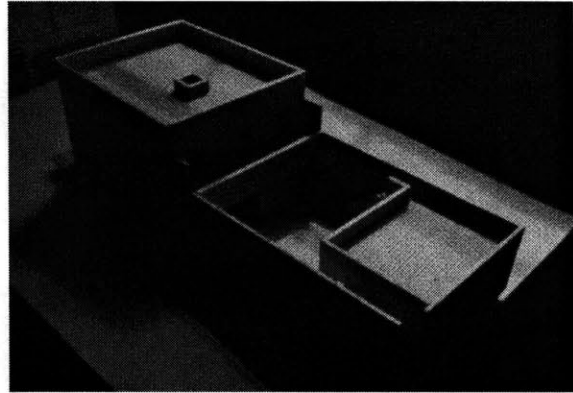




Figure 4.32 - Type Y, 1994. (Not considered in the corpus.)



Site



Physical model

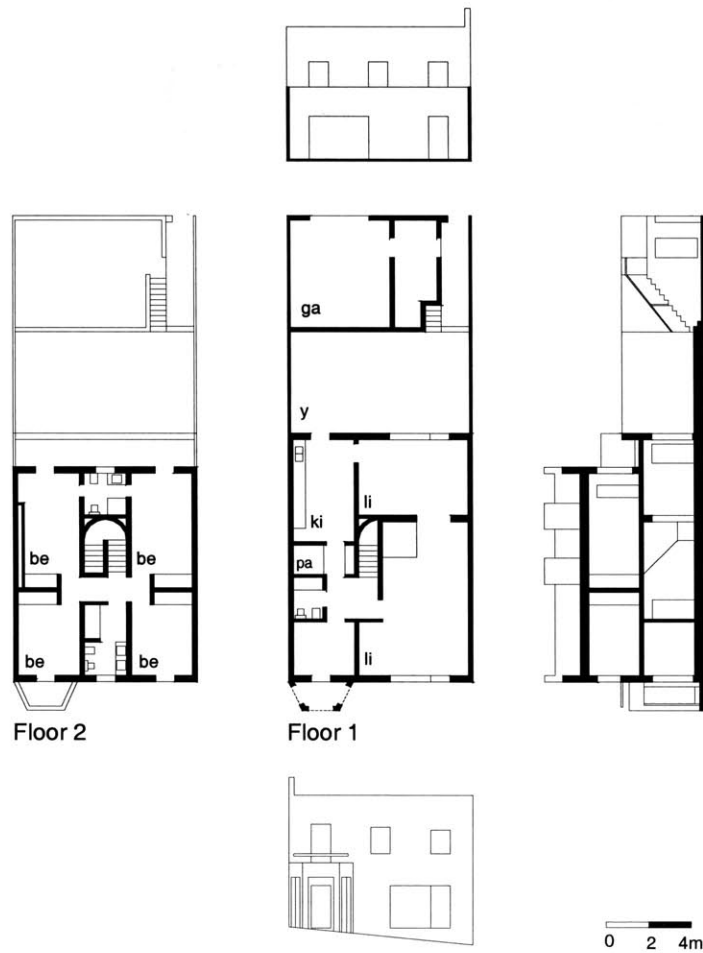


Figure 4.33 - Type Z, 1994. (Not considered in the corpus.)

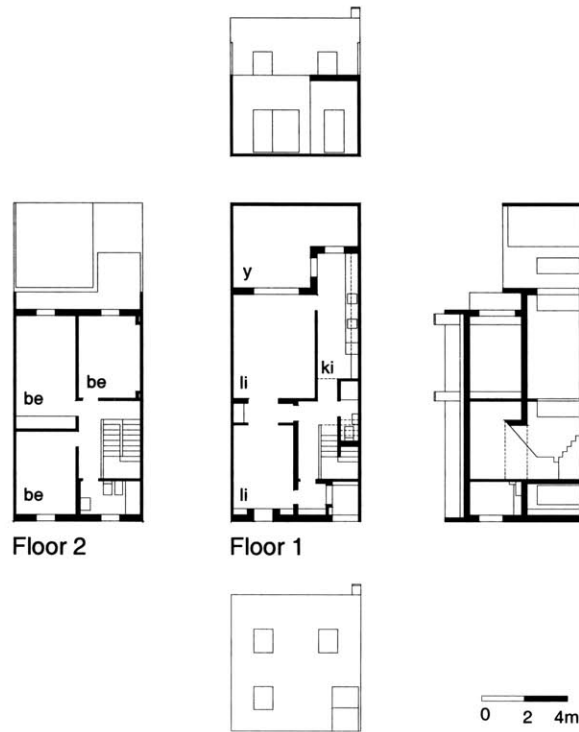
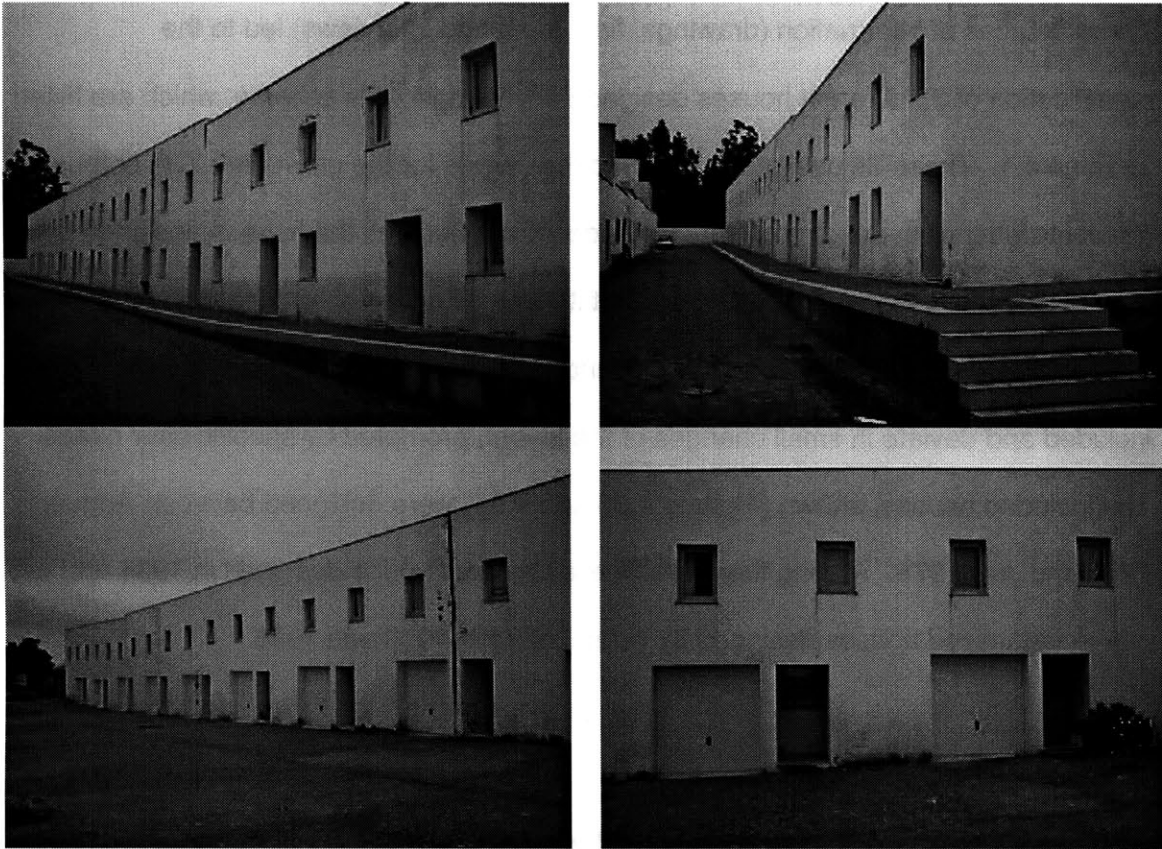


Figure 4.34 - Type W, 1997. (Not considered in the corpus.)

Three sources of information (drawings, field trips, and interviews) led to the identification of 35 different houses designed according to this scheme, which are listed in Table 4.1. These 35 designs constituted the corpus for the grammar. The corpus is not comprehensive, but almost; that is, it does not include all the houses designed by Siza or Nuno Lopes within the scheme, but it includes all the different housetypes. The houses that were left out have the same functional organization as those that were included and deviate in small changes of the layout, prompted by specific user needs. The included houses, shown (Figures 4.35 and 4.36) were designed between August 1977 and July 1976. Among them are Siza's personal house designed in 1984 and two other customized houses designed by Nuno Lopes in 1980 and 1996.

**Table 4.1 – Houses designed by Siza and Lopes according to the 1<sup>st</sup> Malagueira scheme**

Family	Type	Subtype	Variation	Number	Date (1)	Scale	Designer
Front Yard	A	Aa1	t1, 2, 3, 4, 5	1-5	Aug 77	1/100	Siza
		Aa2	t1, 2, 3, 4, 5	(2)	Nov 77	1/100	Siza
		Ab1	t3	6	Jan 78	1/100	Siza
		Ab2	t2, 3, 4, 5	7-10	May 78	1/100, 1/50, 1/20	Siza
		Ac	t2, 3, 4, 5	11-14	Jan 80	1/100, 1/50	Siza
		Ad	t4	16	Jan 96	1/100	Lopes
	C	Ca	t2, 3, 4, 5	17-20	Jan 85	1/100, 1/50	Siza
		Cb	t1, t3	21-22	(3)	1/100	Siza
	D	Da	t2, 3, 4a, 4b, 5 (4)	23-26	Dec 88	1/100	Siza
		Db	t2, 3	27-28	May 95	1/100	Siza
	E	Siza house	t2	29	Jan 84	1/100	Siza
		Siza house (final)	t2		Mar 84	1/100, 1/50	Siza
	Back yard	B	Ba1	t1, 2, 3, 4, 5	30-34	Aug 77	1/100
Ba2			t1, 2, 3, 4, 5	(2)	Nov 77	1/100	Siza
Bb			t2, 3, 4, 5	35	Feb 78	1/100, 1/50, 1/20	Siza

**Notes**

- (1) The dates are those of the earlier drawings found.
- (2) Types Aa2 and Ba2 were not counting towards the total number of types because the only difference from types Aa1 and Ba1 is an additional window.
- (3) It is not certain that type Cb was built because only sketches were found.
- (4) Variations Da t4a, Da t4b, and Da t5 were identified during fieldtrips and documented with slides and sketches.

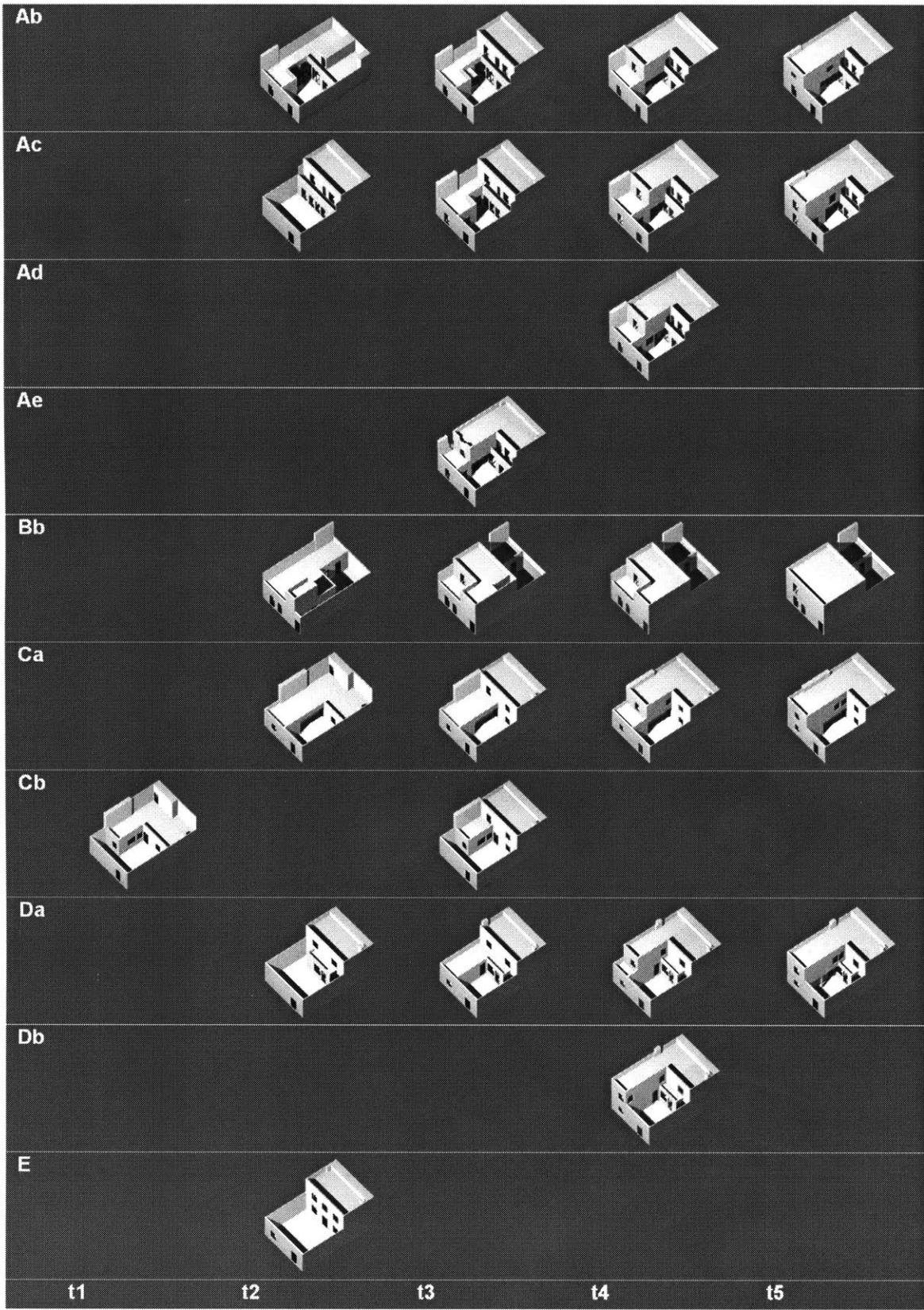
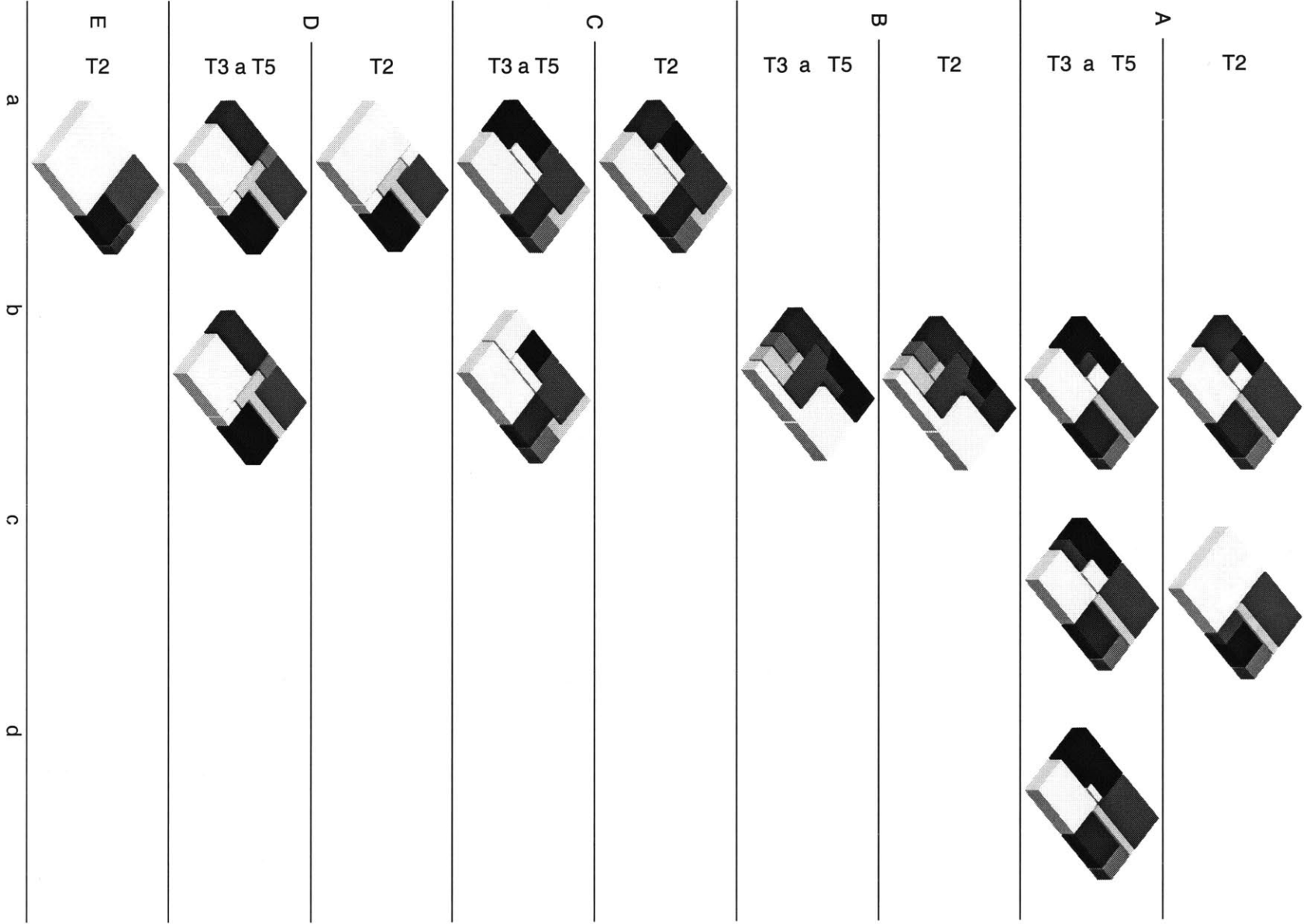


Figure 4.35 - Main types and variations considered in the corpus.





- Bedrooms
- Bathroom
- Living room
- Pantry
- Kitchen
- Transitional room
- Laundry
- Circulation
- Yard

Figure 4.36 - First floor functional organization of the main types and variations in the corpus.

The corpus of houses can be classified into two general types, depending on whether the yard is at the front, as in most cases, or at the back. The smaller number of backyard houses was simply the result of the lack of demand. The corpus can be further subdivided into five basic types: here called A, B, C, D, and E.<sup>3</sup> Types A and B were the first types to be designed and were named as such by Siza. Type B is the only backyard type. Types A, C, D, and E all have a front patio and differ from each other by the combined effect of the location of the four basic functional zones within the layout, and the location of the staircase within one of the interior zones. In types A and D the staircase is placed against the wall between the living room and the adjacent space, a bedroom in type A, and the kitchen in type D. In types C and E, the stairs are against the back wall, and they differ from each other in the location of the kitchen, which is at the front of the lot in type C and at the back in type E.

---

<sup>3</sup> Some authors prefer to refer to types C, D, and E, as variations of type A. The term type in their terminology means general type in the terminology followed in this work.

All these basic types, except for E, include subtypes that differ from one another in details of the layout. Such differences are due to different divisions of the functional zones into specialized rooms, and are denoted in the name by a lowercase letter placed after the letter that identifies the type. For instance, type A has four subtypes, named Aa, Ab, Ac, and Ad. The first subtype, Aa, corresponds to the preliminary study of Ab (as Ba is the preliminary study of Bb), and was never built. Ab differs from Ac in the laundry location within the service area. Finally, Ad, a customized subtype designed by Lopes differs from Ab in the placement of the walls that divide the service area. The list of all the functional zones and rooms in the Malagueira houses are listed in Table 4.2, which also includes the abbreviations that identify them in Figures 4.37-42.

**Table 4.2 – List of rooms found in the corpus houses in the corpus**

Floor	Symbol	Functional Zone	Symbol	Rooms
1 <sup>st</sup> floor	pa	Patio		
	li	Living zone	lr	Living-room
			cl	Closet
	se	Service zone	ki	Kitchen
			la	Laundry
			pa	Pantry
			ts	Transitional space
	sl	Sleeping zone	be	Bedroom
			ba	Bathroom
			ci	Circulation
		st	Stairs (1)	
		co	Corridor to backyard (2)	
2 <sup>nd</sup> floor			be	Bedroom
			ba	Bathroom
			cl	Closet
			ci	Circulation
			st	Stairs
			te	Terrace

Notes:

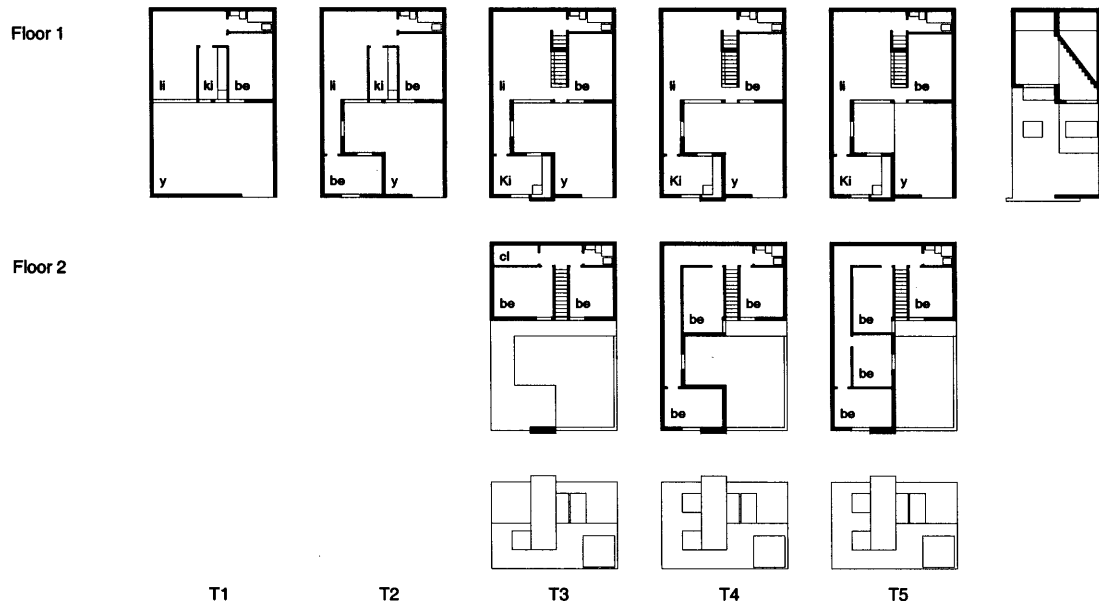
- (1) The stairs can be included in any of the interior zones
- (2) The corridor to the backyard does not belong to any of the zones



The majority of subtypes have several variations, each with a specific number of bedrooms ranging from one to five. Each variation is expandable as it can evolve from a single up to five bedrooms. Such variations are named  $t_n$ , where  $n$  indicates the number of bedrooms. No  $t_1$  were built due to a lack of demand, and  $t_1$  designs only appear in preliminary designs of subtypes Aa, Ba, and Cb. Nevertheless, it would be straightforward to derive  $t_1$  variations for the remaining subtypes, by eliminating one bedroom from the respective  $t_2$  variations. All the remaining subtypes, except E, have  $t_2$  through  $t_5$  variations. Type E only has a  $t_2$  variation, the one that Siza designed for himself, but it would be possible to design the remaining variations by adding more bedrooms.

Figures 4.37-42 show the plans of houses in the corpus, Figures 4.43-51 show the corresponding photos, and Figures 4.52-61 the 3D digital models. The general design principles of the scheme considered in the corpus are summarized in Table 4.3. This table was part of the building regulations of the Malagueira plan that existed in the town hall, which also included the projects of the housing types. They were to be followed by all the promoters that intended to participate in the development. The private promoters had the possibility of choosing from the available types, fully respecting the designs, or to ask a designer to adjust or to design a new house, following the design principles set by Siza. Some gave the commission to Siza, others preferred Nuno Lopes, and others resorted to other designers. In summary, Siza laid out a design game, based on a set of rules, that permitted to adjust the design of the houses to different site, promotion and user constraints.

Type Aa 1977



Type Ab 1978



Figure 4.37 - Plans, sections, and elevation of types Aa, and Ab (Siza.)

Type Ac 1980



Figure 4.38 - Plans, sections, and elevation of types Ac (Siza,) and Ad and Ae (Lopes.)

Type Ba 1977



Type Bb 1978



Figure 4.39 - Plans, sections, and elevation of types Ba and Bb (Siza.)

Type Ca 1985



Type Cb



Figure 4.40 - Plans, sections, and elevation of types Ca and Cb (Siza.)

Type Da 1988



Type Db 1995



Figure 4.41 - Plans, sections, and elevation of types Da and Db (Siza.)

Type E 1984 (AlvaroSiza House)

---

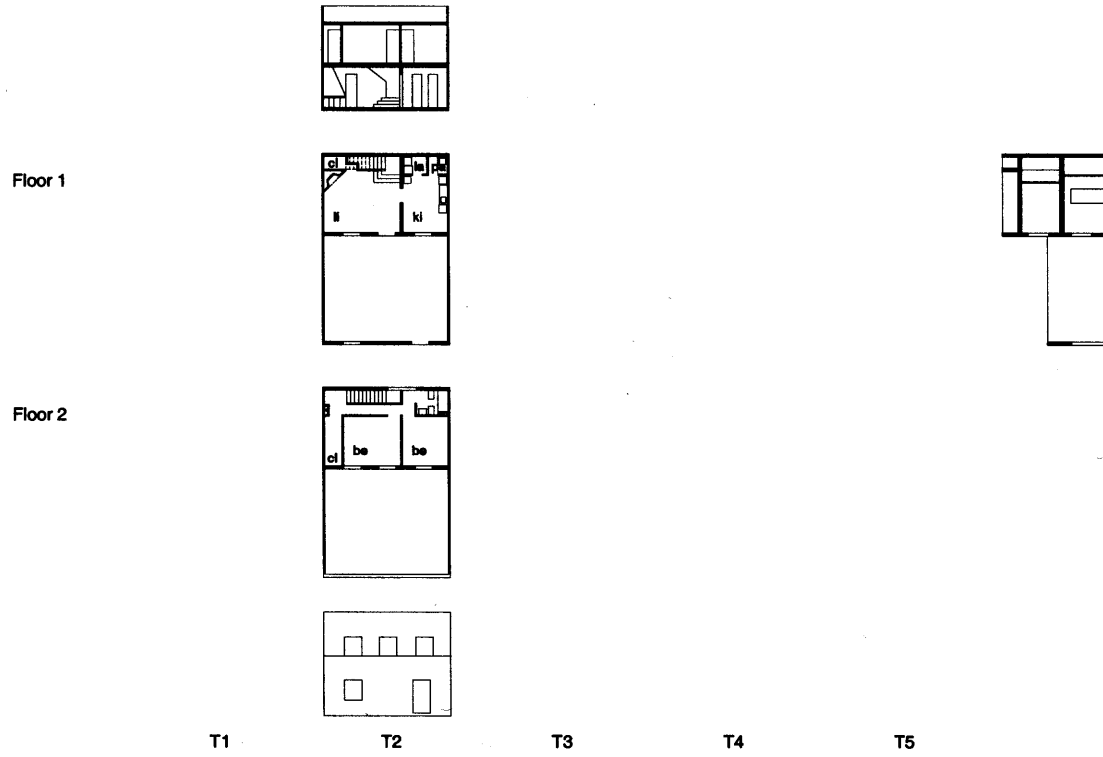


Figure 4.42 - Plans, sections, and elevation of type E (Siza's own house.)

# Housetype Ab - 1978

Malagueira - Alvaro Siza Vieira

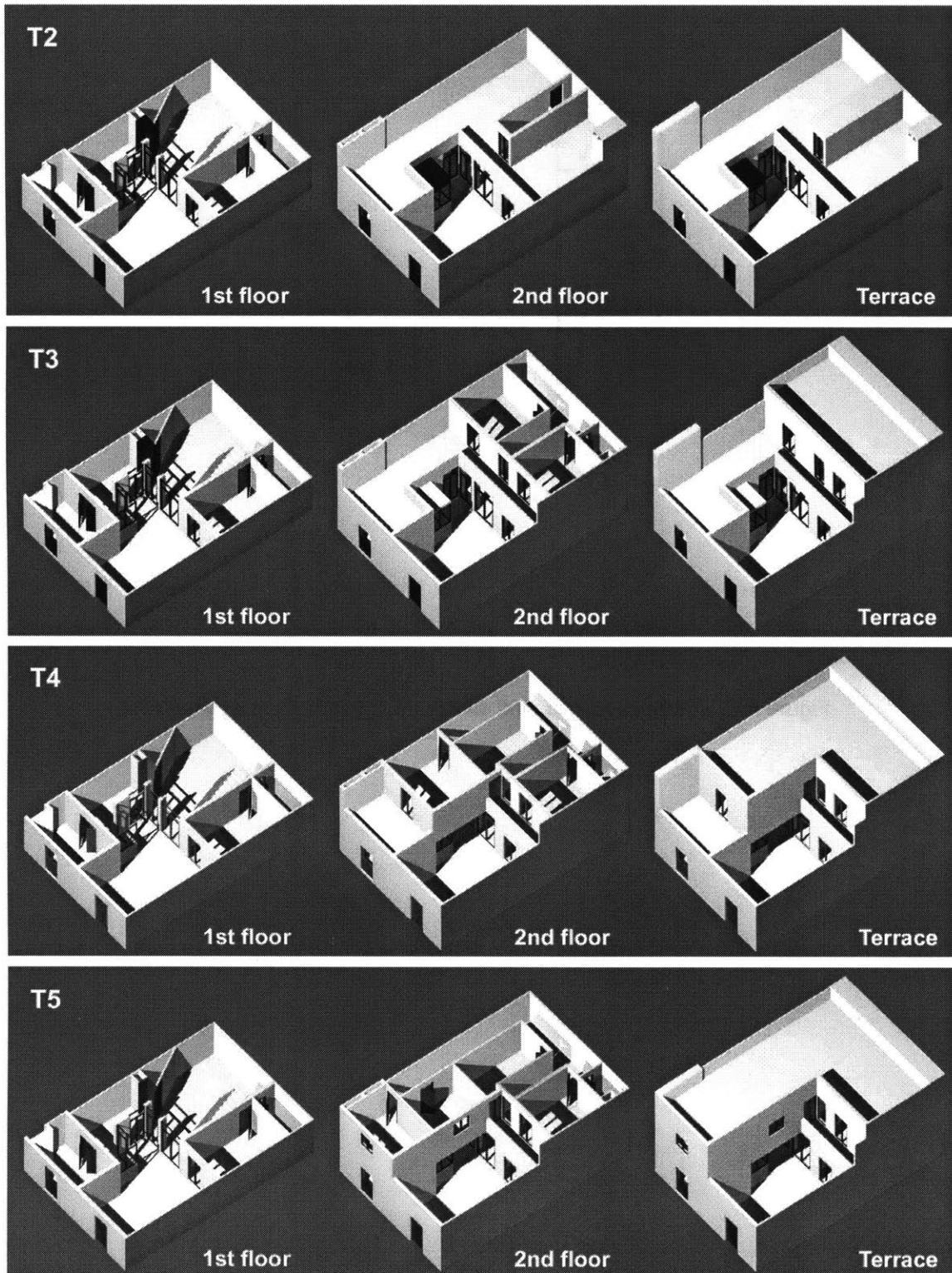


Figure 4.52 - Digital model of Subtype Ab (Siza.)



# Housetype Ac - 1980

Malagueira - Alvaro Siza Vieira

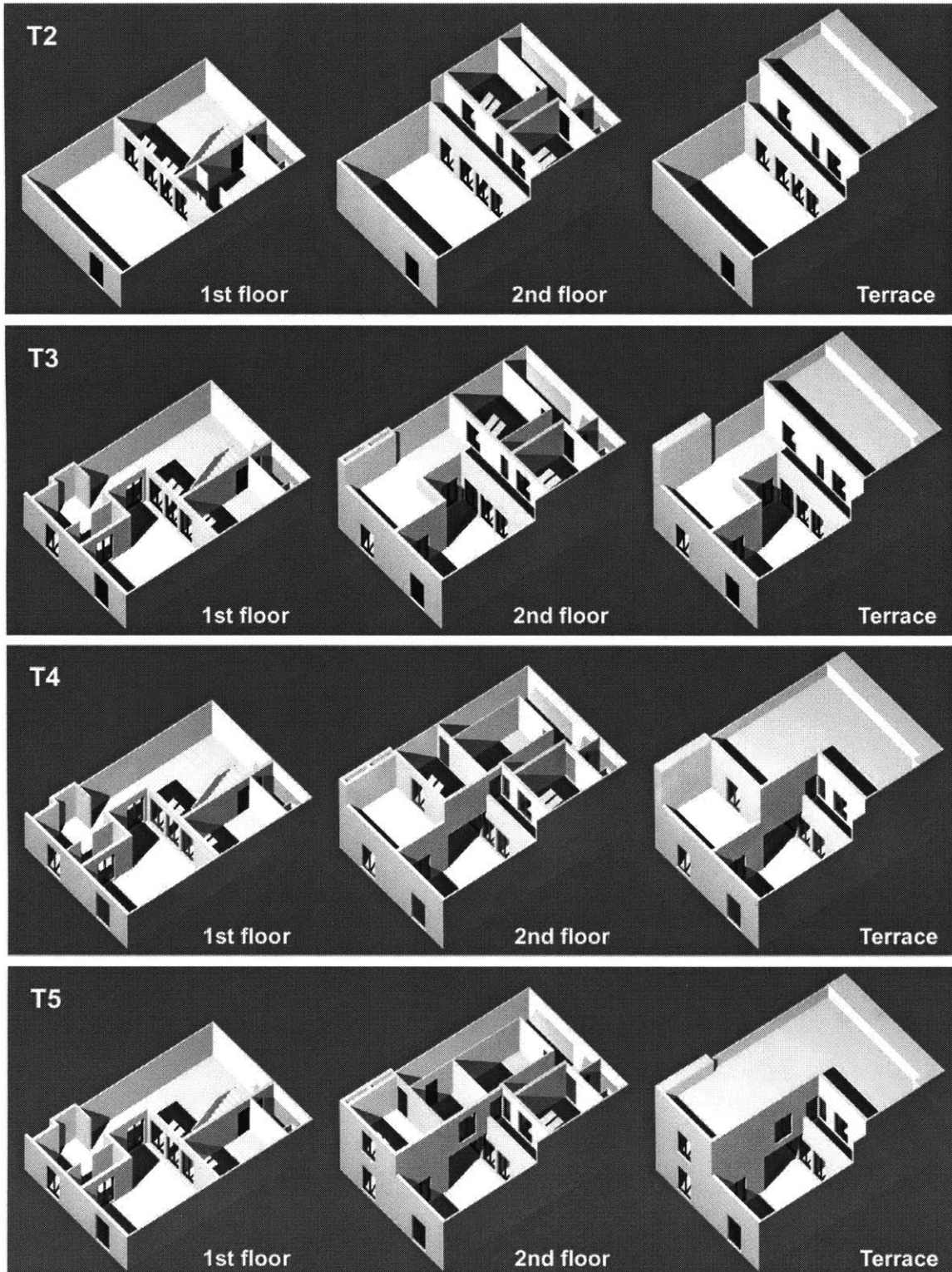


Figure 4.53 - Digital model of Subtype Ac (Siza.)

# Housetype Ad - 1996

Malagueira - Alvaro Siza Vieira

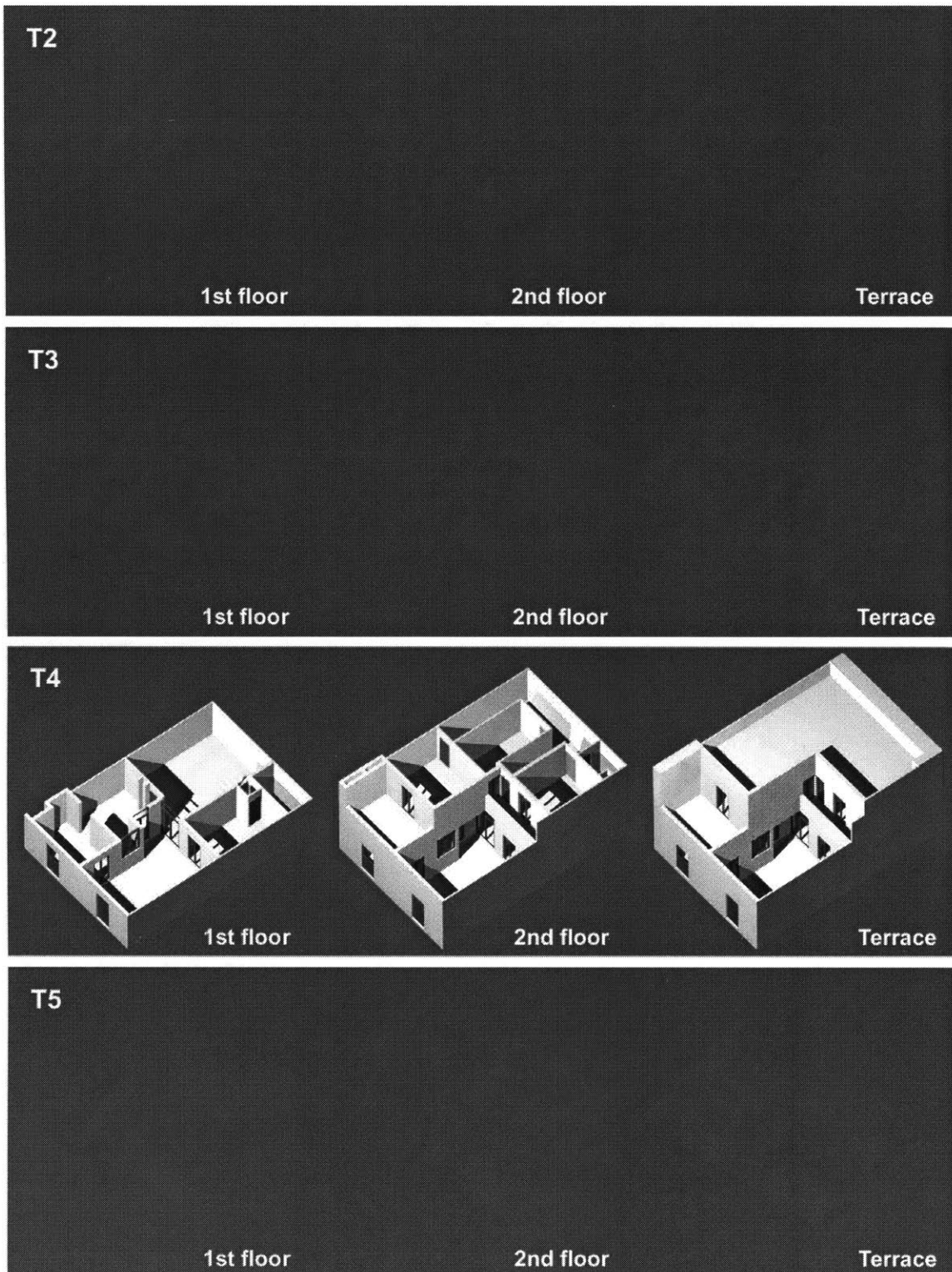


Figure 4.54 - Digital model of Subtype Ad (Lopes.)

# Housetype Ae - 1980

Malagueira - Alvaro Siza Vieira

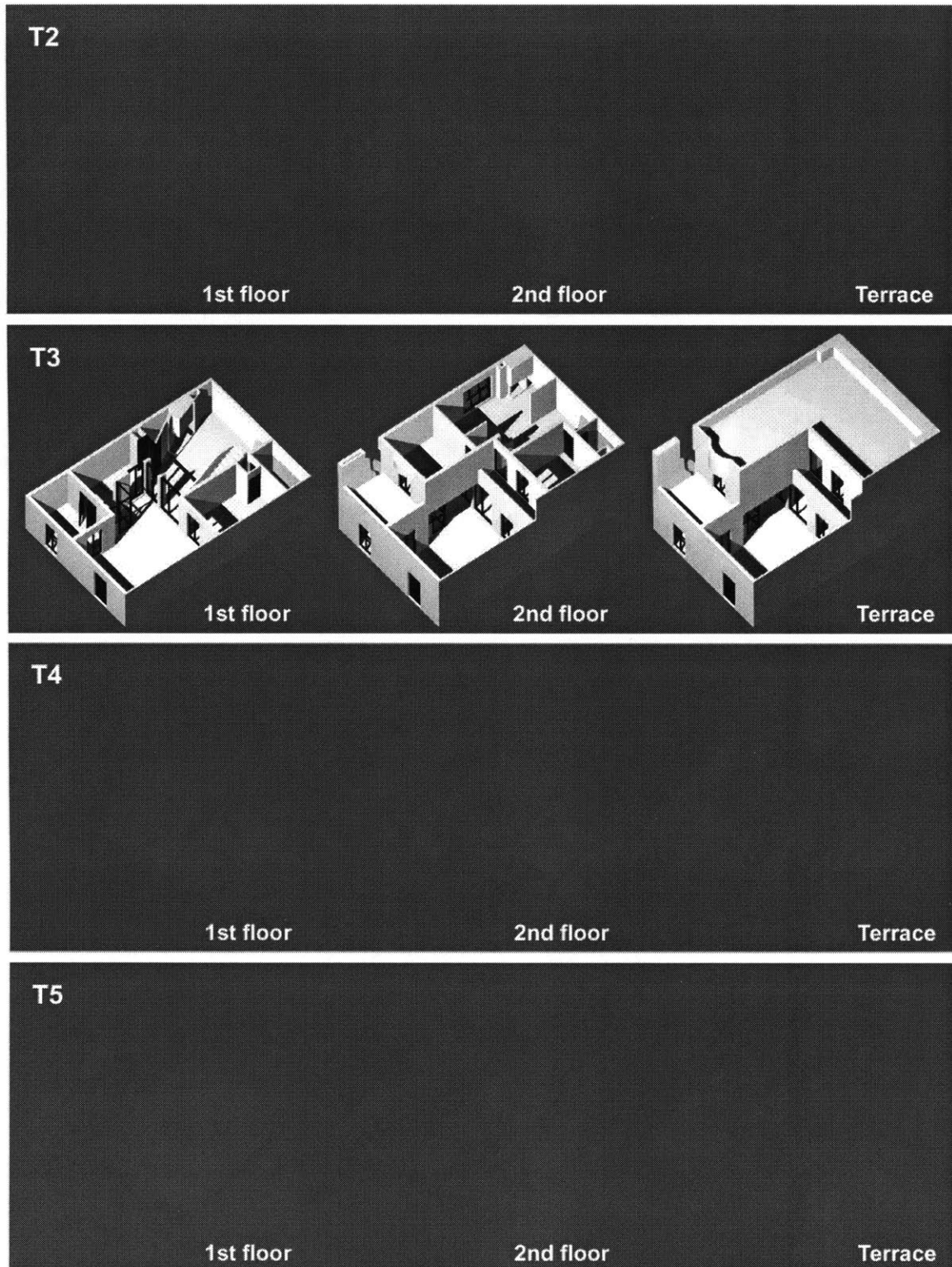


Figure 4.55 - Digital model of Subtype Ae (Lopes.)



# Housetype Bb - 1978

Malagueira - Alvaro Siza Vieira

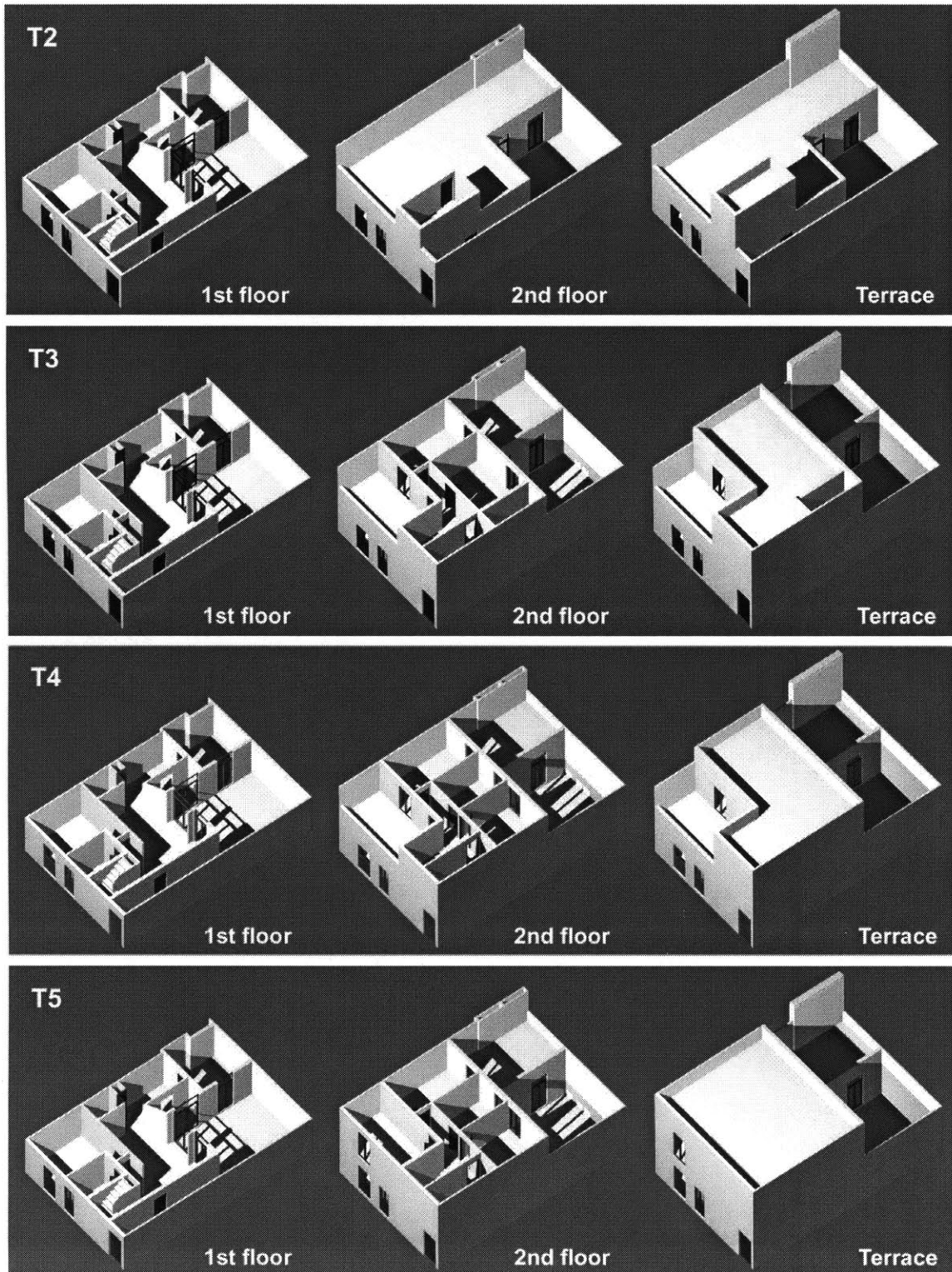


Figure 4.56 - Digital model of Subtype Bb (Siza.)

# Housetype Ca - 1985

Malagueira - Alvaro Siza Vieira

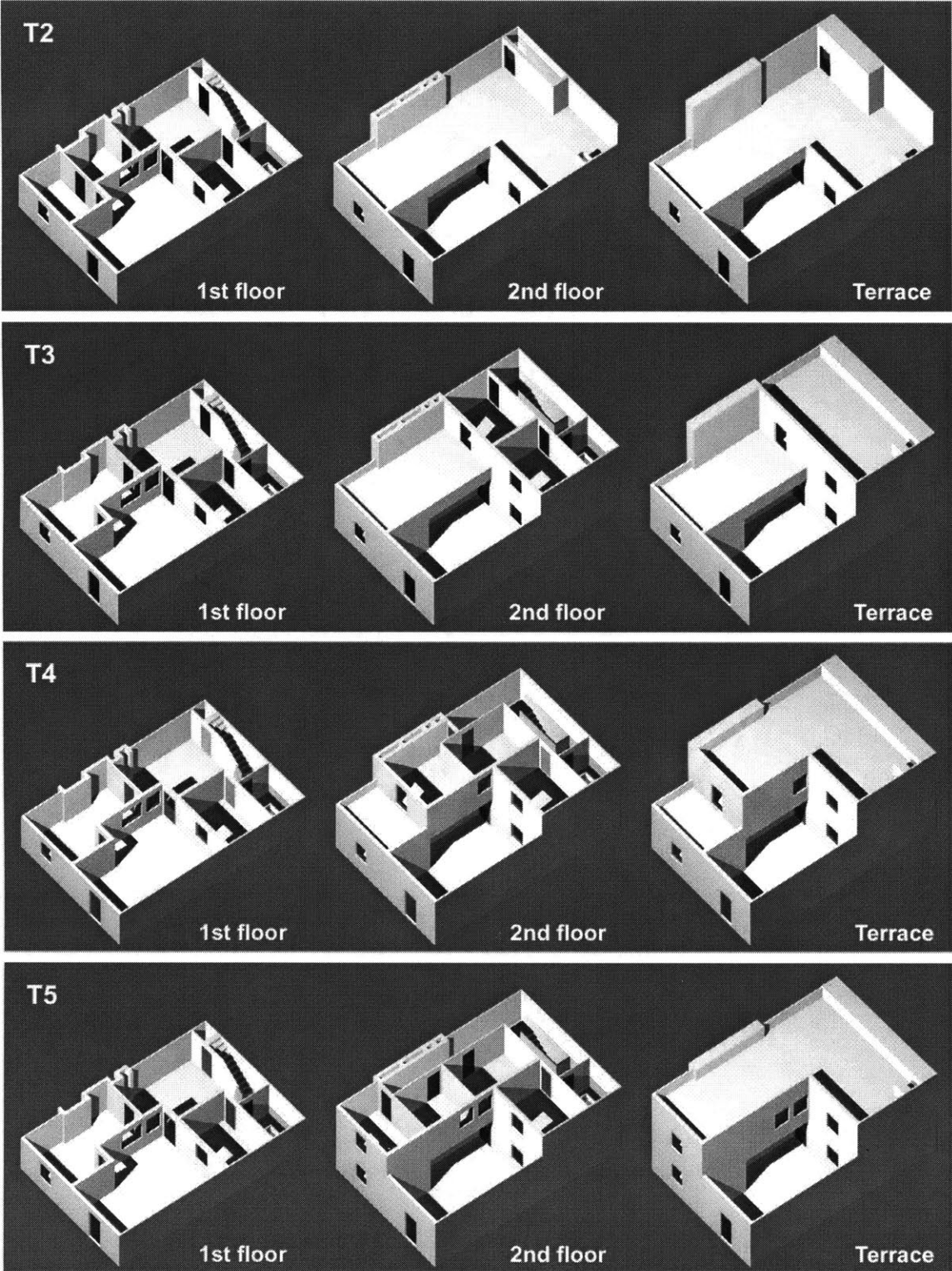


Figure 4.57 - Digital model of Subtype Ca (Siza.)

# Housetype Cb - 198(?) Malagueira - Alvaro Siza Vieira

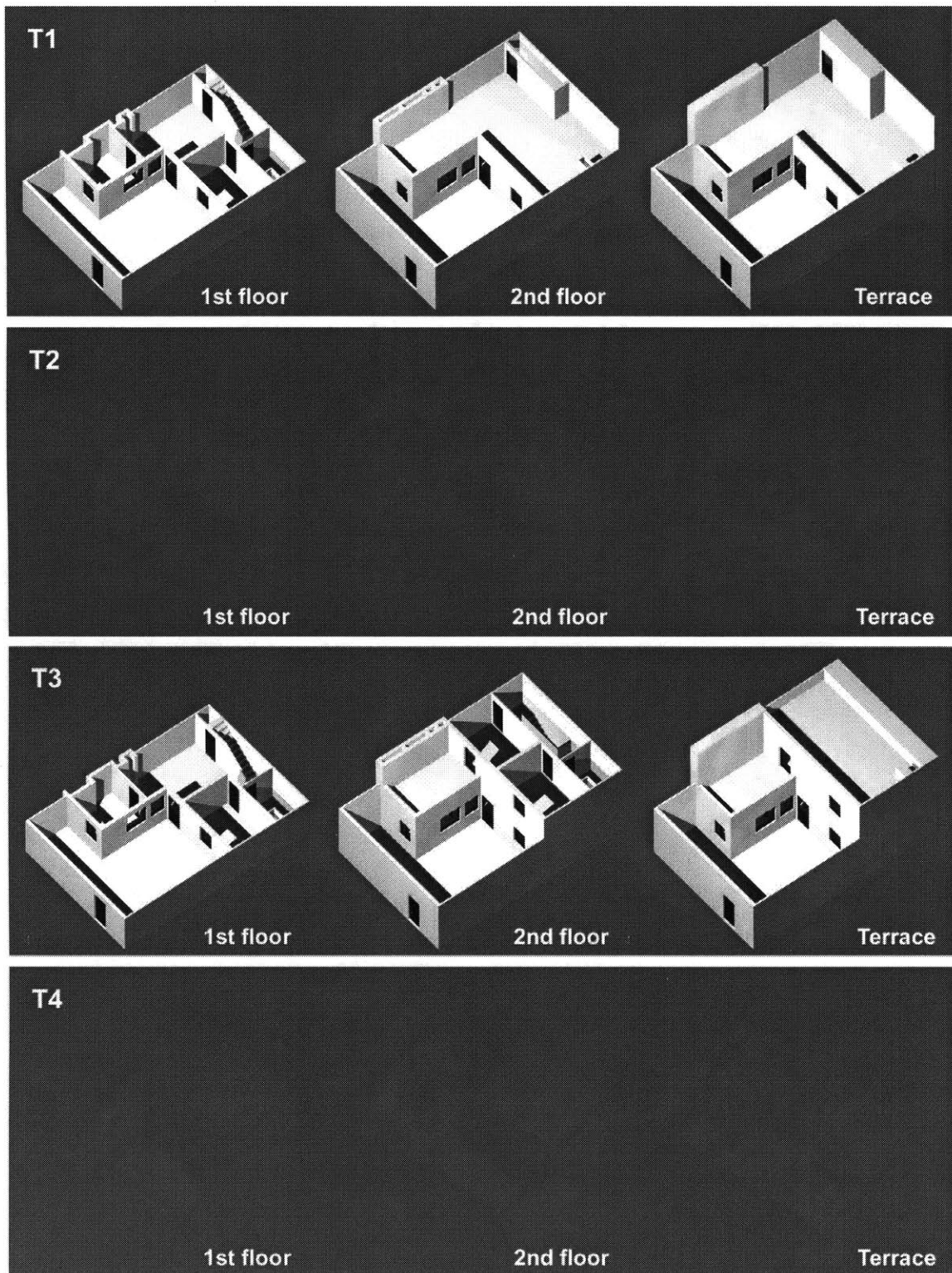


Figure 4.58 - Digital model of Subtype Cb (Siza.)



# Housetype Da - 1988

Malagueira - Alvaro Siza Vieira

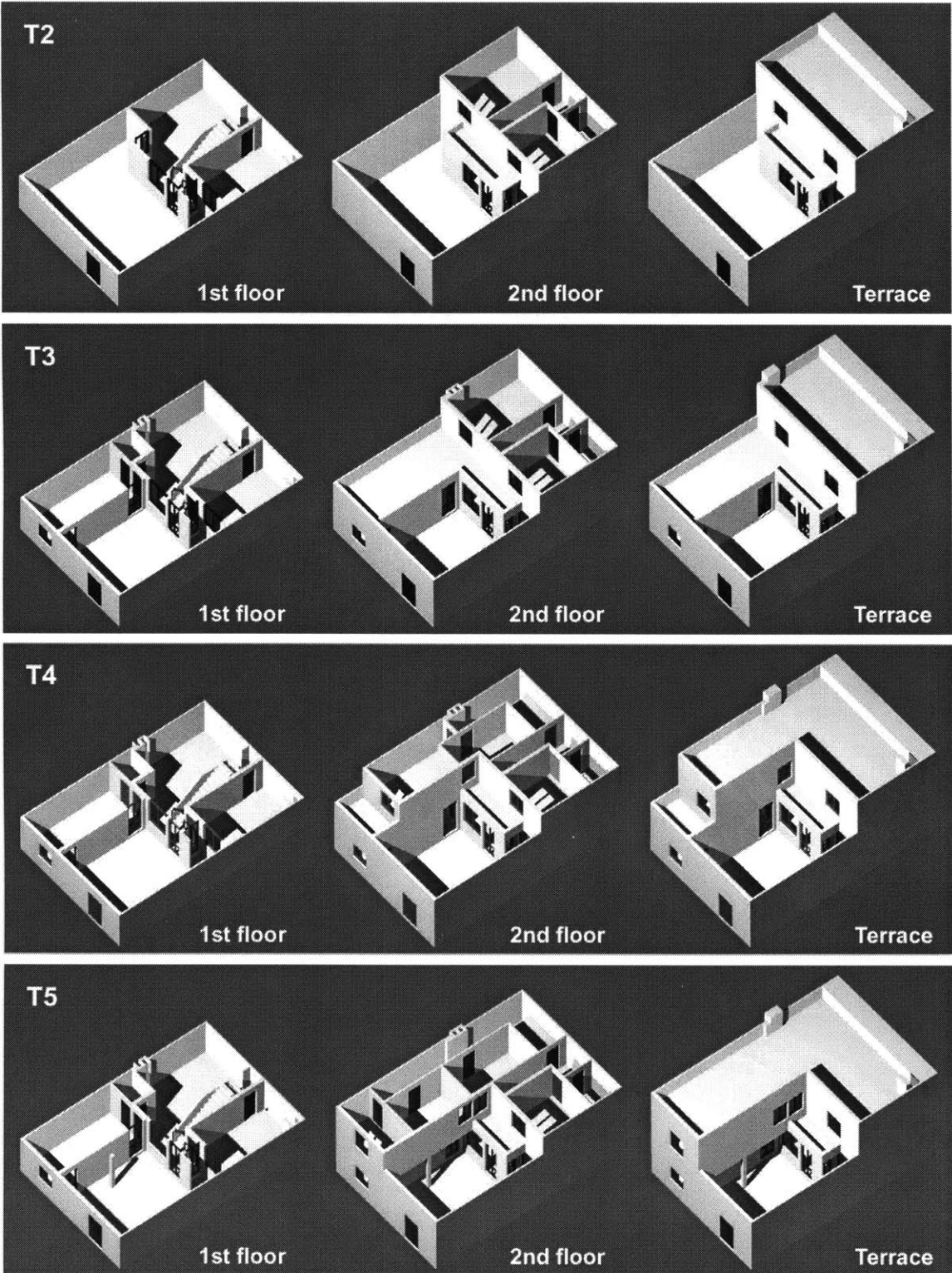


Figure 4.59 - Digital model of Subtype Da (Siza.)

# Housetype Db - 1988

Malagueira - Alvaro Siza Vieira

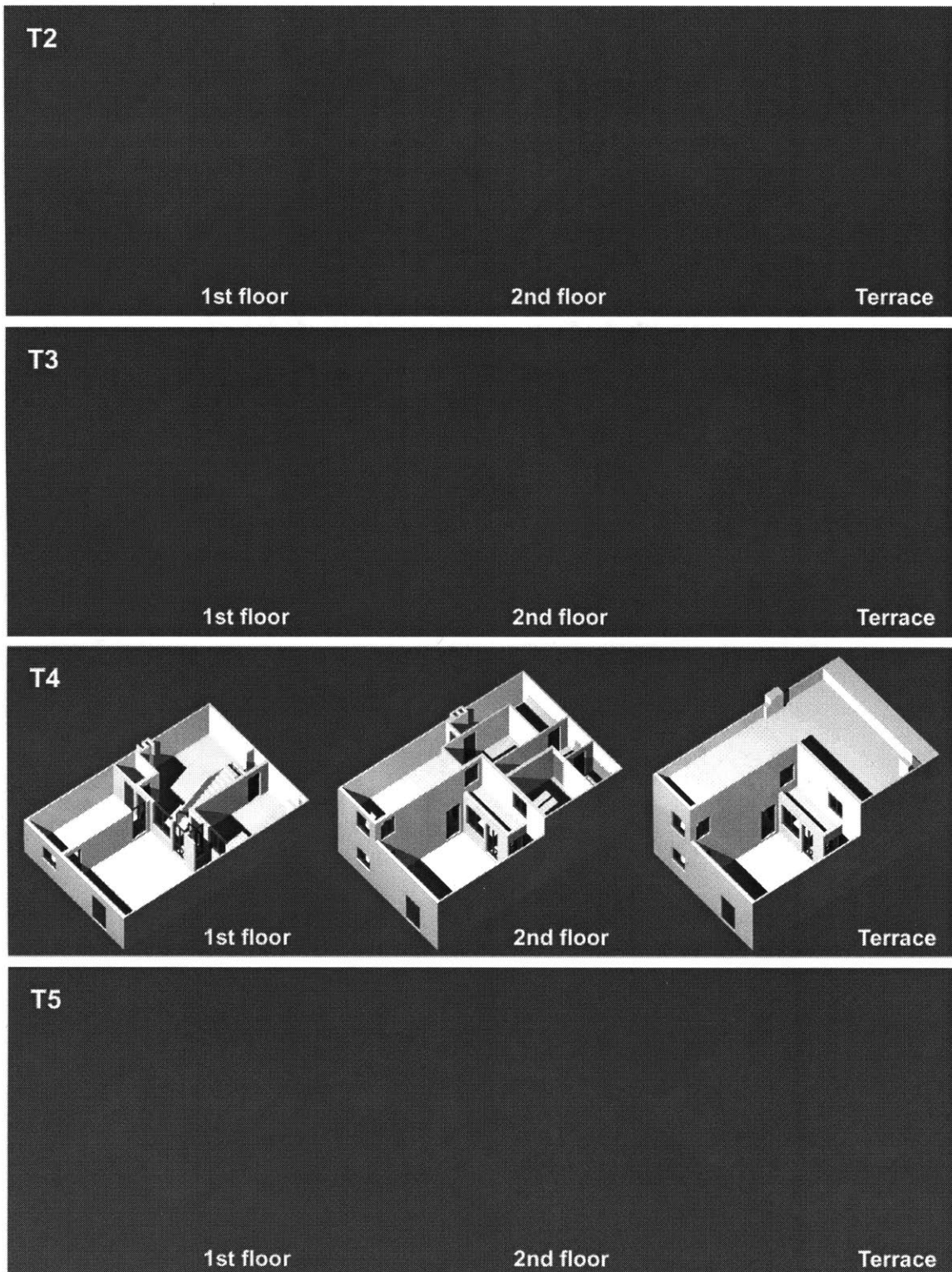


Figure 4.60 - Digital model of Subtype Db (Siza.)



# Housetype E - 1984

Malagueira - Alvaro Siza Vieira

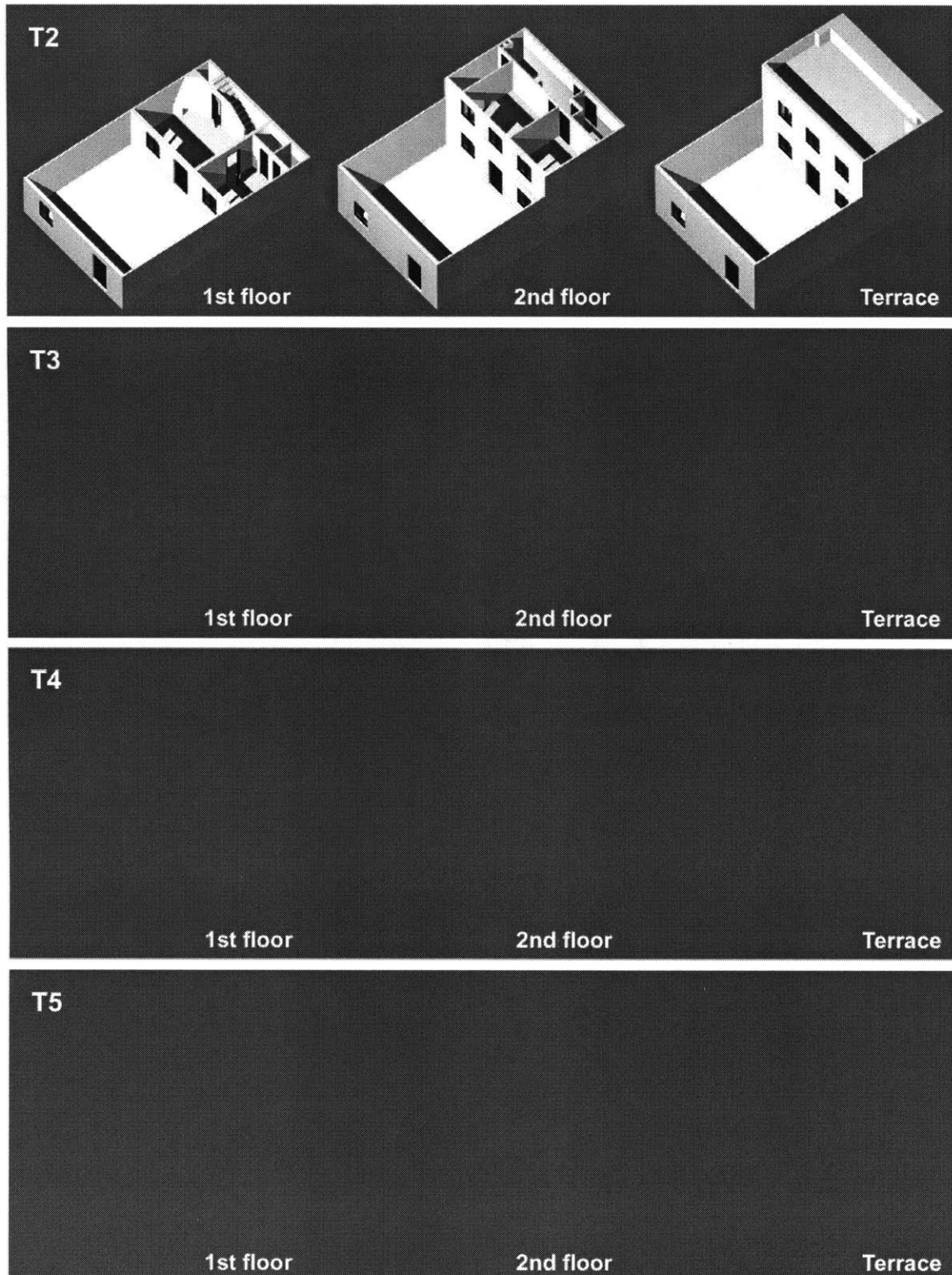
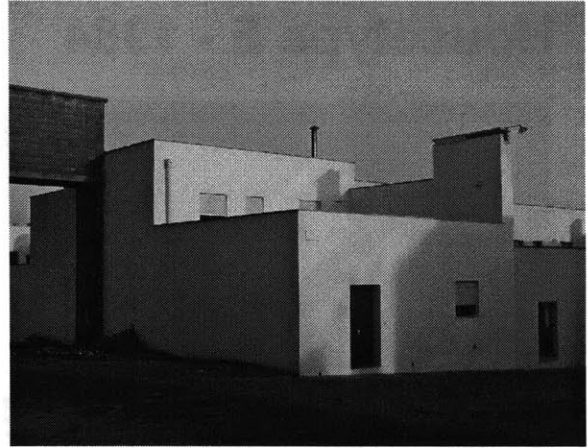


Figure 4.52 - Digital model of Type E (Siza's own house.)



t2



t3



t4



t5

Figure 4.43 - Subtype Ab, 1978.



t2



t3



t4



t5

Figure 4.44 - Subtype Ac, 1980.



South

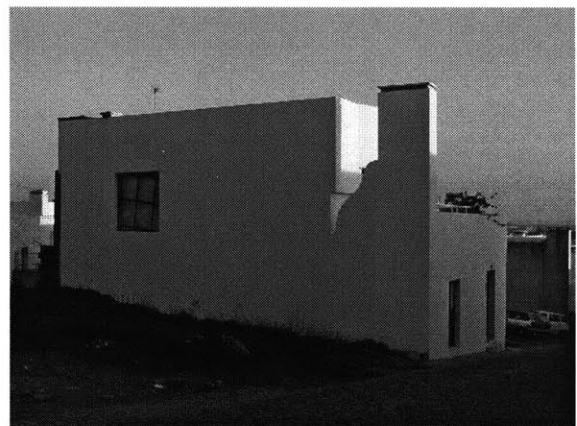


North

Figure 4.45 – Subtype Ad, 1996 (Nuno Lopes.)



Southeast



Northwest

Figure 4.46 - Subtype Ae, 1980 (customized house by Nuno Lopes.)



None was built.



t2

t3



t4

t5

Figure 4.47 - Subtype Bb, 1978.



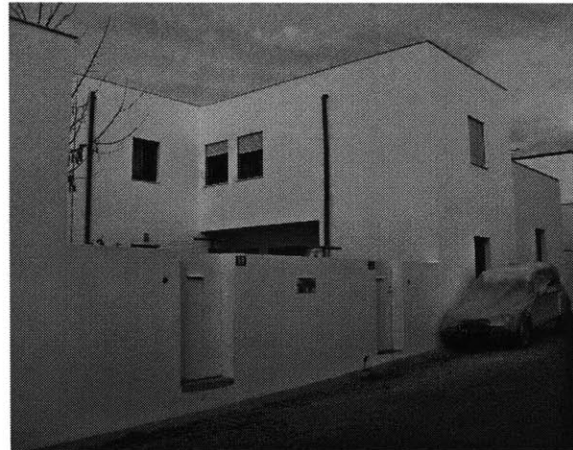
t2



t3



t4



t5

Figure 4.48 - Subtype Ca, 1985.



t2



t3



t4



t5

Figure 4.49 - Subtype Da, 1988.



t3



t4

Figure 4.50 - Subtype Db, 1995.



Front façade

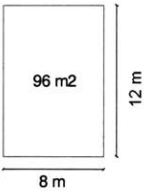
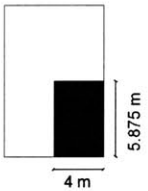
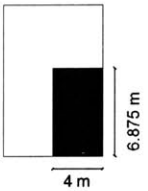
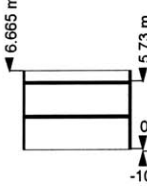
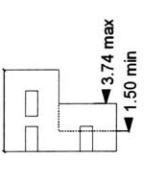
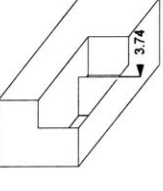

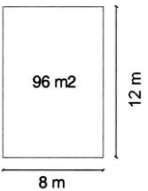
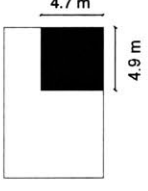
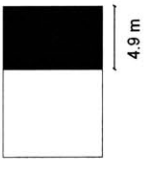
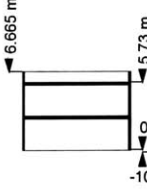
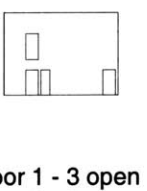
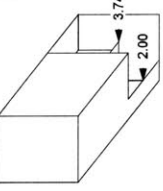
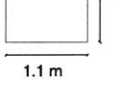


Back façade

Figure 4.51 - Type E, 1984 (Siza's own customized house.)



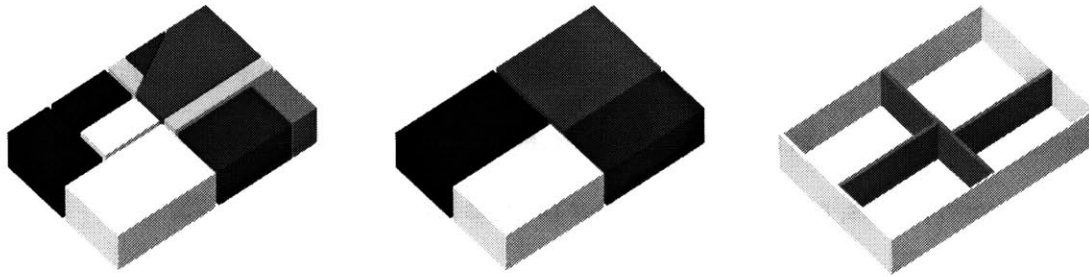
**Table 4.3 - Malagueira Building Regulations**

	Lot area and dimensions	Alignments and mandatory free-space 1 <sup>st</sup> floor	Alignments and mandatory free-space 2 <sup>nd</sup> floor	Maximum number of floors	Street elevation: maximum surface area, number of openings, and wall height	Maximum volume	Openings: maximum dimension (only second floor's street elevation)	Recommendations
				<p>0 = floor 1 level</p> 	 <p>floor 1 - 2 open floor 2 - 1 open</p>		<p>0 = floor level</p> 	<p>Check 1 Town H project-</p> <p>Enclosure and chimney should be studied collaboratively with the Hall</p>
				<p>0 = floor 1 level</p> 	 <p>floor 1 - 3 open floor 2 - 1 open</p>			<p>The yard should have garden cover with ivy lattice</p>
<p>ons</p>	<ul style="list-style-type: none"> <li>a) One or two-storey houses</li> <li>b) Annexes, store-rooms, and garages are not allowed</li> <li>c) Respect for the National and the Municipal building regulations</li> <li>d) First and second floor levels should be requested in the Town Hall</li> <li>e) Use a Town Hall's expandable project-type or a project that respects these regulations (subject to approval by town hall)</li> <li>f) Individual or collective garages available, according to Town Hall plan and regulations</li> <li>g) Use service gallery and its walls</li> <li>h) External whitewashed walls, terraces, wooden or colored aluminum mullions are mandatory</li> <li>i) Overhangs or cantilevered volumes are not allowed</li> <li>j) Number and dimensions of openings are constrained. Mortar frames with a maximum overhang of 1 cm and 20 cm wide, painted in traditional colors (gray, yellow, green, blue, and rose) are allowed</li> </ul>							

### **4.3.2 Structural and building systems**

When the Malagueira project was initiated, undertaking a development with such a dimension in Évora was problematic because there was no local building company with the required financial and technical means. (Molteni, 1997) In addition, it was important to choose local techniques and materials and to create local jobs to maintain the desired independence from the central government. Therefore, in 1977 when Siza started his work with the first group of future inhabitants, self-construction was sought. The original structural system used in the Malagueira houses consisted of load-bearing walls and pre-stressed concrete beams, with both the walls and the pavement infill material made of concrete blocks. A slab or a wall of reinforced concrete poured on site was occasionally used for adjusting this system. The building system was devised to allow construction to proceed incrementally, in small units. As time went by, the economic situation improved and the power struggle with the central government faded away. Then market pressures led the cooperatives and other housing promoters to hire contractors, and by the end of 1980s, a structural frame of reinforced concrete poured on site with infill brick walls, started to be used as well. This shift in structural system, however, did not greatly affect the design, as pillars and beams were designed to be completely incorporated into the walls and pavements.

The choice of structural system impacted the design in three ways. First, as the maximum that beams could span was 5 meters, the lot was divided into four by load-bearing walls. Such a division had a correspondence with the functional division of the home into four distinct zones: yard, living, sleeping, and service. (Figure 4.62) The maximum span, in turn, determined the thickness of the floor slab.



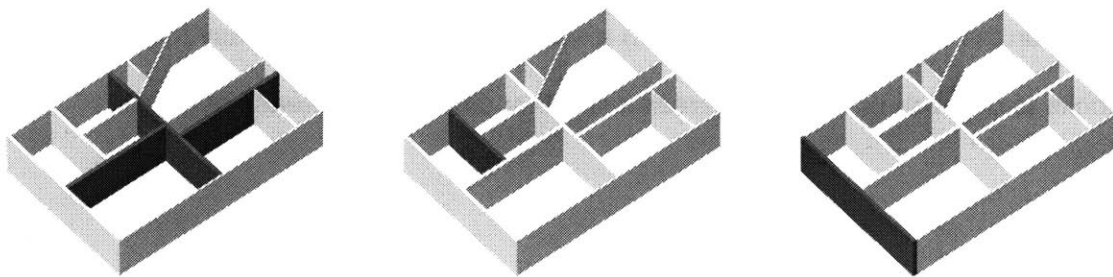
Functional organization

Division into functional zones

Main load bearing walls

Figure 4.62 – Correspondence between functional organization and structural system (Ab.)

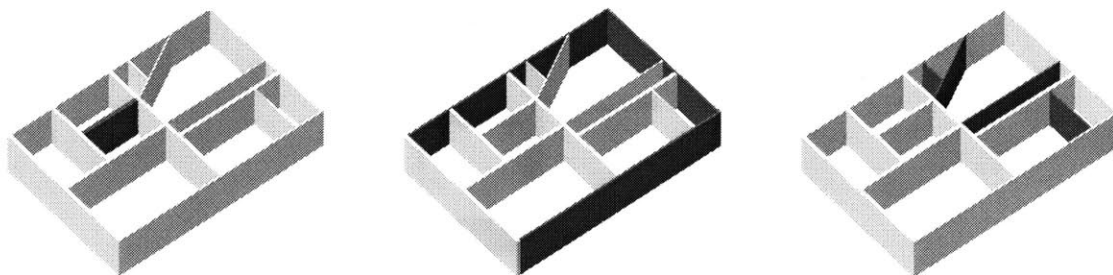
Second, blocks of decreasing thickness, 0.20, 0.10, and 0.075 m were used selectively to create a hierarchy of walls. (Figure 4.63) The thicker blocks were used for load-bearing walls or external walls. The middle blocks were used for the walls that enclosed the lot. Finally, the thinner blocks were used for partition walls.



Main Load-bearing walls (0.20m)

Secondary load-bearing wall (0.20m)

Exterior wall (0.20m)



Secondary exterior walls (0.20m)

Enclosing walls (0.10m)

Partition walls (0.075m)

Figure 4.63 – Hierarchy of walls (subtype Ab.)

Third, the height of the blocks (0.20 m) determined the floor height, according to the following equation:  $h_f = h_l + n \times h_b + h_j$ , in which  $h_l$  is the height of the grounding beam above the floor level,  $h_b$  is the height of concrete blocks,  $h_j$  is the height of the mortar joints between blocks, and  $n$  is the number of rows that form the wall. As the Portuguese regulations determine 2.40 m to be the minimum floor height, this value is the lower-bound value for the parameter. As shown in Table 4.4, the specific sectional dimensions varied from house to house due to variations on the dimensions of the specific blocks used. The choice of blocks, the number rows in the wall, and the height of the mortar joint were constrained by adjustments to topographic conditions and the need to maintain a certain rhythm on the street facade. To simplify the grammar, such a dependency of parameters was disregarded and the standard dimensions shown in Figure 4.64 were adopted. The floor height and slab thickness imposed constraints on the stair design, as it will be explain in Chapter 7. The length of the blocks did not constrain the dimensioning of rooms in a similar way because it is easy to break the blocks to diminish its length, whereas it is rather difficult to decrease its height.

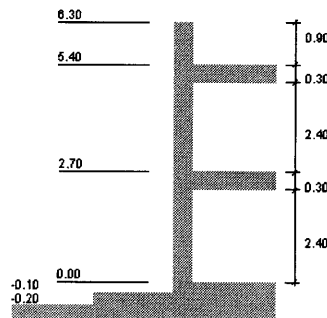


Figure 4.64 – Standard sectional dimensions

**Table 4.4 – Variation of height parameter values from type to type**

Subtype	Aa Aug	Aa Nov	Ab	Ac	Ba	Bb	Ca	Cb	Da	Db	E	min	max
↓ house	<b>6.05</b>	<b>5.40</b>	<b>6.28</b>	<b>6.64</b>	<b>5.40</b>	<b>6.28</b>	<b>6.24</b>	<b>6.15</b>	<b>6.16</b>	<b>6.30</b>	<b>6.60</b>	<b>5.40</b>	<b>6.60</b>
h <sub>te</sub>	0.65	0.00	0.91	0.84	0.00	0.91	0.91	0.90	0.88	0.90	1.10	0.84	1.10
↓ terrace	<b>2.70</b>	<b>2.70</b>	<b>2.65</b>	<b>2.90</b>	<b>2.70</b>	<b>2.65</b>	<b>2.64</b>	<b>2.60</b>	<b>2.64</b>	<b>2.70</b>	<b>2.64</b>	<b>2.64</b>	<b>2.90</b>
h <sub>s3</sub>	0.20	0.20	0.18	0.25	0.30	0.18	0.20	0.20	0.20	0.30	0.20	0.18	0.30
h <sub>l2</sub>	2.50	2.50	2.47	2.65	2.40	2.47	2.44	2.40	2.44	2.40	2.44	2.40	2.65
↓ 2 <sup>nd</sup> fl.	<b>2.70</b>	<b>2.70</b>	<b>2.72</b>	<b>2.80</b>	<b>2.70</b>	<b>2.72</b>	<b>2.69</b>	<b>2.65</b>	<b>2.64</b>	<b>2.64</b>	<b>2.86</b>	<b>2.64</b>	<b>2.86</b>
h <sub>s2</sub>	0.30	0.30	0.25	0.25	0.30	0.25	0.25	0.25	0.20	0.30	0.20	0.20	0.30
h <sub>l1</sub>	2.40	2.40	2.47	2.65	2.40	2.47	2.44	2.40	2.44	2.40	2.66	2.44	2.66
↓ 1 <sup>st</sup> fl.	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
↓ patio (-)	0.20	0.20	0.05	0.05	0.05	0.05	0.07	0.07	0.10	0.10	0.10	0.05	0.10
↓ street (-)	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.22	0.20	0.20	0.22

#### 4.4 Summary

Siza followed a systemic approach to both the design of the plan and the design of the houses. This approach can be summarized as the use of existing references as a basis to create an enduring structure that supports growth based on a set of rules. Structure is understood both as a physical artifact and a compositional framework. In the plan such references were illegal developments, pathways, and an aqueduct. The backbone of the structure was a service duct that branched off to create a grid of lots, adjusted to the topography. This lot formed a self-supported basic unit that allowed houses to be built independently. In the design of the houses, the references were traditional and illegal housetypes that were interpreted and transformed to create housetypes for contemporary lifestyles. These types provided a structure that allowed different houses to be built and expanded to adjust themselves to the needs of their inhabitants. The housetypes are encoded by a set of rules for formal and functional composition, which are explained over the next chapters using a shape grammar and a descriptive grammar, respectively.

## References

Ensemble d'habitations Quinta da Malagueira, Évora, 1977, L'architecture d'Aujourd'hui n. 211, 1980, pp. 60-65.

Molteni, Enrico. "Álvaro Siza, barrio de la Malagueira, Évora" Edicions UPC, Universitat Politècnica de Catalunya, 1997.

Siza, Álvaro, Notas sobre o trabalho em Évora, Arquitectura, n.132, Lisboa, 1979, pp-36.

Testa, Peter "The Architecture of Álvaro Siza," S.M.Arch.S. Thesis, Massachusetts Institute of Technology, Cambridge, 1984.

Fleck, Brigitte, "Álvaro Siza," Birkhäuser, Basel, 1992.

## **5. The Malagueira Shape Grammar**

### **5.1 Introduction**

This chapter presents the initial shape grammar developed for Siza's Malagueira houses. The grammar is based on the corpus of thirty-five houses designed between 1977 and 1996 described in the previous chapter. The grammar shows that the generation of houses is defined by the recursive dissection of rectangles, locating four different functional zones (patio, living, services, and sleeping) and the key placement of the staircase. The generation of one prototypical design, the 1978 housetype A, is described based on the grammar.

### **5.2 Algebras**

Shapes, labels, and weights can be combined to form shape grammars that encode specific languages of designs. Moreover, these grammars can combine several of these components to form different, but useful, ways of seeing and describing designs. For instance, one can define different grammars to generate plans, elevations, and axonometrics. The type and number of descriptions or viewpoints depend on the kinds of designs captured by the grammar, the purpose one has in developing it, and the level of detail desired. In the Malagueira grammar, the descriptions considered were the ones shown in Figure 5.1, namely, axonometrics of the envelope and the spaces, first, second, and terrace floors plans, and front elevation. The level of detail considered was

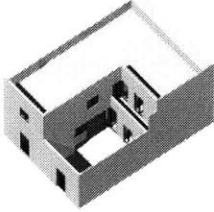
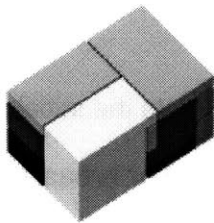
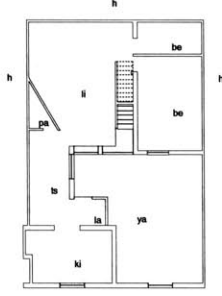
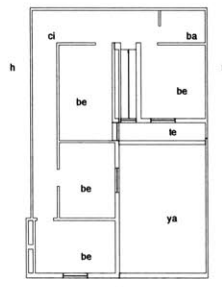
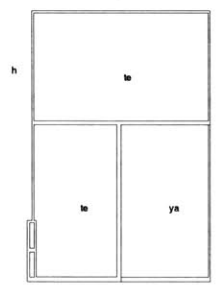
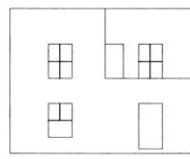
	<b>View</b>	<b>Algebra</b>	
W	Envelope (walls)	$U_{33}$	
R	Spaces (rooms)	$W_{33}$	
F1	1st floor plan	$U_{12}V_{02}$	
F2	2nd floor plan		
F3	Terrace plan		
E	Front elevation		

Figure 5.1 – Viewpoints and algebras considered in the Malagueira shape grammar.



the one that corresponds to 1/100 scale drawings. These viewpoints were chosen because they were used in Siza's office at the preliminary design stage.

### 5.3 Parallel grammars

The Malagueira grammar is a parametric shape grammar defined in the cartesian product of the algebras represented by the following matrix:

$$\begin{bmatrix} \langle U_{33} V_{03} \rangle \\ \langle W_{33} V_{03} \rangle \\ \langle U_{12} V_{02} \rangle \\ \langle U_{12} V_{02} \rangle \\ \langle U_{12} V_{02} \rangle \\ \langle U_{12} V_{02} \rangle \end{bmatrix}.$$

Both the envelope and the spaces axonometrics (rows 1-2, respectively) were defined in the cartesian product of the algebras  $W_{33}$  and  $V_{03}$ . Labeled dots indicating the origin of the referential system in which designs are described and the insertion points of shapes in  $W_{33}$  are the elements in the algebra  $V_{03}$ . (To simplify graphic representation, these labeled dots were omitted in the derivation of an existing design shown further below.

(Figure 5.1) As the grammar was defined, the floor plans, and the elevations control the generation of designs. The axonometrics are only used for visualization purposes as a way of facilitating visual understanding. The three floor plans and the elevation (rows 3-6) were defined in the cartesian product of the algebras  $U_{12}$  and  $V_{02}$ . Together, they provide two-dimensional representations of the three-dimensional shapes of Malagueira house designs.

In the product of algebras considered, a rule has the format

$$\left[ \begin{array}{c} \langle S_A L_A \rangle \\ \langle S_A L_A \rangle \\ \langle S_A L_A \rangle \\ \langle S_A L_A \rangle \\ \langle S_A L_A \rangle \\ \langle S_A L_A \rangle \end{array} \right] \rightarrow \left[ \begin{array}{c} \langle S_B L_A \rangle \\ \langle S_B L_A \rangle \\ \langle S_B L_B \rangle \\ \langle S_B L_B \rangle \\ \langle S_B L_B \rangle \\ \langle S_B L_B \rangle \end{array} \right],$$

meaning that if certain shapes A are found in each description, these shapes are replaced by shapes B. Nevertheless, only severely constrained rules require the existence of certain shapes in all the descriptions for the rule to be applied. The application of most rules only requires the presence of certain shapes in some of the descriptions and it only affects some of them as well. Consider, for instance, a rule to pierce an opening in the front elevation of the first floor has the following matrix format:

$$\left[ \begin{array}{c} \langle S_A L_A \rangle \\ \langle S_\emptyset L_\emptyset \rangle \\ \langle S_A L_A \rangle \\ \langle S_\emptyset L_\emptyset \rangle \\ \langle S_\emptyset L_\emptyset \rangle \\ \langle S_A L_A \rangle \end{array} \right] \rightarrow \left[ \begin{array}{c} \langle S_B L_B \rangle \\ \langle S_\emptyset L_\emptyset \rangle \\ \langle S_B L_B \rangle \\ \langle S_\emptyset L_\emptyset \rangle \\ \langle S_\emptyset L_\emptyset \rangle \\ \langle S_B L_B \rangle \end{array} \right],$$

where  $S_\emptyset L_\emptyset$  are empty empty labeled shapes.

In the Malagueira grammar, the plan of the first floor drives the generation of designs in the grammar. The layout of upper floors is, to a considerable extent, constrained by decisions made on the first floor, due to structural constraints. The elevations also are determined by the layout of the floors. Decisions about the elevations never imply changes in the layouts. This dependency is encoded into the grammar through the use of sequential, parallel grammars, one for each floor and the elevation, as diagrammed in Figure 5.2.

Gram Stages

-mar	1: define first floor	2: define second floor	3: terrace
F <sub>1</sub>	S Z C R D O T		
F <sub>2</sub>	↓ ↓ ↓ ↓ ↓ ↓ ↓	S Z C R D O T	
F <sub>3</sub>		↓ ↓ ↓ ↓ ↓ ↓ ↓	S Z D T
E			

Legend: F<sub>1</sub> – first floor; F<sub>2</sub> – 2nd floor; F<sub>3</sub> – terrace; E – elevation; S – Start; Z – Locate functional zones; C – Define circulation scheme; R – Divide zones into rooms; D – Introduce details; O – Introduce openings; T – Terminate.

Figure 5.2 – Use of sequential, parallel grammars in the derivation of a Malagueira house. Dark shaded areas identify the currently active grammars (viewpoint,) light shaded areas identify passive grammars, and non-shades areas identify non-active grammars. Letter symbols identify steps of the derivation, and arrow symbols identify the placement of labels in the passive grammar viewpoint to be recalled later in the derivation.

The derivation of a design in the grammar goes through three successive stages: defining the first floor (F<sub>1</sub>), defining the second floor (F<sub>2</sub>), and defining the terrace (F<sub>3</sub>). While the generation of the first floor proceeds, labels are placed on the second floor and on the elevation (E). When the generation of the first floor finishes, a state label changes thereby activating the generation of the second floor, which proceeds using the previously placed labels as beacons. The articulation between the generation of the second floor and the terrace works in a similar fashion.

Each of these stages, in turn, includes several steps as listed in Figure 5.3. For instance, the stages of the first floor are locating functional zones, locating staircase, dividing functional zones (into rooms), introducing details (fireplaces, chimneys, etc.), and introducing openings. The separation into steps is merely analytical, as there are no state labels like those used to take the derivation from on stage to another.

<p><b>Stage 0:</b> F0 – introduce initial shape</p> <p><b>Stage 1:</b> F1 – define 1<sup>st</sup> floor</p> <p>Step 1.1: E – start</p> <ul style="list-style-type: none"> <li>introduce slab</li> <li>enclose floor</li> <li>adjust wall thickness</li> </ul> <p>Step 1.2: L – locate functional zones</p> <ul style="list-style-type: none"> <li>locate patio</li> <li>locate external corridor</li> <li>locate living-room</li> <li>locate sleeping area</li> <li>locate service area</li> </ul> <p>Sep 1.3: C – define circulation scheme</p> <ul style="list-style-type: none"> <li>locate main entrance</li> <li>locate staircase</li> </ul> <p>Step 1.4: D – divide zones into rooms</p> <ul style="list-style-type: none"> <li>divide yard</li> <li>divide service zone</li> <li>divide living zone</li> <li>divide sleeping zone</li> <li>extend rooms</li> <li>assign rooms</li> <li>connect rooms</li> <li>permute rooms</li> </ul> <p>Step 1.5: A – introduce details</p> <ul style="list-style-type: none"> <li>introduce chimneys</li> <li>adjust wall thickness</li> <li>adjust the patio wall height</li> <li>pierce patio</li> <li>detail stairs</li> </ul> <p>Step 1.6: O – pierce openings</p> <ul style="list-style-type: none"> <li>pierce exterior openings</li> <li>pierce interior openings</li> <li>introduce openings</li> </ul> <p>Step 1.7: R – terminate</p> <ul style="list-style-type: none"> <li>erase labels and changing state</li> </ul>	<p><b>Stage 2:</b> F2 – define 2<sup>nd</sup> floor</p> <p>Step 2.1: E – start</p> <ul style="list-style-type: none"> <li>introduce slab</li> <li>enclose floor</li> <li>adjust wall thickness</li> </ul> <p>Step 2.2: L – replicate 1<sup>st</sup> floor's divisions:</p> <ul style="list-style-type: none"> <li>into inside and outside zones</li> <li>into functional zones</li> <li>into rooms</li> </ul> <p>Step 2.3: C – define circulation scheme</p> <ul style="list-style-type: none"> <li>extending staircase</li> <li>defining circulation</li> </ul> <p>Step 2.4: D – define rooms</p> <ul style="list-style-type: none"> <li>divide zones into rooms</li> <li>assign rooms</li> <li>connect rooms</li> <li>(locate closets)</li> </ul> <p>Step 2.5: A – introduce details</p> <ul style="list-style-type: none"> <li>extend chimney</li> <li>adjust patio wall height</li> <li>pierce patio</li> </ul> <p>Step 2.6: O – pierce openings</p> <ul style="list-style-type: none"> <li>pierce exterior openings</li> <li>pierce interior openings</li> <li>introduce openings</li> </ul> <p>Step 2.7: R – terminate</p> <ul style="list-style-type: none"> <li>erase labels and changing state</li> </ul> <p><b>Stage 3:</b> F3 – define terrace</p> <p>Step 3.1: E – start</p> <ul style="list-style-type: none"> <li>introduce slab</li> <li>enclose floor</li> <li>adjust wall thickness</li> </ul> <p>Step 3.2: L – replicate 1<sup>st</sup> floor's divisions:</p> <ul style="list-style-type: none"> <li>into inside and outside zones</li> <li>into functional zones</li> </ul> <p>Step 3.3: C – introduce details</p> <ul style="list-style-type: none"> <li>extend chimney</li> <li>adjust the patio wall height</li> <li>pierce patio</li> </ul> <p>Step 3.4: R – terminate</p> <ul style="list-style-type: none"> <li>erase labels</li> </ul>
--	---

Figure 5.3 – Stages, steps, and operations in the derivation of Malagueira houses

## 5.4 Context

In the grammar proposed for Siza's Malagueira houses, the initial shape is a rectangle with a label "Lot" representing the lot. The Malagueira housing plots are clustered together to form housing blocks. In most cases, these blocks are rectangular, but they might take other forms to adapt to the shape of curvilinear roads (please, see Chapter 4.) As a result, if the typical Malagueira lot has the front side facing the street and the remaining sides surrounded by houses, in other lots the remaining sides might be bordered by streets, within certain limitations. For instance the house cannot be bordered by streets on all the sides as detached houses are not permitted. The type of surroundings define the urban context of the lot and, taking into account the limitations mentioned above, there is a total of four possible contexts, with different symmetry properties. (Figure 5.4) The context impacts the functional organization of the house by restricting the number of elevations that can have openings, as shown in Figure 5.5 for types Ac and Ae.

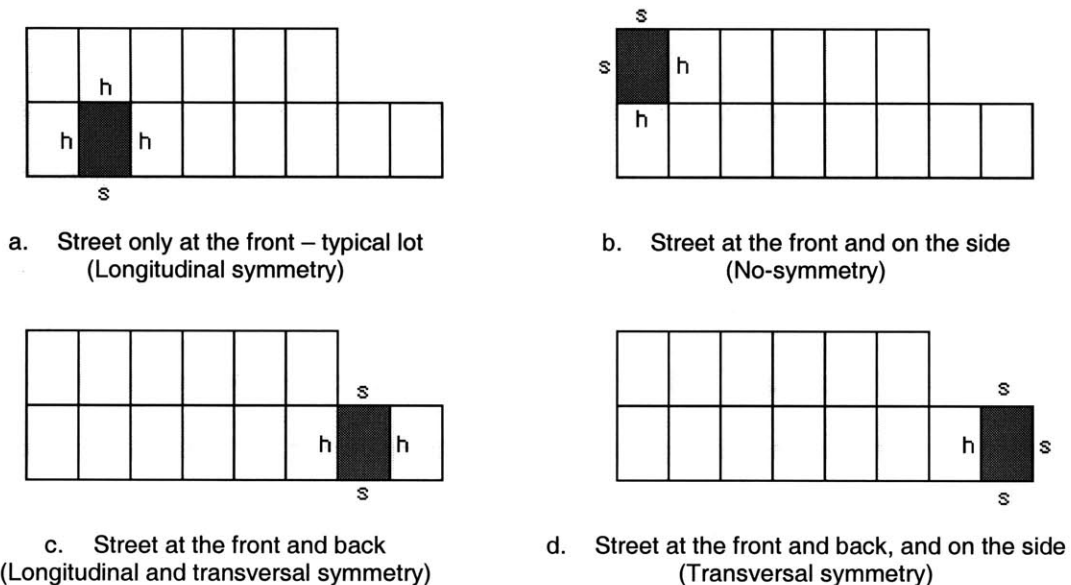


Figure 5.4 – Urban context of Malagueira lots.

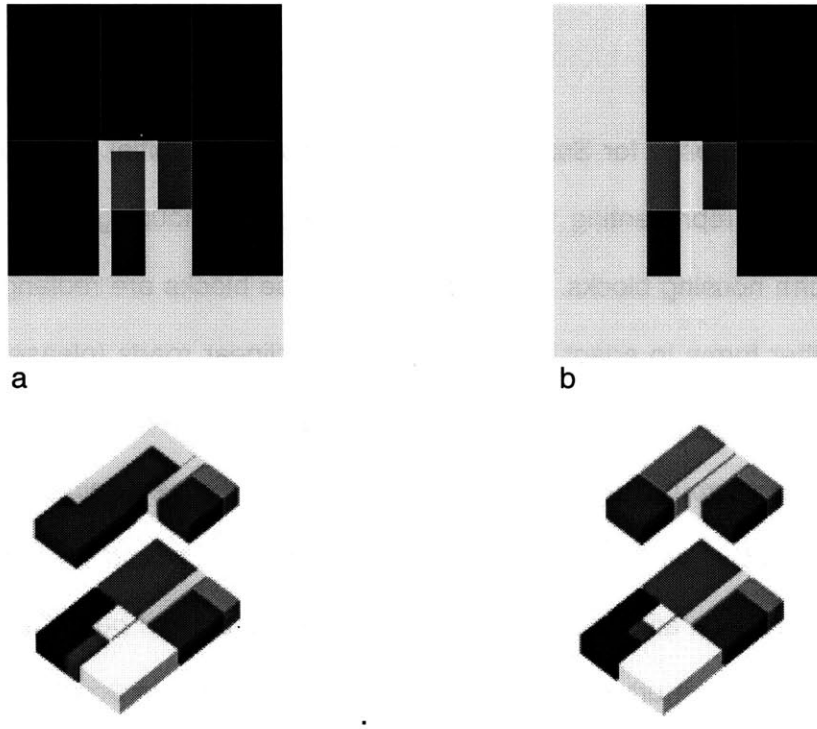


Figure 5.5 – The urban context impacts the functional organization: context a required a long circulation bordering the outer wall in subtype Ac, whereas context b permitted a short circulation bordering the patio wall in subtype Ad.

## 5.5 Composition: dissecting rectangles

The compositional strategy of a Malagueira house is based on rules for manipulating rectangles representing rooms. These rules include rules for dissecting, connecting, and extending rectangles, as well as rules for assigning and changing the functions associated with them. Dissection is the primary compositional principle. Figure 5.6 shows some of such rules and their extension in a simplified representation so that we can focus on the overall strategy captured by the grammar. In this simplification, lines represent walls, and rules are shown simply in 2D.

The labels "fn" denote the functions of the rooms that the rectangles and other shapes represent. The dot • is a label that identifies the last line placed and indicates on which side the next dissection may occur: on both sides (Rule A) or on only one side (Rule B). In rules A and B, dissections are perpendicular to the bigger side of the rectangle and to the previous dissection, whereas in Rule C it is perpendicular to the smallest one and parallel to the previous dissection. Rule D deletes the label •, preventing further dissections. Rule E concatenates two adjacent rectangles to form a larger room. Rule F, extends a room at the expense of an adjacent one. Rule G assigns a function to a room. Finally, Rule H permutes the function of two adjacent rooms.

In addition to rectangular dissections, the prototypical Malagueira designs included diagonal dissections, although with certain limitations. Rules I and J dissect a rectangle by tracing lines that establish 30 and 60 angles with its longer and smaller edges, respectively. The result of such dissections are right triangles and trapezoidal shapes. To deal with these shapes other rules are required in the grammar. Rule K dissects a triangle, in the case shown by tracing a line perpendicular to its longer side. Rule L dissects a trapezoid in a similar manner. None of these shape can be dissected by diagonal lines, thus preventing further deviation from rectangular shapes. Rules M and N concatenate a rectangle with an adjacent triangle and trapezoid, respectively. The use of non-perpendicular dissections was limited to the two prototypical designs because dwellers did not like non-rectangular rooms, and so, Siza avoided them in subsequent designs. These rules are used to derive the basic functional organization of the floor plan, as shown in Figure 5.7 for the first floor of subtypes Ab and Bb, the two prototypical designs.

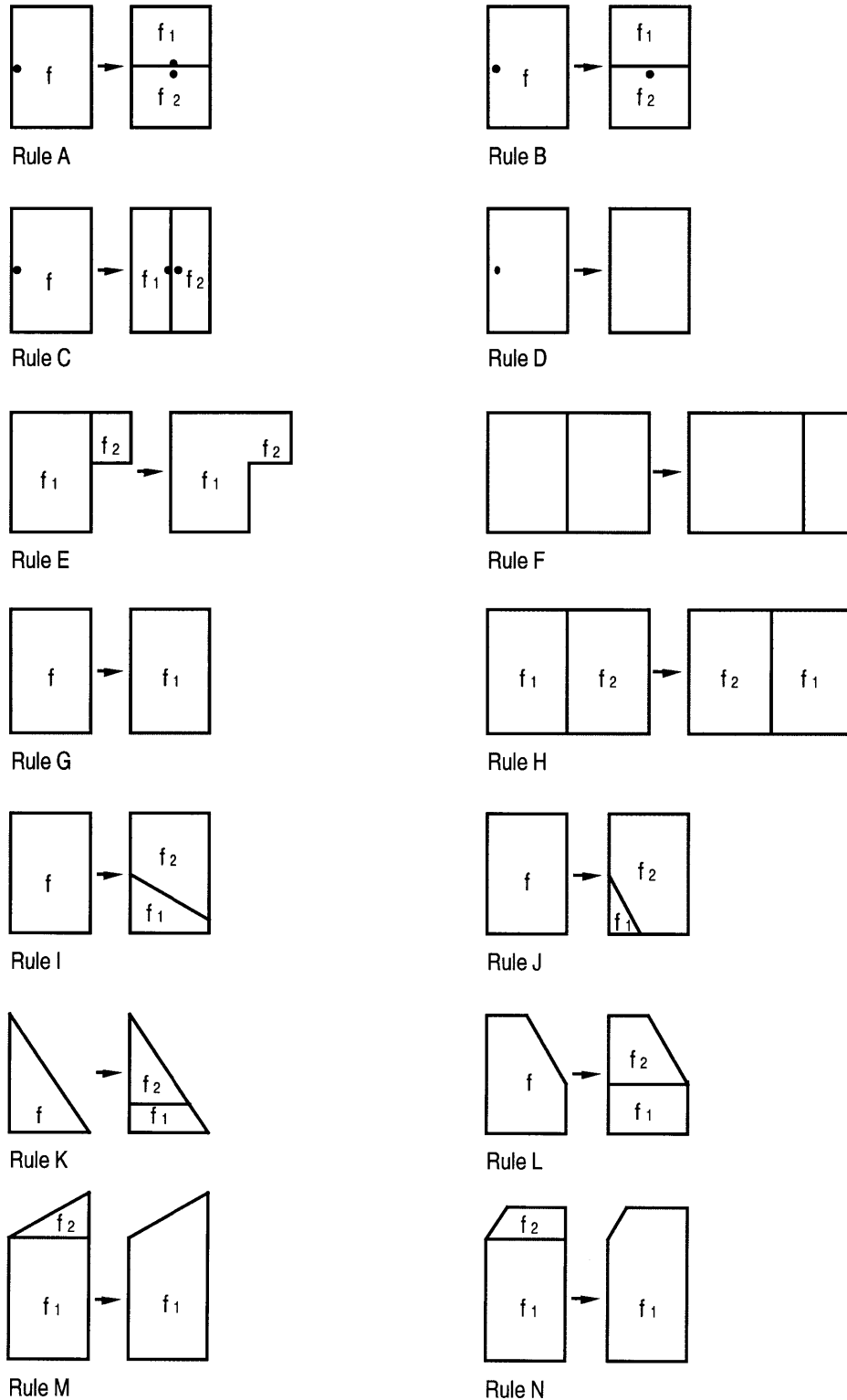
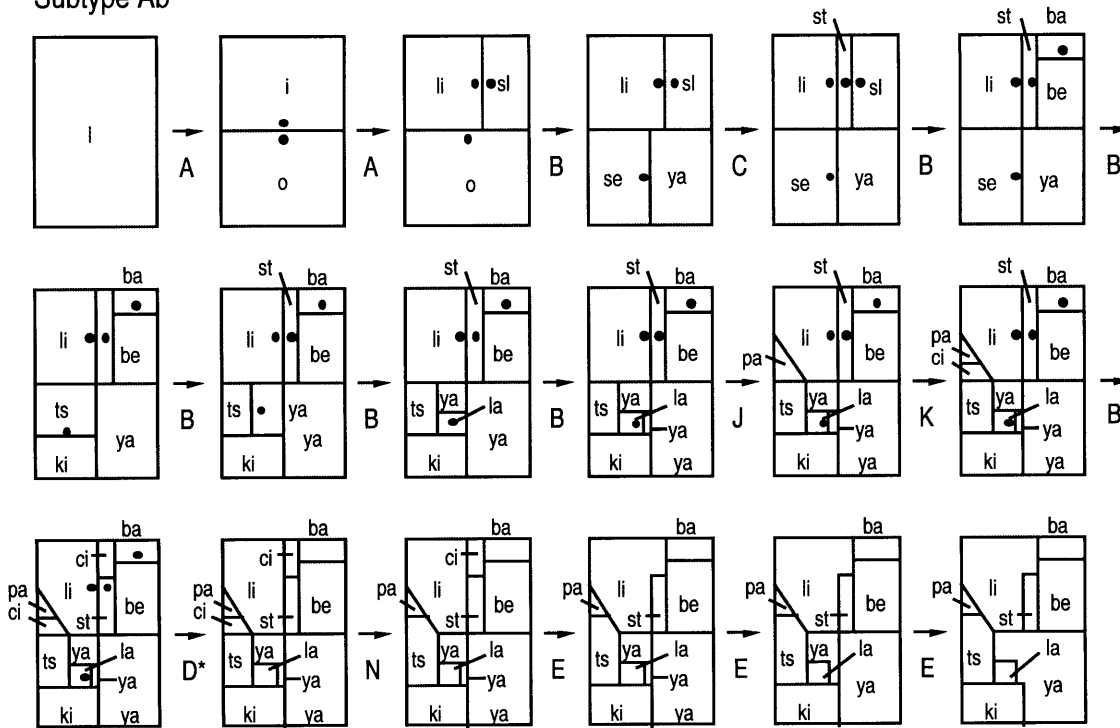


Figure 5.6 – Subset of simplified Malagueira houses compositional rules: rules for perpendicular dissection (A, B, and C), connection (E), and extension (F) of rectangles; rules are for deleting a marker (D), assigning a function (G), and permuting functions (H); and rules are for diagonal dissection of rectangles (I and J), perpendicular dissection of triangles and trapezoids (K and L), and for concatenating rectangles with triangles and trapezoids (M and N).



### Subtype Ab



### Subtype Bb

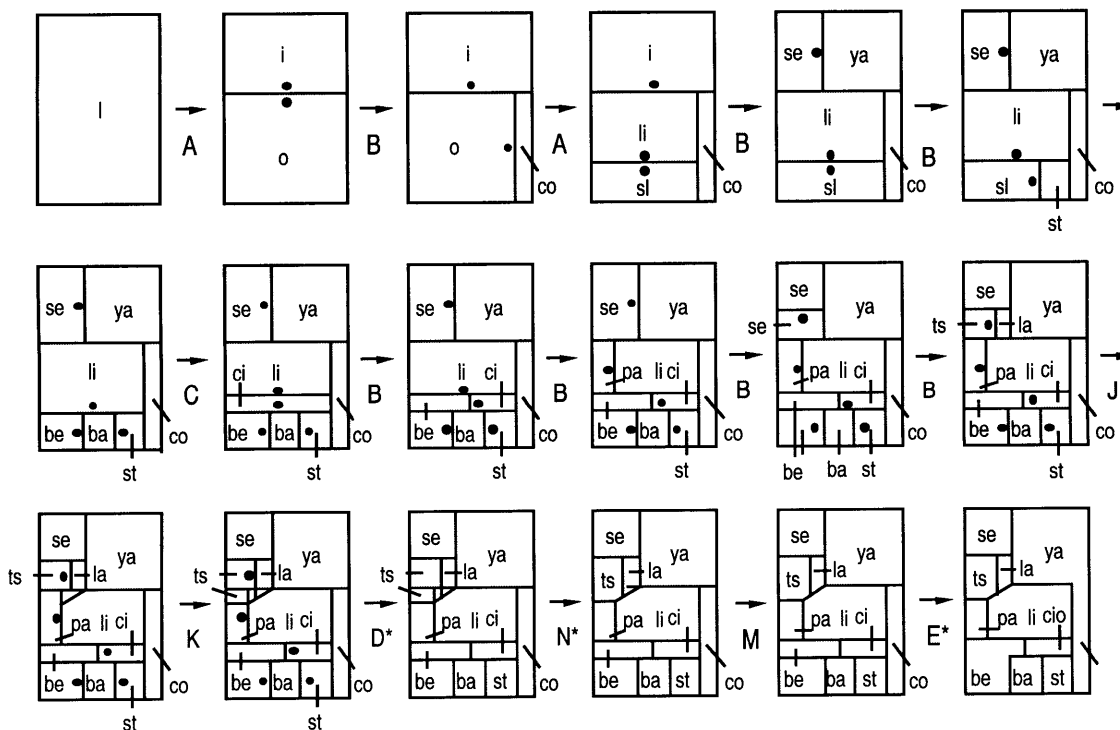


Figure 5.7 - Derivation of the 1st floor functional organization of subtypes Ab and Bb. Key: I – lot, i – inside zone, o – outside zone, li – living zone, sl – sleeping zone, se – service zone, ya – yard zone, be – bedroom, ba – bathroom, ki – kitchen, ts – transitional space (dining), la - laundry, pa – pantry, ci - circulation, st -stairs. The asterisk means that a rule was applied several times.

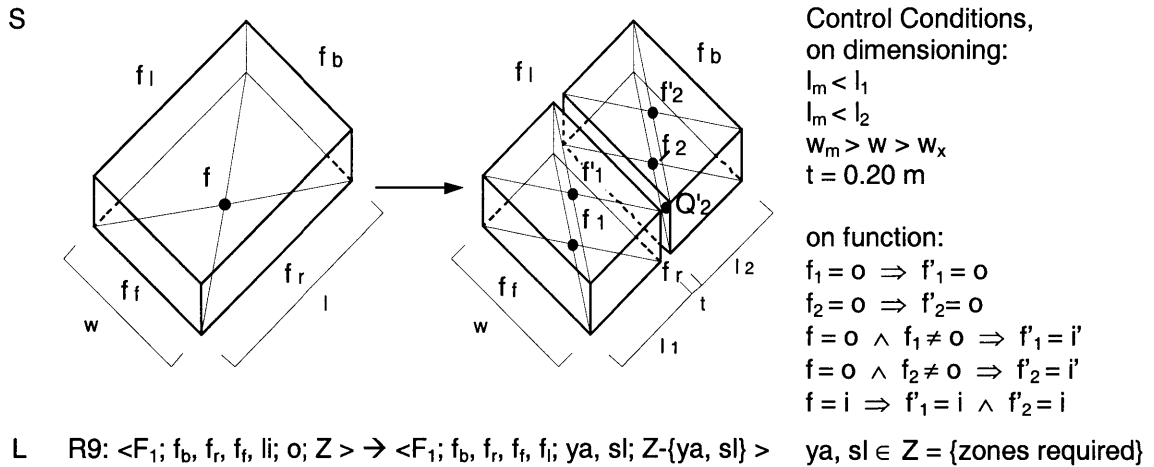


Figure 5.8 - Rule for dissecting the outside zone into yard and sleeping zones shown in the cartesian product of algebras  $U_{13}$  and  $V_{03}$ . The  $\times$  thin lines are visual aids to point out the location of the labels. Dashed lines represent hidden lines.

## 5.6 Function

### 5.6.1 First floor functional organization

As a way of facilitating the spatial and the mathematical understanding of the grammar rules, a detailed rule is shown in Figure 5.8, in which the plans are integrated into a single axonometric view in the cartesian product of the algebras  $U_{13}$  and  $V_{03}$ . The parametric rule shown is 'Rule 9: dissecting the outside zone into yard and sleeping zones.' As all the rules in the grammar, it has a shape part (S), a label part (L), and a set of control conditions on functional and dimensional aspects. As said previously, the shape part is used to specify the compositional strategy. In Rule 9, the shape part is generic as it is shared by several dissecting rules.

The label part is mainly used to deal with the contextual requirements and the functional strategy involved in the generation of Malagueira houses. The generic format of the label part of a dissecting rule has the form  $R_i: \langle F_n; f_b, f_r, f_f, f_i; f; Z \rangle \rightarrow \langle F_n; f_b, f_r, f_f, f_i; f_1,$

$f_2; Z - \{f_1, f_2\} >$ . In this expression,  $R_i$  is the rule number, such as R9 for Rule 9. The label  $F_n$ ,  $n \in \{1, 2, 3\}$ , indicates the stage of the derivation to which the rule applies. In Rule 9,  $n = 1 - 1^{st}$  Floor, which means that the rule only applies to the generation of the first floor. The labels  $f_b$ ,  $f_r$ ,  $f_f$ , and  $f_l$  identify the functions associated with adjacent rectangles at the back, right, front, and left side of the rectangle currently considered for dissection.

These labels, coupled with conditional statements are used to express adjacency requirements, thus determining the topology. In Rule 9,  $f_l = li$  – living room, which restricts rule application to finding in the evolving design a rectangle adjacent to the living room. The label  $f$  identifies the function currently associated with the rectangle being dissected (in R9,  $f = o$  – outside zone), whereas labels  $f_1$ , and  $f_2$  identify the function of the resulting rectangles (in R9,  $f_1 = ya$  – yard zone,  $f_2 = sl$  – sleeping zone). Together with conditional statements these labels specify the type of possible functional dissections (in R9,  $ya, sl \in Z$  – the set of required zones). Each time a dissecting rule is to be applied, the zones allocated by the rule are retrieved from the set of required zones. Once the rule is applied and the zones are created, they are deleted from this set (in R9,  $Z - \{ya, sl\}$ ), preventing further allocations in subsequent steps of the generation.

Other control conditions specify dimensional constraints (e.g.  $w_m < w_{sl} < w_x$ ,  $l_m < l_1$ ,  $l_m < l_2$ ). Namely, they assure that the dimensions of the zone to be dissected are such that they permit the allocation of the intended zones. In other words, allocation takes place only when the dimensions of the new zones can be within a certain range. This range was established after dimensional analysis of the existing designs and it is shown in

Table 5.1. The result of allocating functional zones is a basic pattern of the 1st floor layout.

The allocation of rooms within zones, including the allocation of the staircase, proceeds in a similar fashion. The rooms in the set of required rooms are specified before the generation starts and form the housing program. Computation terminates when all the rooms are allocated and the functional organisation is defined. The range of dimensions for rooms follows existing regulations as the analysis of drawings showed that Siza respected the regulations, without further constraints.

**Table 5.1 – Range of variation of the dimensions of the functional zones**

Minimum Values													
Zone	Corridor	Yard			Services Zone			Living Zone			Sleeping Zone		
Dimensions	$w_m$	$w_m$	$l_m$	$a_m$	$w_m$	$l_m$	$a_m$	$w_m$	$l_m$	$a_m$	$w_m$	$l_m$	$a_m$
	(m)	(m)	(m)	(m <sup>2</sup> )	(m)	(m)	(m <sup>2</sup> )	(m)	(m)	(m <sup>2</sup> )	(m)	(m)	(m <sup>2</sup> )
T1	1.10	5,01	6,68	33,44	2,59	6,68	17,29	3,80	4,80	18,24	2,80	4,80	15,68
T2	1.10	3,70	5,80	21,46	2,80	4,80	13,44	4,50	4,80	21,60	2,80	5,60	15,68
T3	1.10	3,70	4,80	21,00	2,59	4,80	12,48	3,20	4,80	18,24	2,70	4,80	15,68
T4	1.10	3,70	4,80	21,00	2,60	4,80	12,48	3,20	4,80	18,24	2,70	4,80	15,68
T5	1.10	3,70	4,80	21,00	2,60	4,80	12,48	3,20	4,80	18,24	2,80	4,80	15,68
Maximum Values													
Zone	Corridor	Yard			Services Zone			Living Zone			Sleeping Zone		
Dim.	$w_m$	$w_x$	$l_x$	$a_x$	$w_x$	$l_x$	$a_x$	$w_x$	$l_x$	$a_x$	$w_x$	$l_x$	$a_x$
	(m)	(m)	(m)	(m <sup>2</sup> )	(m)	(m)	(m <sup>2</sup> )	(m)	(m)	(m <sup>2</sup> )	(m)	(m)	(m <sup>2</sup> )
T1	--	5,80	7,60	44,08	2,59	6,68	17,29	4,70	5,60	26,32	3,80	5,60	18,24
T2	--	6,70	7,80	52,26	3,10	4,80	14,88	4,80	5,80	27,26	2,90	5,80	16,82
T3	--	5,01	6,70	33,44	3,90	6,68	25,37	4,70	6,50	27,26	3,80	6,70	22,10
T4	--	4,90	6,70	32,83	3,90	6,68	25,37	4,70	6,50	27,26	3,80	6,70	22,10
T5	--	4,80	6,68	25,37	3,90	6,68	25,37	4,70	6,50	27,26	3,80	6,50	22,10

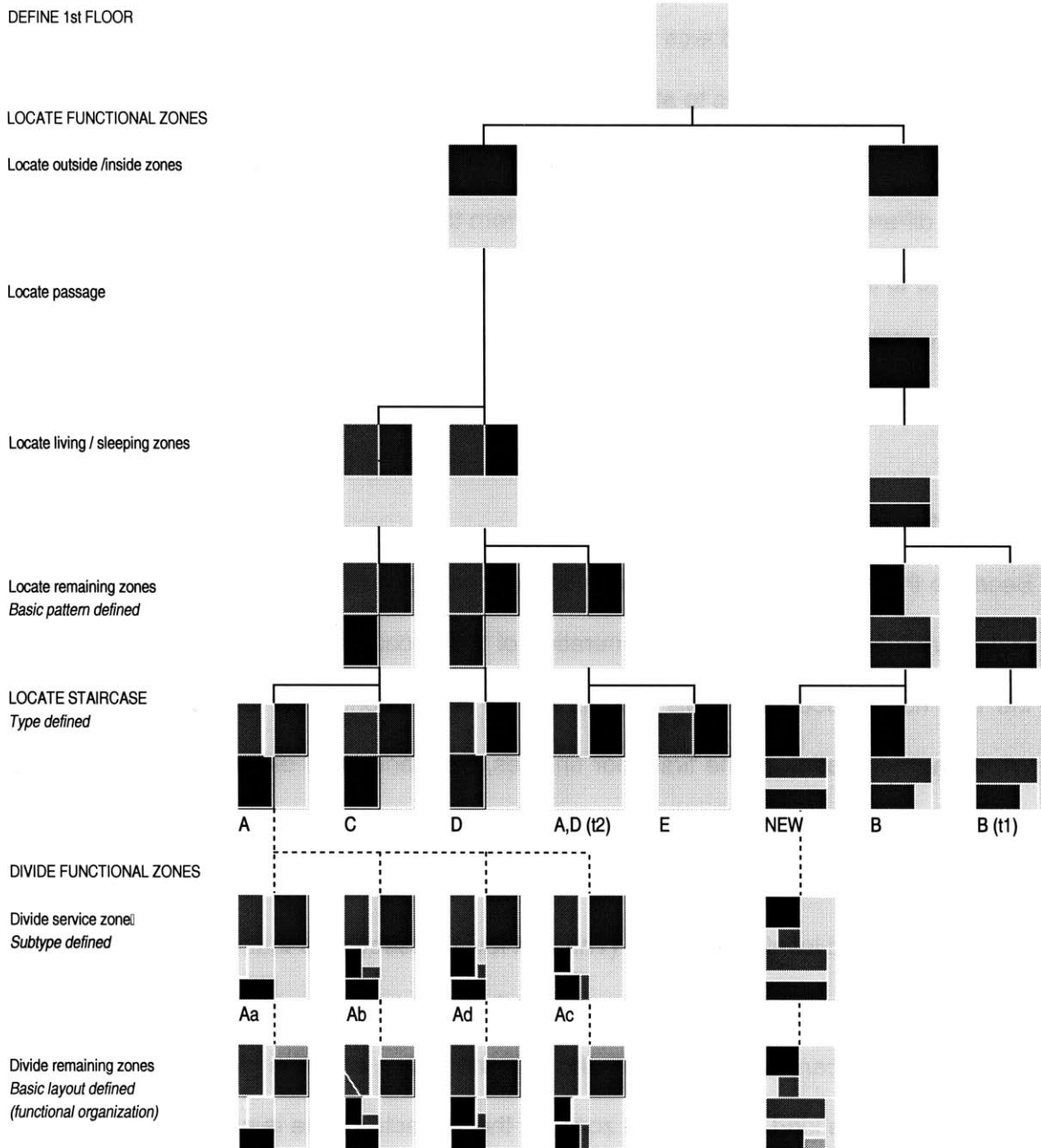


Figure 5.9 – Partial tree diagram showing the derivation of basic patterns, types, subtypes, and layouts by applying rules for locating functional zones, locating staircase, and dividing zones into rooms. The application of rules to introduce details and openings is not shown. The basic patterns are reduced to topological patterns just to highlight the commonalities among types. The diagram includes designs in the corpus and a new design presented in the next chapter.

The steps involved in the definition of the first floor's functional organisation are diagrammed in Figure 5.9. The diagram takes the form of a tree where nodes represent the state of the design and arcs represent the application of rules. The tree illustrates how the application of rules to allocate functional zones, generates the five basic topological patterns (see Section 5.8) behind the houses in the corpus. It also shows how the different types in the corpus derive from these patterns by a different application of the rule to locate the staircase. Finally, it shows that subtypes differ from one another in small variations of the layout caused by different applications of the rules for dividing zones.

### **5.6.2 Articulation between floors**

Because the structural system used is a load-bearing wall system, the generation of the first floor largely determines the generation of the second floor. While the generation of the first floor proceeds, labels are placed on the second floor and on the elevation. When the generation of the first floor finishes, the generation of the second floor starts using these labels as beacons. Consider the rule in Figure 5.8. Label Q'2 is placed on the second floor to indicate where the dissection took place. The exponent ' indicates that it is a second level dissection. (The first level dissection is that of the lot into outside and inside zones.) The index 2 refers to the floor. Labels  $f_1$  and  $f_2$  determine the kind of the zones that can be created on the second floor, in terms of indoor or outdoor, depending on the kind of dissected zone. If the dissected zone is an outside zone, as in Figure 5, two things can happen. If the resulting zone is a yard zone, then the label is o, meaning that the zone above will become an outside zone. If the resulting the zone is a zone other than the yard zone, then it is i'. Finally, if the dissected zone is an inside zone, then the label is i. Other circumstances will determine to which kind of rooms will i

and  $i'$  zones give origin (terraces or indoor rooms). In the case of the rule for dissecting the outside zone into yard and sleeping zones shown in Figure 5,  $f_1 = o$  and  $f_2 = i'$ .

## 5.7 Structure

As described in Section 4.3.2, the structural system of Malagueira houses is based on load-bearing walls. Recall that this system impacts the design by constraining the thickness and height of walls, the span between load-bearing walls, and the need for load-bearing walls on upper floors to be aligned with walls on lower floors. Such structural constraints are encoded into the grammar in a number of different ways. Constraints on thickness and span are encoded into dissecting rules by placing conditions on the thickness and location of dissecting walls. Consider, for instance, Rule R9 in Figure 5.8, which divides the inside or outside zones into functional zones. Because at this step in the derivation the dissecting wall is a load-bearing wall, it requires its thickness to be 0.20 m, and it limits its location so that a 6.0 m maximum span is respected. Constraints on wall alignment are encoded into dissecting rules by requiring dissections on upper floors to be aligned with dissections on lower floors through the placement of appropriate labels as described in the previous section. The influence of the span between load-bearing walls on the thickness of the slab they support is encoded by using the span value to retrieve the required thickness from a table when placing the slab, later in the derivation. Constraints on wall height are encoded simply by choosing a standard value that is a multiple of the concrete block's height and higher than the minimum floor height. Other structural constraints operate on rules for concatenating adjacent rooms and on rules for piercing openings to prevent the deletion of large extensions of load-bearing walls.

## 5.8 The universe of design solutions

Determining the universe of solutions is crucial to determine how useful it is to invest time and other resources in the development of a grammar and an interpreter, and to choose an appropriate control mechanism to search for a solution that matches given requirements. As we shall see, the universe of solutions of the Malagueira grammar is large enough to make the development of the grammar and the interpreter worthwhile. The identification of the universe of solutions, however, presents two difficulties. One is related to the design of the grammar itself, and the other is the difficulty in counting the number of solutions, as discussed below.

### 5.8.1 Designing the grammar

Designing the grammar poses a paradox. On the one hand, one needs a grammar that generates a large set of design solutions to increase the potential of generating customized designs. On the other hand, one wants to make sure that the grammar only generates designs in the Malagueira style and that a solution can be found in practical time. The analysis of Siza's designs provided some clues on how to solve the paradox. Consider the derivations of the two prototypical designs in Figure 5.7. The generation of the first floor of Type Ab reveals an obsessive, recursive use of dissections perpendicular to the last dissecting lines (Rules A and B). In Type Bb, the use of rules is slightly more relaxed as it also uses dissections that are not perpendicular to the previous level dissection. The generation of the second floors and the layouts of the remaining types shows an increasingly relaxed use of dissections, with an alternation of different rules. This supports the idea of Type Ab as a canonical type that established the basic rules of the grammar. The need to generate new, diverse houses then caused



the breaking of this initial canon in order to enlarge the universe of designs that the grammar could generate. This procedure provided a basis to develop the set of rules of the proposed grammar. In fact, the main goal behind the development of the grammar was to provide a coherent but large set of designs to enable customization within the grammar.

Three steps were then followed in the design of the grammar. The first step was to develop the exhaustive set of rules that could be derived from the compositional principles of dissecting and concatenating rectangles. This principle was followed, for instance, in the rules for perpendicular dissections by eliminating the marker •, in the rules for diagonal dissections by including all the dissections that did not involve dissecting diagonally to another diagonal dissection (Rules 42-47), and in the rules for concatenating rectangles by adopting of a very general rule for concatenating spaces (Rule 9).

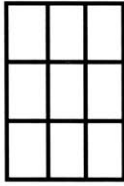
The second step was to limit such an exhaustive set of rules whenever it seemed that it would oppose Siza's compositional rules. This principle was followed, for instance, in the restriction of dissections perpendicular to the smaller edges to those few cases found in the corpus, such as the dissection of the patio and the living zones. Another example was the limitation of possible divisions of the lot into functional zones as to respect what seemed to be the functional organization intended by Siza. Thus, in the set of rules proposed at this stage, it is the functional meaning assigned to shapes and restrictions on the dimensions of the spaces they represent that mostly determine how shapes can be dissect and concatenated.

Following these steps requires one to answer questions such as, did Siza intentionally excluded a certain rule? Would he consider using it had he reached the number of different houses that included rules could generate? Answering these questions implies a subjective judgement, and to overcome this problem a third step was taken. This third step was to generate new designs with a closed set of rules and then ask Siza whether he considered them to be in the grammar as described in Chapter 6. A set of rules is closed if it is possible to generate complete designs within the set. These three steps roughly are part of the methodology described in Chapter 2 based on the undertaking of the descriptive, analytic, synthetic, and goal tests.

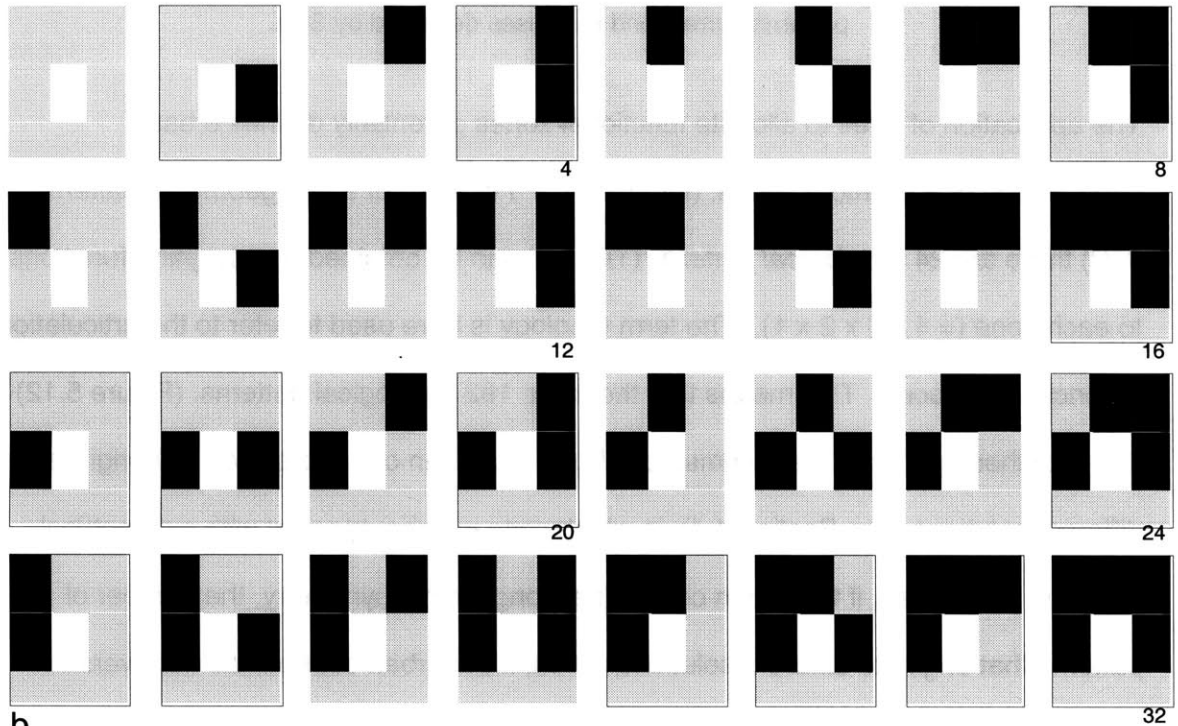
### **5.8.2 Estimating the number of solutions**

An exact counting of the universe of solutions is difficult to perform, but it is possible to obtain a good idea of its size by estimating the upper and lower bounds of the interval that corresponds to steps one and two mentioned above, at each step in the derivation of designs.

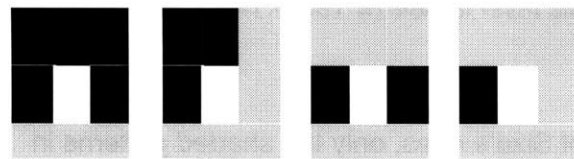
The Malagueira plan defines a grid of 8 x 12 m lots, which means that each plot is bordered by other 8 lots. (Figure 5.10a) The front three lots are always occupied by a street. The other five neighboring lots can be potentially occupied by a house or a street, which yields a total of 32 ( $2^5$ ) *context patterns* (CP). (Figure 5.10b) The application of urban planning rules, however, restricts the number of such patterns to the 14 patterns bordered by black lines. By considering that the corner lots have no impact on the functional organization of the central lot, and by eliminating symmetrical patterns, the number of context patterns is reduced to 4. (Figure 5.10c)



a



b



c

Figure 5.10 – Context patterns. The Malagueira plan is based on a 8 x 12 m grid (a). There are 32 possible context patterns that can be derived from a broad interpretation of Siza's rules (b). Of these, only the 14 patterns bordered by black lines are in accordance with a strict interpretation of such rules (b). By disregarding the corner lots and by eliminating symmetrical patterns, such patterns can be reduced to 4 patterns (c), which correspond to the contexts shown in Figure 5.4.

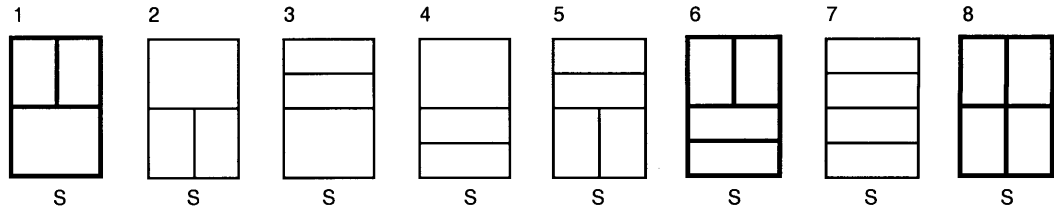


Figure 5.11 - The 8 geometric patterns that can be derived from Siza's dissection rules. Bold patterns correspond to houses designed by Siza.

The application of rules to allocate functional zones potentially defines 8 basic subdivisions of a lot into 4 zones. (Figure 5.11) For each of these *geometric patterns* (GP) there are 24 *topological patterns* (TP) that can be obtained by assigning functions to each zone ( $= 4 \times 3 \times 2 \times 1$ ). The term topology is here used to refer to the articulation of functional spaces. This means that there are 192 topological patterns. (Figure 5.12) Some of these patterns are symmetrical, but if the urban context is not, they originate different solutions (see Section 4.4), in which case we should count with symmetrical patterns. Otherwise, if the urban context has longitudinal symmetry, the number of patterns that originate different solutions is 144; if the urban context has transversal symmetry, this number is 96; and if possesses both symmetries, it is 52. Therefore, the upper bound of the interval of solutions at this level is 484 ( $= 192 + 144 + 96 + 52$ ).

Nevertheless, in the strictest interpretation of Siza's rules, only the shaded patterns in Figure 5.12 are in the grammar. The remaining patterns are ruled out either due to compositional or functional reasons. Patterns are ruled out for compositional reasons if they require the dissection of a rectangular zone by tracing a wall perpendicular to its smaller edge. Patterns in such case are those that result from geometric patterns 3 through 7, in which the patio has direct access from the street, and so there is no corridor to shorten the rectangle. Patterns are ruled out for functional aspects because they require the patio, the service, or the living zones to be located on the second floor;

because the patio is in the middle; because the patio's proportions do not comply with building regulations set in Table 4.3; or because they imply the dissection of the outside zone into living and patio zones. None of these situations is present in Siza's designs. Therefore, the lower bound of the universe of solutions is 108 (= 28 + 22 + 10 + 48).

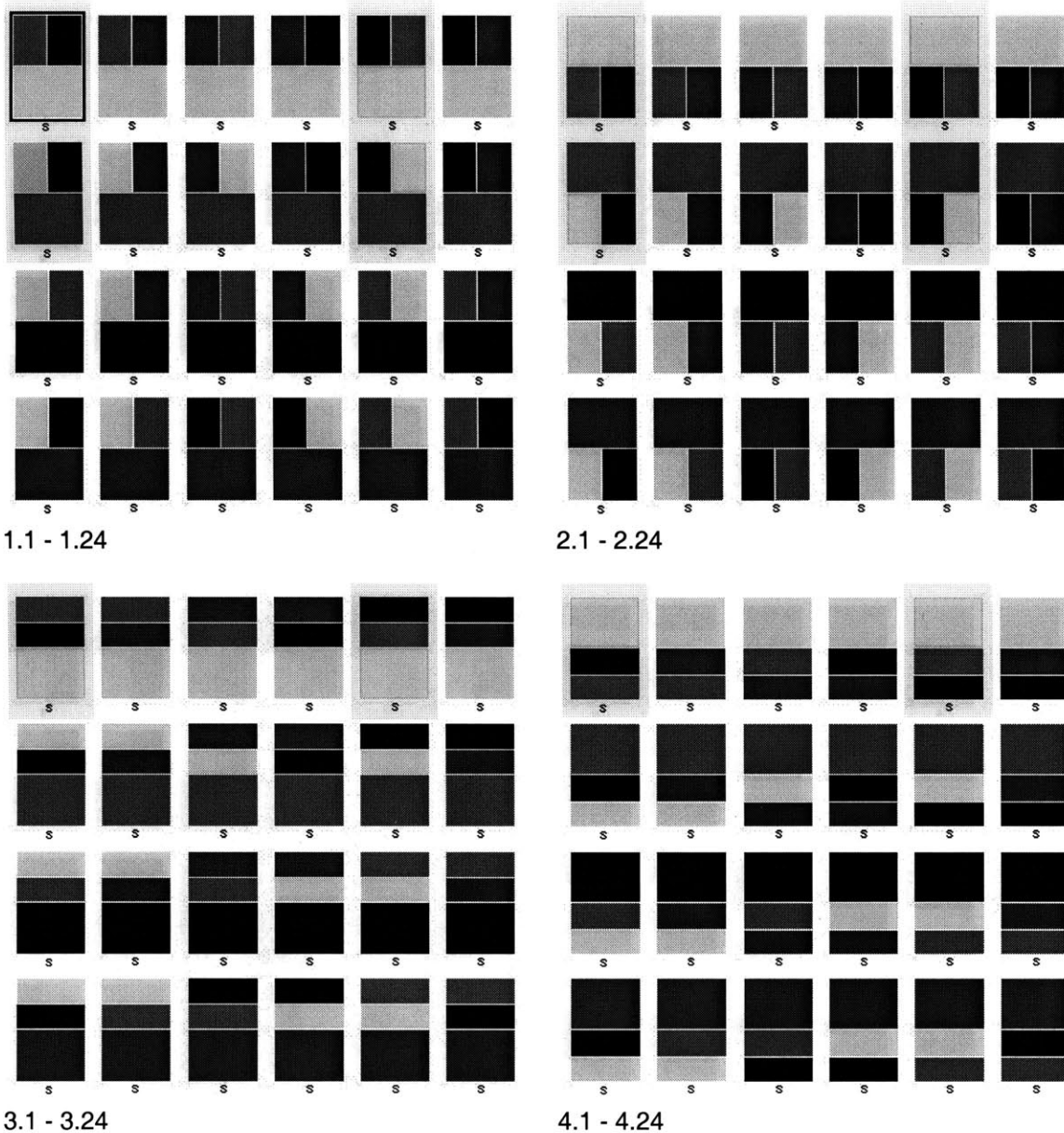
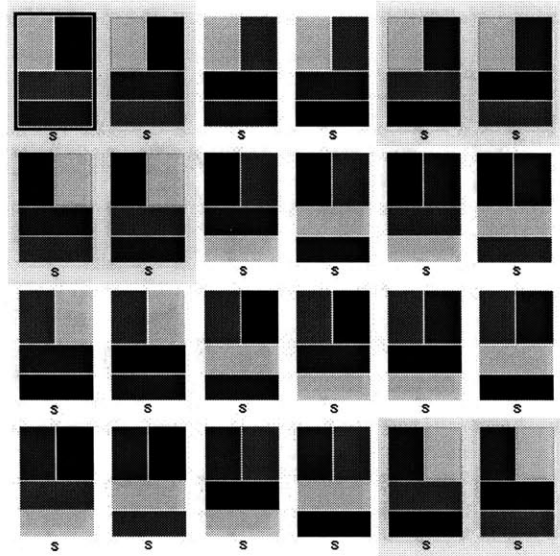


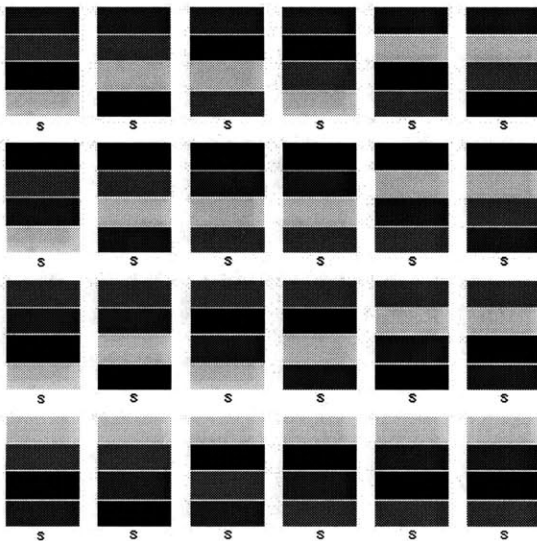
Figure 5.12 - The 192 topological patterns that can be derived from the 8 geometric patterns in Figure 5.10, following a broad interpretation of Siza's design rules. Pattern bordered by black lines correspond to houses designed by Siza. Shaded patterns are patterns considered in the grammar.



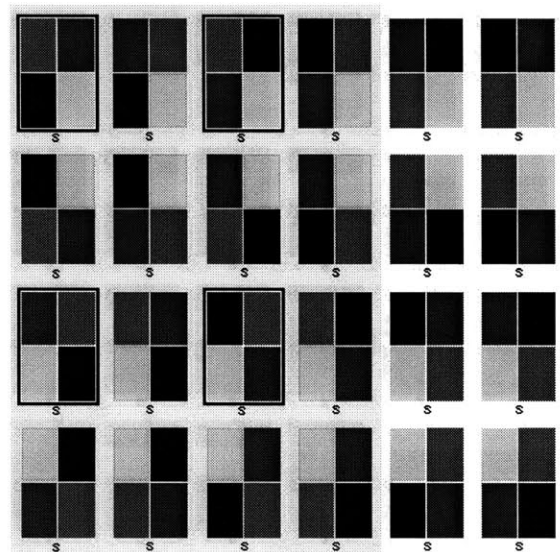
5.1 - 5.24



6.1 - 6.24



7.1 - 7.24



8.1 - 8.24

Figure 5.12 (continued) - The 192 topological patterns that can be derived from the 8 geometric patterns in Figure 5.10, following a broad interpretation of Siza's design rules. Patterns bordered by black lines correspond to houses designed by Siza. Shaded patterns are patterns considered in the grammar.

Consider now patterns with the staircase included. Figure 5.13 and 5.14 shows all the possible *stair patterns* (ST) for a location of the staircase in one of the functional zones (living in the case shown,) considering variables such as type of dissection that yielded the zone, the location and orientation within the zone, and the type of staircase. There are 12 possible ways of locating an L-shaped staircase in a rectangular zone, being 8 on the side wall and 4 in the middle. In addition, there are 24 possible ways of locating an U-shaped staircase. Therefore, there are 36 possibilities for each topological pattern, which yields a total on the order of about stair patterns (CP x GP x TP x FZ x ST). In the strictest interpretation of Siza's rules the number is considerably smaller. The staircase is always placed on the side wall of the functional zone, and never in the middle. An L-shaped staircase is always located in the living zone and in such a way as to minimize circulation. In addition, a U-shaped staircase is always located in the sleeping or service zones, and placed transversally. With these constraints the number possibilities is considerably reduced. There are 4 possible ways of locating L-shaped staircases in the living zone, and 8 possible ways of locating U-shaped staircases in the service and sleeping zones. The total number of stair patterns is, thus, 7680.

If the division of zones into rooms is also considered, the space of design solutions of Siza's design system for Malagueira becomes even larger. Consider, for instance, dissections perpendicular to the last dissection. The living and patio zones can be dissected into one (assignment), two, or three rooms and that the sleeping and service zones can be dissected into two up to five rooms. Then, the universe of solutions upper bound rapidly raises to about 20,250,000 ( $=3 \times 3 \times 5 \times 5 \times 90,000$ ).

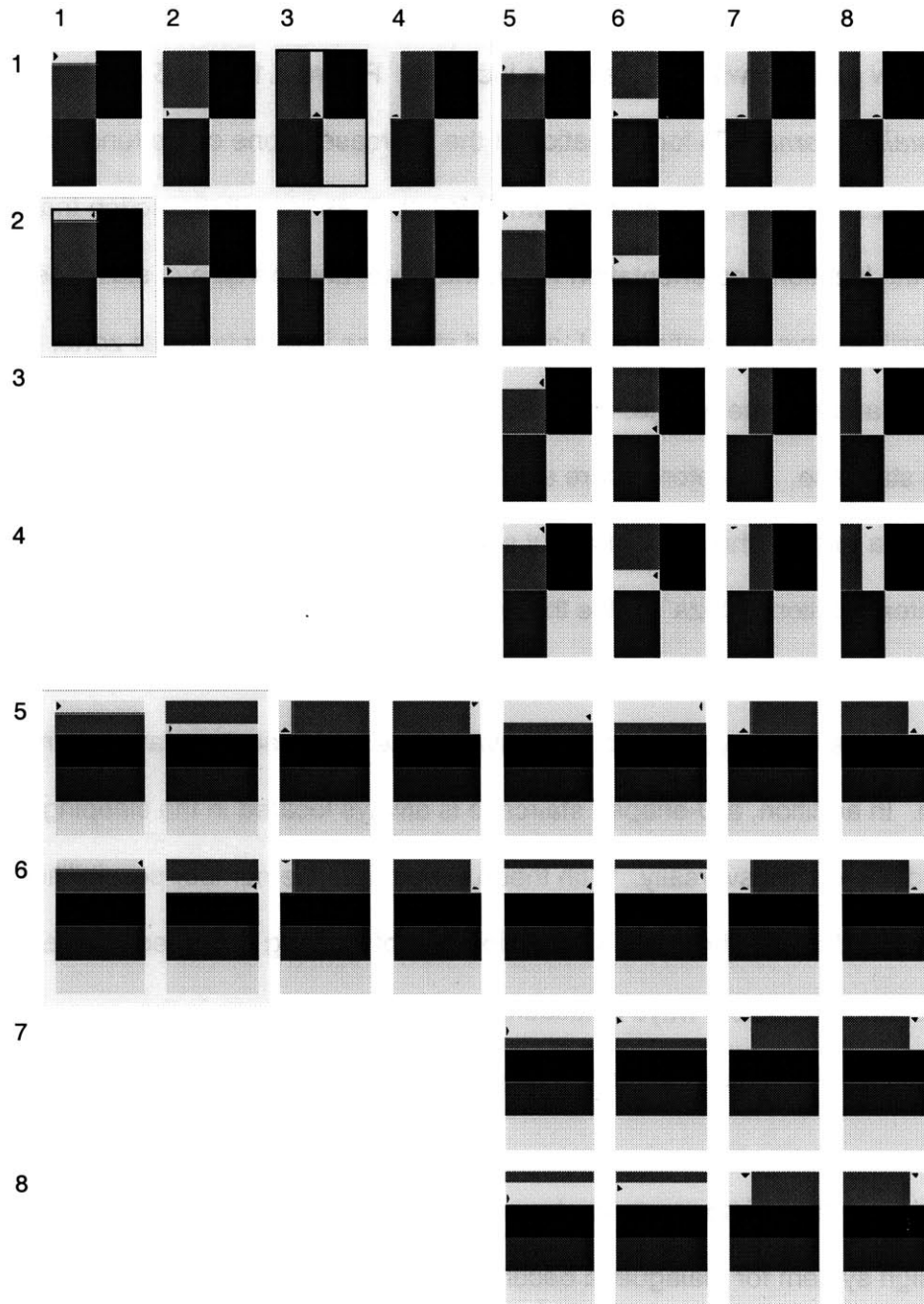


Figure 5.13 – Stair patterns with the staircase located on the side of a functional zone (the living in the case shown.) Rows 1-4: location in a “vertical” zone; rows 5-8: location in a “horizontal” zone. Columns 1-4: I and L-shaped staircase; columns 5-8: U-shaped staircase. Stairs are shown in yellow. The black arrow indicates the location of the starting step. Patterns bordered by black lines correspond to houses designed by Siza. Shaded patterns are patterns considered in the grammar.



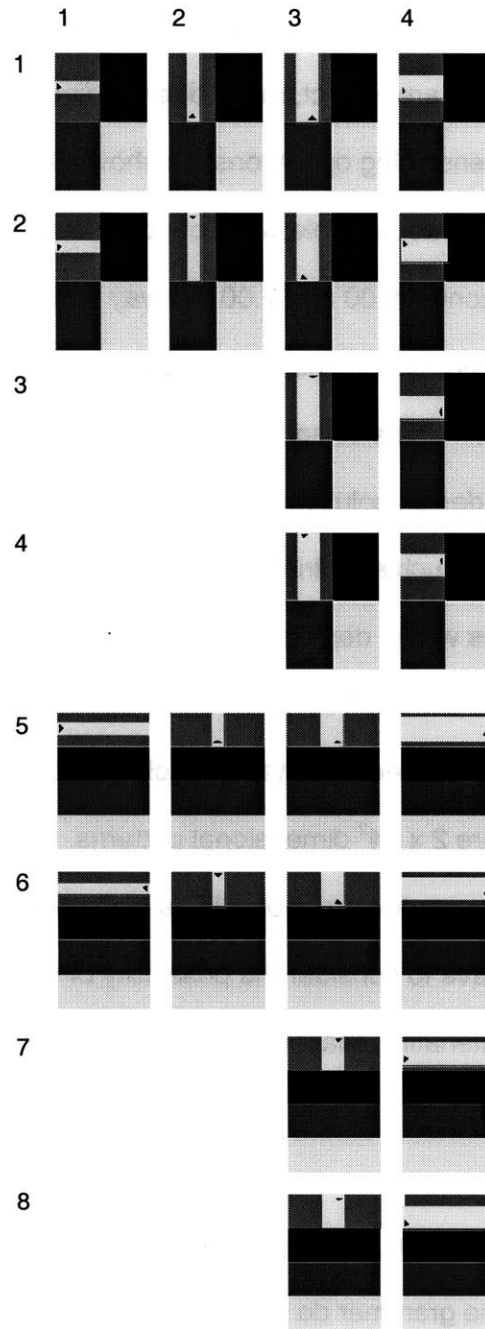


Figure 5.14 – Stair patterns with the staircase located in the middle of a functional zone (the living in the case shown.) Rows 1-4: location in a “vertical” zone; rows 5-8: location in a “horizontal” zone. Columns 1-2: I and L-shaped staircase; columns 3-4: U-shaped staircase. Stairs are shown in yellow. The black arrow indicates the location of the starting step. None of these houses in the corpus fall into this category and none is considered in the proposed grammar.

In the discussion above, the dimensioning of zones was not taken into account. For each topological pattern there are, in fact, many distinct *dimensioned patterns* that can be generated after two dimensioning operations, as shown for the houses in the corpus. (Figure 5.15) First, there are two possible ways of positioning the line that dissects the lot into inside and outside zones, 6.00 and 7.00 m away from the front of the lot (Figure 5.16), which automatically doubles the number of solutions. Then, consider the walls that divide the inside and outside zones into functional zones. Just for the sake of estimating the universe of design solutions, accept that such wall can be placed at 0.05 m intervals (Siza did not use such a restriction.) (Figure 5.17) The number of dimensioned patterns varies wildly, depending on the geometric pattern. For each topological pattern derived from geometric pattern 3, there are  $4 + 24$  dimensional patterns. On the other extreme, for each topological pattern derived from geometric pattern 8 there are  $2 \times 64^2$  dimensional patterns. The upper bound of the universe of solutions is, thus, on the order of billions ( $CP \times GP \times TP \times FZ \times SP \times 2 \times 64 \times 64$ ). But, then, we would have to consider the possibility of using rules for diagonal dissections, rules for concatenating spaces, let alone the rules for detailing the spaces, and the rules for the openings.

The separation between geometric, topological and dimensional patterns is merely analytical, as the rules of the grammar do not make such a distinction. Previous approaches to layout generation did propose first, to generate a valid topological solution, and then to dimension it. See, for instance, Mitchell, Steadman and Liggett (1976). We decided not to proceed this way because this was not Siza's procedure. Thus, the combination of geometric, dimensional and topological constraints in the same rule reflects Siza's way of designing.

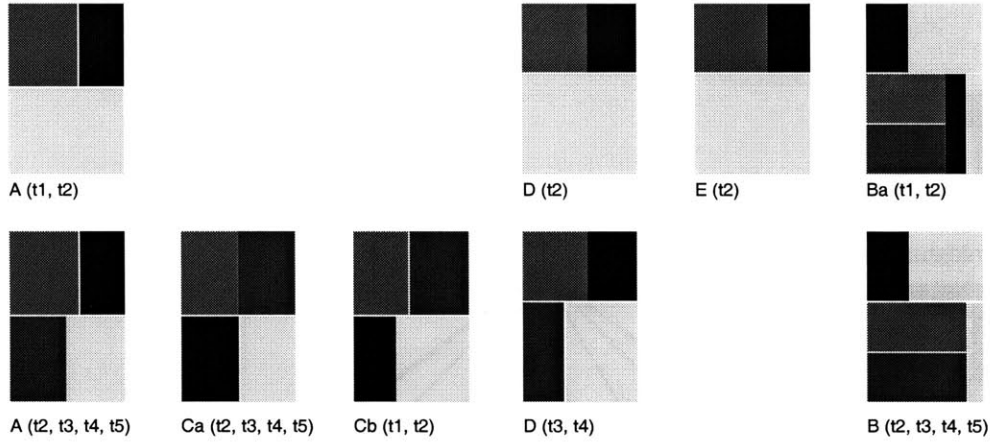


Figure 5.15 – Dimensional patterns of houses in the corpus. Pattern Ba, which corresponds to the preliminary study of subtype Bb, was not included in the grammar because the design was not settled yet (division into four zones not respected.)

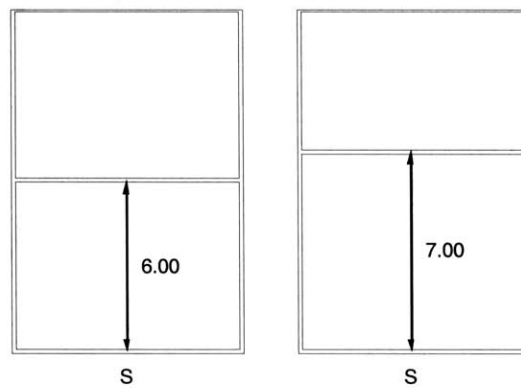


Figure 5.16 - The two possible ways of positioning the line that dissects the lot into inside and outside zones.

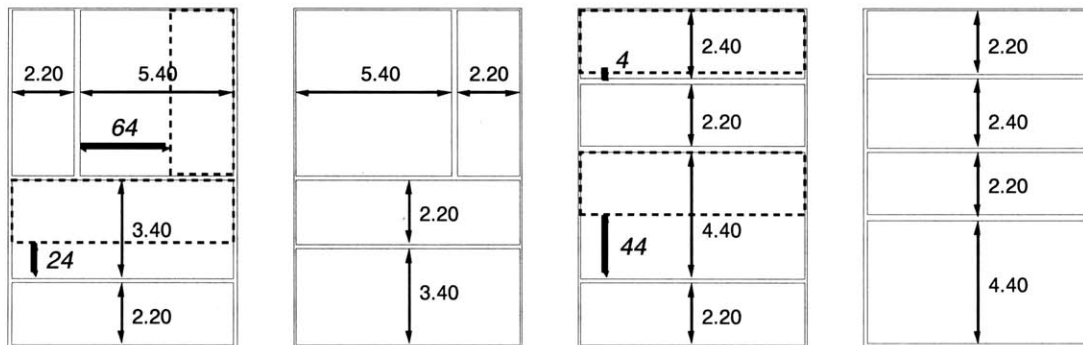


Figure 5.17 - Possible positions of the walls that divide the inside and outside zone into functional zones, considering existing regulations that requires rooms to be at least 2.20m wide. Numbers in italics represent the number of positions.

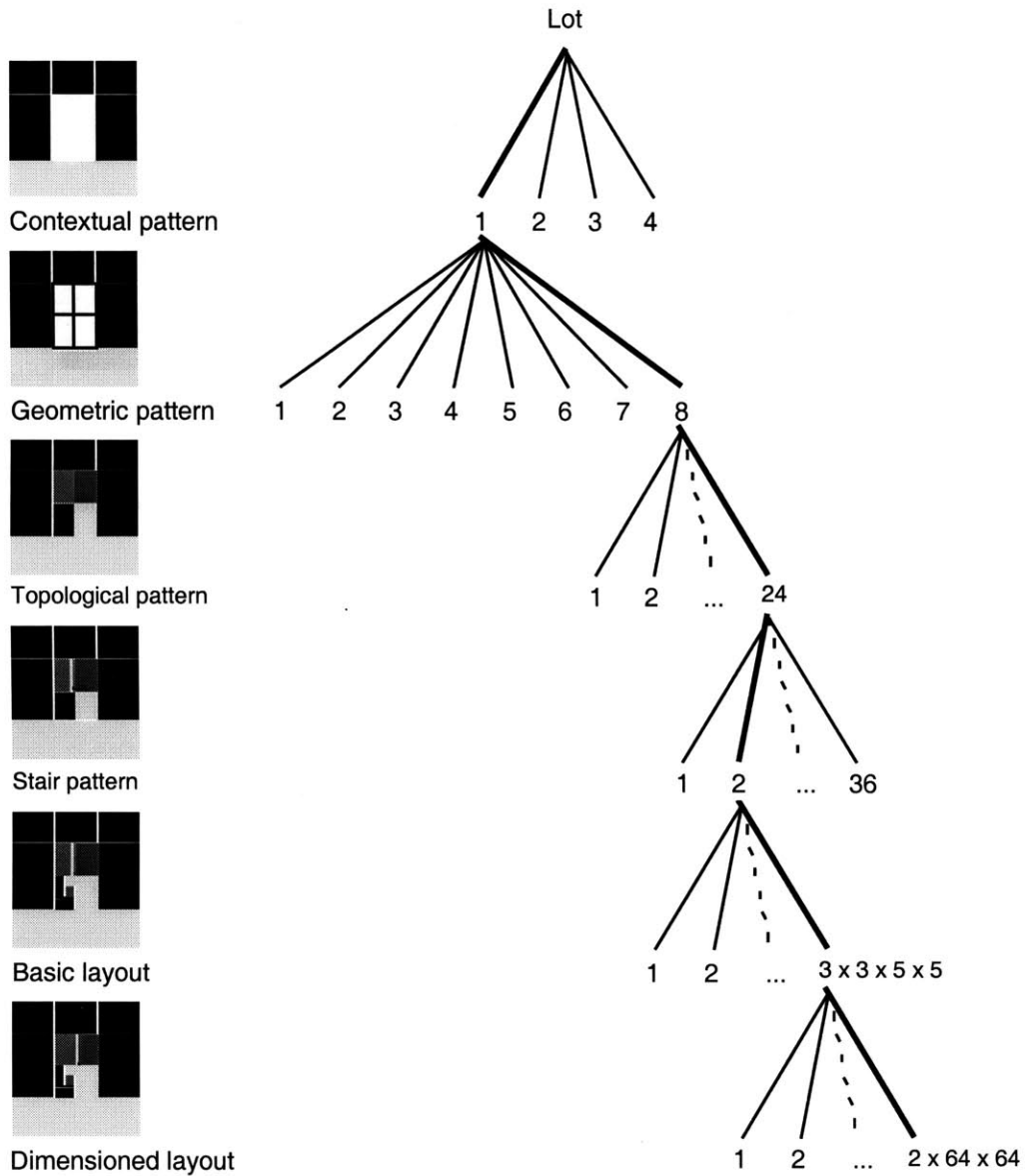


Figure 5.18 – Tree representing the universe of Malagueira solutions.

In summary, the universe of possible solution is potentially enormous, even at its lower bound. Such a universe is represented in Figure 5.18 in the form of a search tree. Note how the branching factor leads to the exponential growth of solutions as the tree moves in depth. This will require the use of an appropriate search method to overcome this feature.

## 5.9 Rules

The description of the different stages and rules is provided below in greater detail. To simplify the representation of rules included in the next section, all the shapes in axonometric descriptions and empty labeled shapes in the remaining descriptions are not represented.

### 5.9.1 Stage 0: Introduce initial shape

The derivation starts with the introduction of an initial, rectangular shape representing the lot. The width and the length of the Malagueira lots considered in the grammar are constant and equal to 8.0 m and 12.0 m, respectively. Such dimensions are close to those found in a pre-existing nearby settlement (Testa, 1984) but they clearly fit the courtyard typology of the houses as they permit flexibility of configuration without wasting area, an important concern in the design of social housing. The length is such that it allows a respect for the 45° rule that regulates the distance between walls of confronting buildings in the Portuguese building code RGEU<sup>1</sup>. The width, in its turn, permits a maximum of three rooms facing the street or an eventual backyard, given the minimum width (2.20 m) that each room is required to have according to the same code. Such a configuration permits the design of layouts with five bedrooms as required in the larger houses.

A single rule, Rule 0, applies at this stage. This rule introduces the initial shape and adds a set of labels. The labels *s* and *h* around the edges of the lot that tell whether they border a street or a house, respectively. One of the smaller sides always faces a

---

<sup>1</sup> RGEU: *Regulamento Geral das Edificações Urbanas* (General Regulations for Urban Buildings)

street and at least one of the larger sides borders a house. These labeling constraints express the fact that at Malagueira the lots are always perpendicular to a street and that there are no blocks made of one or two lots. The labels  $Q_1$  specify that there are only two ways in which a lot can be split into two halves in a subsequent step of the computation to allocate the patio and the house. Finally, F0 is a state label placed at the origin that indicates when the computation can proceed on to the next stage.

### **5.9.2 Stage 1: Define the first floor**

The definition of the first floor plan goes through six different steps: start (S), locate functional zones (Z), defining circulation scheme (C), dividing zones into rooms (R), introduce details (D), introduce openings (O), and terminate. Such a division into steps is merely descriptive, to make the grammar easier to explain and understand, as there are no state labels taking the generation of a design from step to step.

#### **Step 1.1: Start**

Rules 1 through 4 apply at this stage. Rule 1 introduces the slab, a 0.20 m thick box that corresponds to the standard difference in level between the ground floor and the street. The adjustments to variations in topography is obtained by stacking steps on the street, at the entrance of the lot, which is a common procedure in the local vernacular architecture. Rule 2 creates the walls that enclose the floor. Rules 3 and 4 add to increase their thickness to 0.20 m when the wall borders the street.

#### **Step 1.2: Locate functional zones**

Rules 5 and 6 decides whether the outside zone (identified by label o) is going to be located at the front or at the back of the lot, thereby deciding to which of the two major

families of housetypes the house belongs. Label Q1, inherited from Rule 1, determines the location of the dissecting wall, load-bearing wall. There are two possible location: 6.0 m (Rule 5) and 7.0 m (Rule 6) away from the front border of the lot. Label Q2 marks where the dissection was done so that this information can be used in the derivation of the second floor. Rule 7 then applies to link an eventual backyard to the street by creating a corridor (co), which is a stand-alone space that does not belong to any of the zones. The width of this corridor is 1.10 m, which is the minimum allowed by regulations.

Rules 8 and 9 apply to locate the living zone (li) by dissecting the inside zone into living zone and sleeping (sl) or into living zone and service zone (se). Rules 10 and 11 locate the remaining zone by dissecting the outside zone, thereby determining the definite location of the yard (y). Rule 12 determines that there will be no further dissection of the outside zone, which then becomes the yard, by changing the label o into y. This rule is optional and can only be applied if the number of bedrooms of the house to generate is two ( $t_n = t_2$ ). The location of the dissecting wall is determined by structural and functional constraints. Structural restrictions limit the maximum span to 6.0 m. Functional constraints require the zone to have certain dimensions (width, length, and area) so that it can be subdivided into correctly dimensioned component rooms. The criteria used to obtain values for such dimensions resulted from the dimensional analysis of zones in the houses included in the corpus. Label Q'2 marks where the zone was actually dissected so that this information can be used in the derivation of the second floor. The symbol Z represents the set of required zones. Each time a rule is applied, the allocated zones are subtracted to this list.

### **Step 1.3: Define circulation scheme**

After allocating the functional zones within the lot, the next step defines the basic circulation scheme of the dwelling. Two sets of rules apply at this stage. The first set includes Rule 13 and Rule 14, which locate the main entrance to the house from the yard, near the wall that separates the living from the adjacent zone. They differ on the placement of label e, which establishes the circulation axis; in Rule 13, this axis is perpendicular to the yard, whereas in Rule 14 it is parallel. The other set of rules includes rules 15 through 18. Each of these rules places the staircase in a way that it overlaps label e. Such a restriction prevents the staircase from being located at the corner opposite to the entrance, which would not be a very good location, circulation wise. Rules 15 and 16 place an L- or an I-shaped staircase in the living zone, whereas Rules 17 and 18 place a U-shaped staircase in the service or in the sleeping zones. It is debatable whether these staircases could be arbitrarily placed in any of the zones. The linear and L-shaped staircases, however, seem to have a formal, almost decorative function that are appropriate for the living but not for the other zones. Such an interpretation is supported by the fact that they are open staircases. The U-shaped one, on the other hand, is a fully enclosed staircase that does not jeopardize either the privacy or the functionality required for the sleeping and service zones, respectively.

All of the staircases have always fourteen steps; the treads are 0.25 m deep, whereas the risers height depends on the floor height. Twelve of these steps constitute the body of the stairs, which is bounded by runaways on each side that form the remaining two-steps. If the living zone is not large enough, the linear staircase takes the form of an L-shape (Rule 15), or invades to the neighboring zone (Rule 16). The minimum length of the long tail of the L-shaped staircase is restricted to ten steps in order to guarantee that



a person does not hit the ceiling when climbing the stairs. This restriction is imposed by structural constraints that require the stairwell to be a rectangle so that the ceiling's pre-stressed concrete beams can run parallel to the stairs without interruption. All the rules for placing the staircase adjust the dimension of the zone in which the staircase is placed in order to comply with the rule for stair design just described.

#### **Step 1.4: Divide zones into rooms**

Computation at this stage is either based on the recursive dissection of zones to create rooms or on the connection of rooms that are functionally related to form larger rooms across the previously defined zones. The full set of rooms that can be included in the program of a Malagueira house are those already shown in Table 4.2. Some of these rooms appear in all the houses in the corpus, whereas others only appear in some. The break down of rooms into obligatory and optional sets that results from such a distinction is shown in Table 5.3 for each functional zone. It is acceptable to consider that the composition of these sets for a particular house would depend on the desired program. These sets are then used to control the derivation in order to guarantee that a house that fits a given program will be generated.  $R'$  is the set of obligatory rooms and  $R''$  is the set of optional rooms. The computation starts with  $R'$ , the set of obligatory rooms, including all the desired rooms. Each time a dissecting rule is applied, it creates a new room out a functional zone and it removes this room from the set of obligatory rooms for the zone. The computation stops when this set becomes empty or can continue until the set of optional rooms becomes empty too. The exact point of each dissection is informed by the values that width, length, and area can take for the room being allocated. The upper and lower limits of the range of values that these parameters can take were obtained after an analysis of the corpus' houses and respect the values prescribed by regulations.

There are five groups of rules that can be applied at this stage: dividing, extending, assigning, connecting and permuting rules. Dividing rules (19-32) divide a functional zone into rooms. Of these rules, only the rules to divide the living and the patio zones to create a circulation area consist of dissections perpendicular to the smaller edges of the rectangular rooms. All the remaining rules make dissections perpendicular to the larger edges. Extending rules (33-47) divide a zone to extend an adjacent zone. Some of these rules perform regular, perpendicular dissections whereas others (Rules 40-47) do 30° and 60° dissections. Rules 40 and 41 only apply to the living room to extend the service zone. Rules 42 through 47 perform perpendicular dissections on triangular and trapezoidal rooms created by non-perpendicular dissections. Assigning rules (48-57) are used to create the last required room, or an optional room, out of the space that remains after recursive application of dissecting rules. The rule that makes the yard out of the yard zone has the additional feature of lowering the level of the ground. Application of these rules prevents further application of the dissecting rules. Rule 55 is a dissecting rule that turns the space beneath the staircase into a closet. Rule 59 connects any two rooms of any shape that share, at least, a 1.20 m wall provided that they have the same function or different but related functions. Rule 60 adds an external cubic laundry to a concave corner of the yard that is adjacent to the service zone. Rule 61 is a special rule that applies to the design if the generated house is a two-bedroom house. This rule turns the kitchen into a bedroom and the transition space into the kitchen, thereby avoiding the need of a second floor.

Table 5.3 - Obligatory and optional sets of rooms by zone

Set of possible rooms:	$R = R' \cup R''$
Set of required rooms:	$R' = R'_{ly} \cup R'_{li} \cup R'_{se} \cup R'_{sl}$
Set of optional rooms:	$R'' = R''_{li} \cup R''_{se} \cup R''_{sl}$
Set of rooms in the yard zone:	$R_y = R'_y = \{y\} \cup R''_y = \emptyset$
Set of rooms in the living zone:	$R_{li} = R'_{li} = \{li\} \cup R''_{li} = \{cl\}$
Set of rooms in the service zone:	$R_{se} = R'_{se} = \{ki\} \cup R''_{se} = \{ts, la, pa\}$
Set of rooms in the sleeping zone (first floor) :	$t_n = t_2 \Rightarrow R_{sl1} = R'_{sl1} = \{be_1, be_2, ba\} \cup R''_{sl2} = \{cl\} \vee R'_{sl1} = \emptyset$ $t_n \neq t_2 \Rightarrow R = R'_{sl1} = \{be_1, ba\} \cup R''_{sl2} = \{cl\}$
Set of rooms in the sleeping zone: (second floor)	$t_n = t_1 \Rightarrow R_{sl1} = \emptyset$ $t_n = t_2 \Rightarrow R_{sl1} = \emptyset \vee R = R'_{sl1} \{be_1, be_2, ba\} \cup R''_{sl2} = \{cl\}$ $t_n = t_3 \Rightarrow R_{sl1} = R'_{sl1} = \{be_2, be_3, ba\} \cup R''_{sl2} = \{cl\}$ $t_n = t_4 \Rightarrow R_{sl1} = R'_{sl1} = \{be_2, be_3, be_4, ba\} \cup R''_{sl2} = \{cl\}$ $t_n = t_5 \Rightarrow R_{sl1} = R'_{sl1} = \{be_2, be_3, be_4, be_5, ba\} \cup R''_{sl2} = \{cl\}$

### Step 1.5: Introduce details

This step takes care of detailing the design of the floor plan. Four sets of rules apply at this stage. The first set (Rules 62-67) creates chimneys. All of these rules can be applied to the kitchen whereas only rules 64 through 67 can be used for the living-room. The next set of rules (Rules 68-73) make adjustments on the thickness of the walls depending on their location. Rules 68 through 72 increase the thickness of exterior walls towards the outside from 0.075 m to 0.20 m. Rule 73 decreases the thickness of interior walls from 0.20 m down to 0.075 m, if the span is smaller than 2.00m. Rule 68 is a special rule that increases the wall between living room and the patio to accommodate window shutters when these are opened. The third set of rules (Rules 74-76) includes rules to complete the design of the stairs. Finally, Rules 77 and 78 decrease the height

of the patio's walls down to 1.50 m. Rule 78 decreases the height of these walls evenly when the patio borders streets on both sides. Rule 77 decreases the height of the wall between the patio and the street without decreasing the height of the wall between the patio and the neighboring lot.

### **Step 1.6: Introduce openings**

The rules that apply at this step of the derivation pierce openings on the walls and introduce mullions in the openings. There are rules for piercing the exterior openings (Rules 79-93) and rules for piercing the interior openings (Rules 94-99.) Rules 79 and 80 are the most important rule for placing exterior openings as it encodes the basic strategy used by Siza for the design of the front elevation. In the front yard houses, the strategy is as follows: each floor has two windows placed symmetrically in relation to each other, and the windows on the second floor are aligned with the windows on the first floor. Such a strategy holds even when the front wall on the second floor is on a different plane as the one on the first floor, which suggests that Siza thought of the design the facades on the two dimensional plane of the drawing board. A logical procedure as the houses are perfectly aligned facing each other on the street and so the view of one house from the other is very close to its two dimensional representation. Rules 79 and 80 thus places two labeled axes  $e_1$  on the first floor and two labeled axes  $e_2$  on to the second floor which are the labels that permit the application of Rules 81 and 82. The remaining rules specify how openings can be pierced in the front elevation in contexts other than the one specified by the previous rules or on other elevations. The placement of openings on the front elevation of backyard houses does not follow the same strategy used in front yard houses as the lateral corridor accessing the yard makes it impossible to design a symmetrical façade. The strategy in this case is to use

the remaining rules and accept asymmetry. In these rules, if the opening is pierced on the front wall, it is shown both on the plan, and on the elevation, whereas if it placed on another wall, it is only shown on the plan. Rules 82 through 85 erase openings on the front elevation that are not placed on the front facade, if they are hidden by the patio wall.

After piercing the openings, the derivation of the elevation proceeds with the definition of the mullions' geometry (*Rules 100-107*). Interestingly enough, the design of this geometry follows the same compositional rule of perpendicular rectangular dissections used in the design of the layout.

### **Step 1.7: Terminate**

The last step in the derivation of the first floor includes rules that delete unnecessary labels (Rule 108) and change the state label from  $F_1$  to  $F_2$  (Rule 104). The derivation then proceeds on to the second floor.

### **5.9.3 Stage 2: Define the second floor**

The second floor's derivation goes, to a certain extent, through the same steps of the first floor's derivation. The rules are also very similar and so they will not be described in detail. Instead, the differences between the derivations of both floors will be highlighted. When the actual derivation of the second floor starts, it has already inherited a series of labels from the derivation of the first floor. Such labels carry information that will be used to make new dissections, extend the staircase and the chimneys, and to place the openings.

The first step of the derivation introduces the slab (Rule 110) and the enclosing walls (Rule 2), and adjust the wall thickness (Rules 111-113.)

The next step replicates the dissections of the first floor, using the existing labels as markers. If the first floor's first dissection was done 6.0 m away from the front of the lot, the corresponding dissection of the second floor can be done at the same place (Rule 114) or 1.0 m meter backwards (Rule 115), so that a verandah zone will be created as a result. The next dissections replicate exactly the first floor's division into functional zones (Rule 116). The following dissections, however, might (Rule 117), or might not (Rule 118) replicate the first floor's dissections, depending whether the number of required bedrooms is equal or bigger than two.

The next step of the derivation extends the staircase (Rule 119-120) and defines the basic circulation scheme by creating a corridor perpendicular (Rules 121 and 124) or parallel (Rules 122 and 123) to the staircase. The choice between these two options depends whether the lot borders a street or a house on the side where the corridor will be placed, as it constrains the location of windows in the room that is adjacent to the staircase.

The next step divides the remaining space into rooms or assigns a bedroom or a terrace to a room that resulted from the replication of first floor's dissections (Rules 125-136). There are some constraints to such operations: the rooms above the inside zone defined by the first dissection cannot become terraces and the remaining rooms can only become terraces if the layout has the required number of bedrooms. Other rules that

can be applied at this step to create corridors, bathrooms, or extend an existing bedroom by dissecting another room or the end of a corridor.

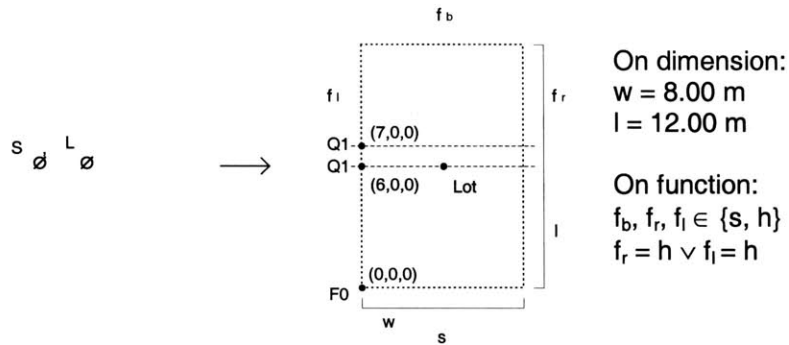
Finally, the last three steps of the derivation introduce the details (Rules 137-146), introduce the openings, and erase unnecessary labels (Rules 147). The last rule (Rule 148) changes the state label.

#### **5.9.4 Stage 3: Define the terrace**

The stage has fewer steps than the previous two. The first introduces the slab and encloses the terrace (Rules 149 and 2). The second replicates the division of the lot into inside and outside zones (Rule 150) and then into functional zones (Rule 151). The third extends the chimneys (Rules 154-159) and erases the walls around the patio (Rule 160). The fourth and last step erases the unnecessary labels (Rules 161) and applies the termination rule (Rule 162) that ends the derivation.

**Stage 0: Introduce initial shape and grammar labels**

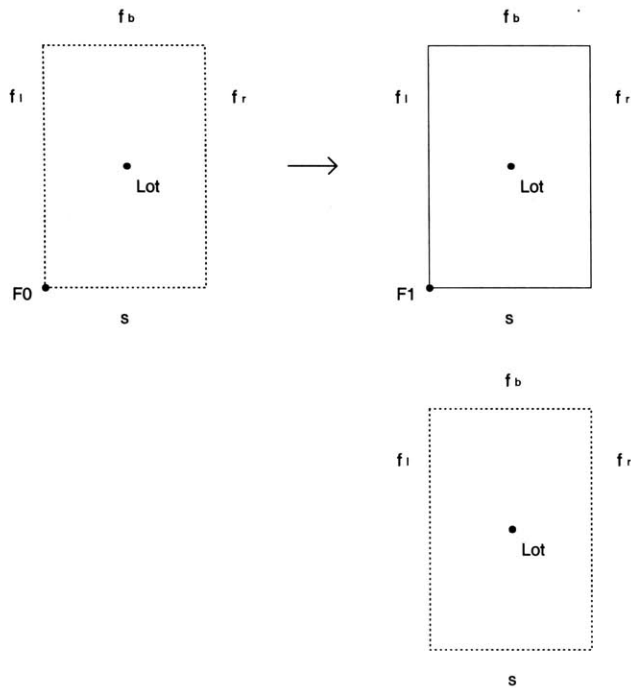
Rule 0: Introduce initial shape



**Stage 1: Define first floor**

Step 1.1: Start

Rule 1: Introduce slab

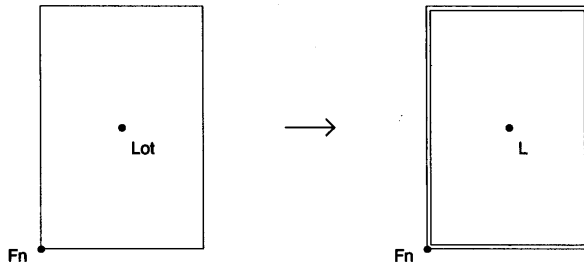


$F1 \text{ — } h_{s1} \quad h_{s1} = 0.20 \text{ m (Table 4.4)}$

Figure 5.19 – Shape grammar rules



**Rule 2: Introduce enclosing walls**

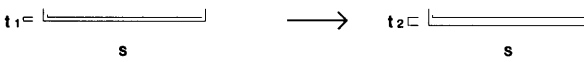


On dimension:  
 $h_{\min} < h_{f1} < h_{\max}$  (Table 4.4)  
 $t = 0.1 \text{ m}$

On function:  
 $n \in \{1, 2\}$



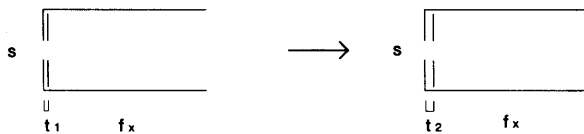
**Rule 3: Adjust front and back wall thickness**



On dimension:  
 $t_1 = 0.10 \text{ m} \Rightarrow t_2 = 0.20 \text{ m}$

On function  
 $p(s) \in \{f_r, f_b\}$

**Rule 4: Adjust left and right wall thickness**



On dimension:  
 $t_1 = 0.10 \text{ m} \Rightarrow t_2 = 0.20 \text{ m}$

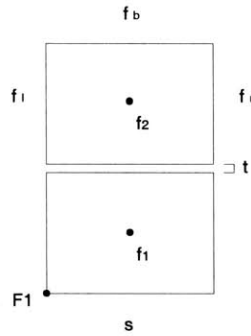
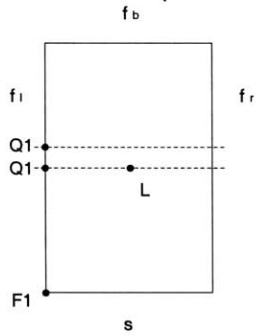
On function:  
 $p(f_x) \in \{f_l, f_r\}$



Figure 5.19 (continued) – Shape grammar rules

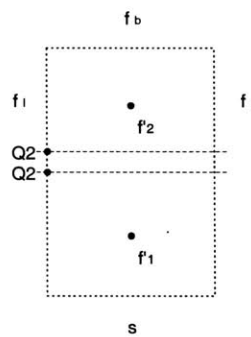
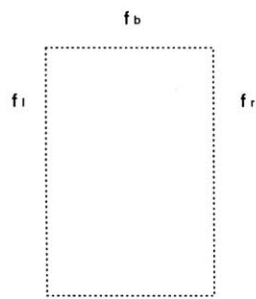
Step 1.2: Locate functional zones

Rule 5: Locate patio



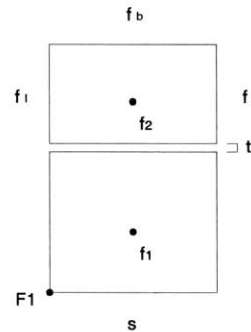
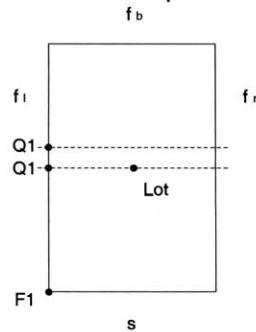
On dimension:  
 $t = 0.20 \text{ m}$

On function:  
 $f_1, f_2 \in \{i, o\} \wedge f_1 \neq f_2$   
 $f_1 = f'_1, f_2 = f'_2$



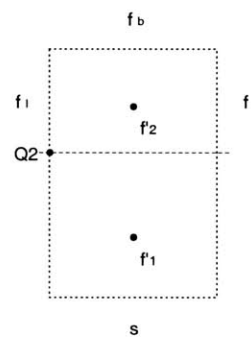
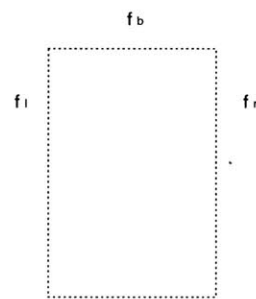
R4:  $\langle f_1, f_b, f_r, s, f_l, L: f_1, f_2 \rangle$

Rule 6: Locate patio



On dimension:  
 $t = 0.20 \text{ m}$

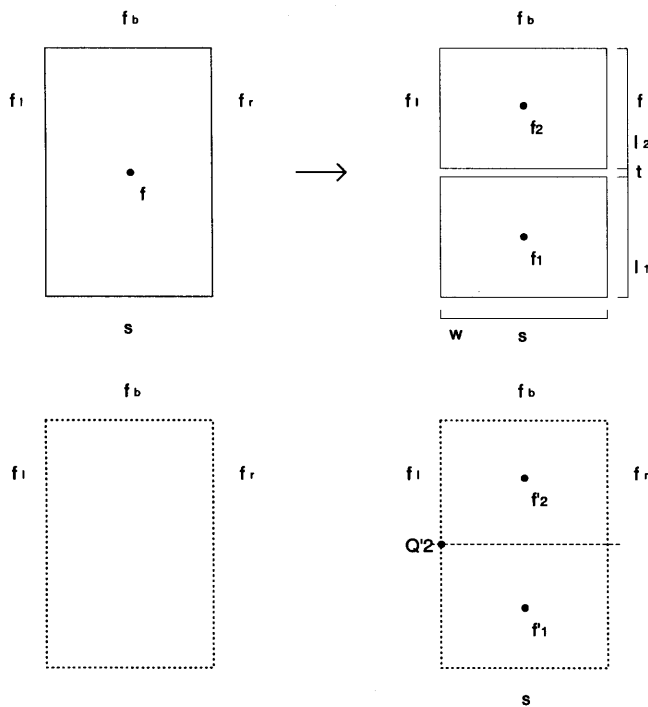
On function:  
 $f_1, f_2 \in \{i, o\} \wedge f_1 \neq f_2$   
 $f_1 = f'_1, f_2 = f'_2$



R4:  $\langle f_1, f_b, f_r, s, f_l, L: f_1, f_2 \rangle$

Figure 5.19 (continued) – Shape grammar rules

Rules 7-11: Locate functional zones



On dimension:

$$l_m < l_1$$

$$l_m < l_2$$

$$w_m > w > w_x$$

$$t = 0.20 \text{ m}$$

On function:

$$f_1 = 0 \Rightarrow f'_1 = 0$$

$$f_2 = 0 \Rightarrow f'_2 = 0$$

$$f = 0 \wedge f_1 \neq 0 \Rightarrow f'_1 = i'$$

$$f = 0 \wedge f_2 \neq 0 \Rightarrow f'_2 = i'$$

$$f = i \Rightarrow f'_1 = i \wedge f'_2 = i$$

Rule 7: Locate backyard corridor

$$R7: \langle F_1, f_b, o, f_t, s, i; i, co \rangle$$

$$f_t, f_b \neq s \wedge co \in Z$$

Rule 8: Locate living and service zones

$$R8: \langle F_1, f_b, f_r, f_t, f_l, i; li, se; Z-\{li, se\} \rangle$$

$$li, se \in Z$$

Rule 9: Locate living and sleeping zones

$$R9: \langle F_1, f_b, f_r, f_t, f_l, i; li, sl; Z-\{li, sl\} \rangle$$

$$x_t \neq co; li, sl \in Z$$

Rule 10: Locate patio and service zones

$$R10: \langle F_1, f_b, f_r, f_t, li, o; y, se; Z-\{ya, se\} \rangle$$

$$se \in Z$$

Rule 11: Locate patio and sleeping zones

$$R11: \langle F_1, f_b, f_r, f_t, li, o; y, sl; Z-\{ya, sl\} \rangle$$

$$sl \in Z$$

Rule 12: No dissection: outside zone becomes patio zone

$$o \rightarrow y$$

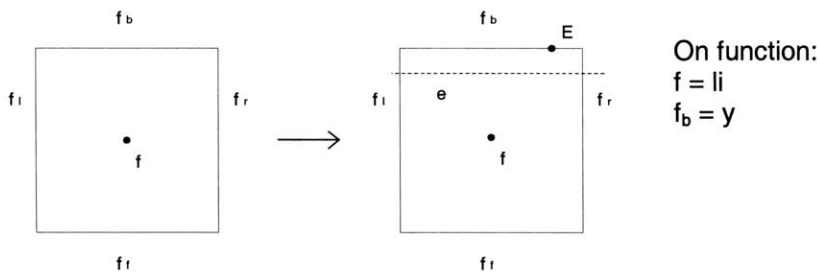
$$R12: \langle F_1, f_b, f_r, f_l, f_t, o; y \rangle,$$

$$t_n = t_2$$

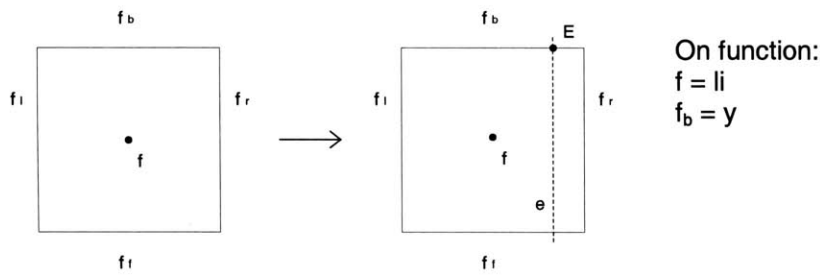
Figure 5.19 (continued) – Shape grammar rules

Step 1.3: Define circulation

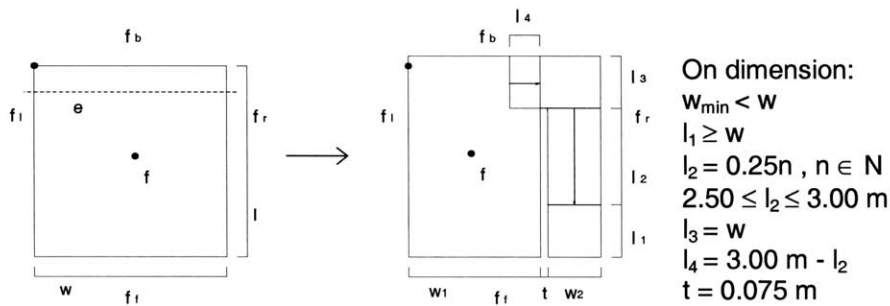
Rule 13: Locate main entrance



Rule 14: Locate main entrance



Rule 15: Locate staircase in the living zone



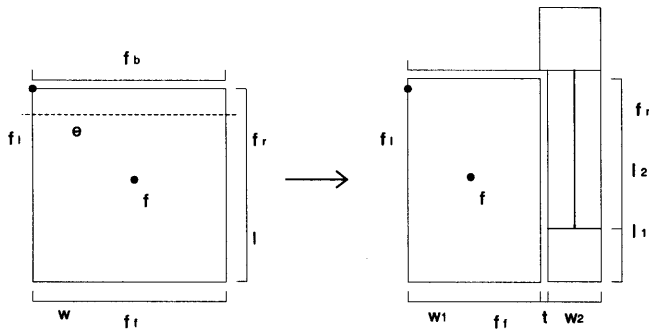
On function:  
 $f_r \neq 0$



R15:  $\langle F_1, f_b, x_r, x_f, x_l, l_i, l_i, st \rangle$

Figure 5.19 (continued) – Shape grammar rules

**Rule 16: Locate staircase in the living zone**



**On dimension:**

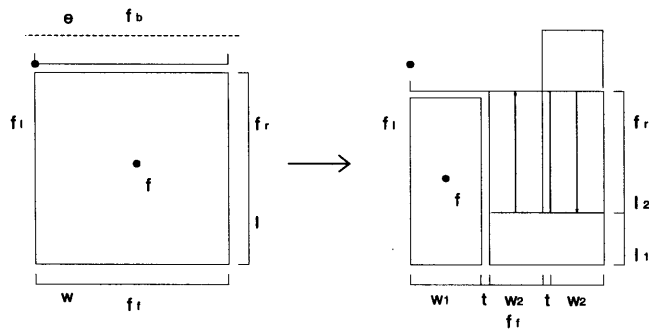
- $w_{min} < w$
- $l_1 \geq w$
- $l_2 = 3.00 \text{ m}$
- $l_3 = w$
- $t = 0.075 \text{ m}$

**On function:**

$f_r \neq 0$

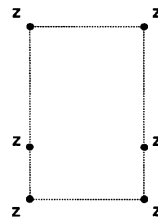


**R16: <F1, f\_b, f\_r, f\_t, f\_e, li; li, st>**



**On dimension:**

- $w_{min} < w$
- $l_1 \geq w$
- $l_2 = 1.50 \text{ m}$
- $l_3 \geq w$
- $t = 0.20 \text{ m}$



**Rule 17: Locate staircase in the sleeping zone**

**R17: <F1, f\_b, f\_r, f\_t, f\_l, li; li, st >**

$f_b = li$

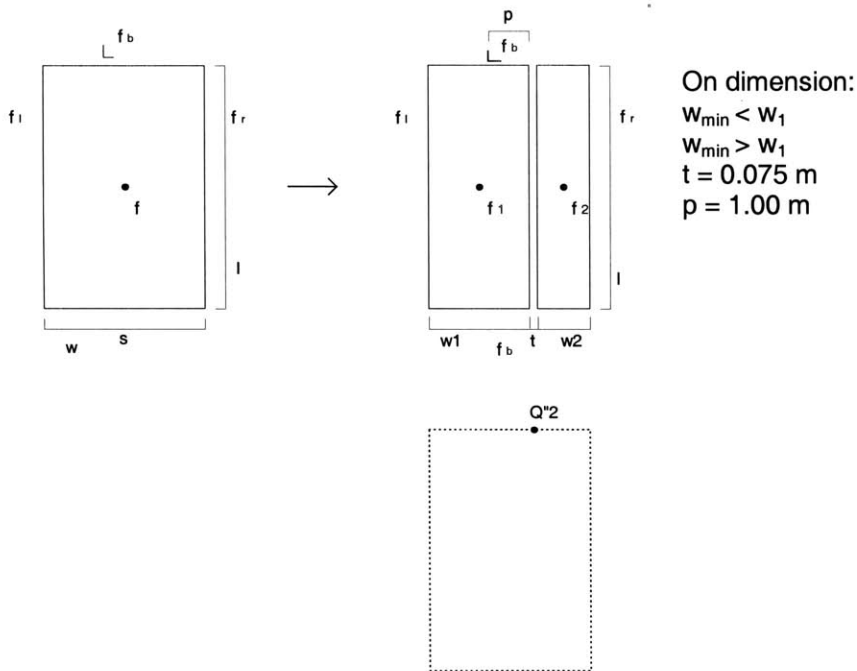
**Rule 18: Locate staircase in the service zone**

**R18: <F1, f\_b, f\_r, f\_t, f\_b, sl; sl, st >**

$f_b = li$

Figure 5.19 (continued) – Shape grammar rules

Step 1.4: Dividing zones into rooms



- Rule 19: Dissecting patio zone into patio and circulation  
 R19:  $\langle F1, f_b, l_i, f_r, f_l, y; y, c_i \rangle$   
 Rule 20: Dissecting living zone into living and circulation  
 R20:  $\langle F1, f_b, f_r, f_l, f_i, l_i, c_i \rangle$

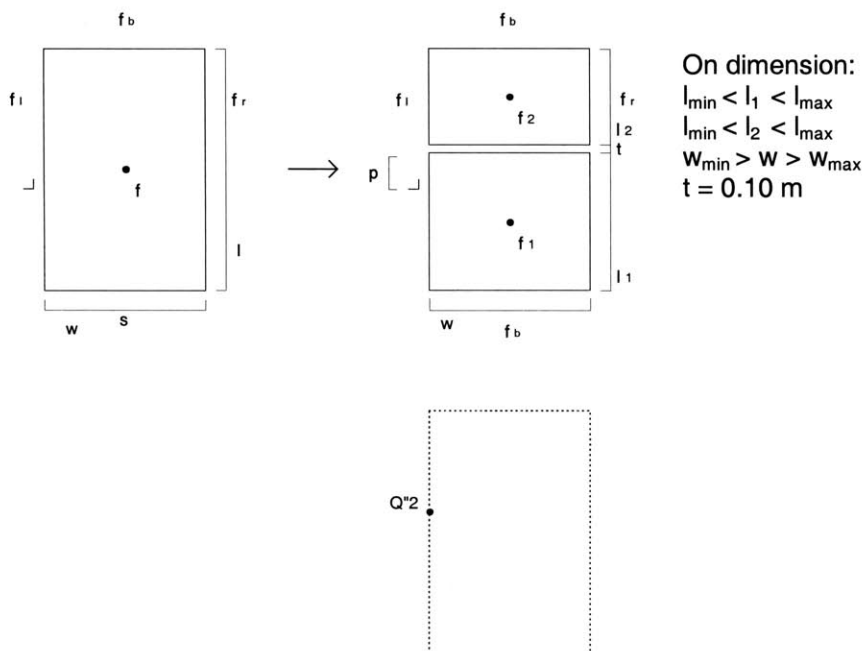


Figure 5.19 (continued) – Shape grammar rules

Rules 21-25: Dividing service zone

R21: <F1, f<sub>b</sub>, f<sub>r</sub>, f<sub>f</sub>, f<sub>i</sub>, se; se, ki>

R22: <F1, f<sub>b</sub>, f<sub>r</sub>, f<sub>f</sub>, f<sub>i</sub>, se; se, ts>

R23: <F1, f<sub>b</sub>, f<sub>r</sub>, f<sub>f</sub>, f<sub>i</sub>, se; se, pa>

R24: <F1, f<sub>b</sub>, f<sub>r</sub>, f<sub>f</sub>, f<sub>i</sub>, se; se, la>

R25: <F1, f<sub>b</sub>, f<sub>r</sub>, f<sub>f</sub>, f<sub>i</sub>, se; se, ci>

$$\exists f_n : f_n = li, f_n \in \{f_r, f_i\}$$

Rules 26-28: Dividing sleeping zone

R26: <F1, f<sub>b</sub>, f<sub>r</sub>, f<sub>f</sub>, f<sub>i</sub>, sl; sl, be>

R27: <F1, f<sub>b</sub>, f<sub>r</sub>, f<sub>f</sub>, f<sub>i</sub>, sl; sl, ba>

R28: <F1, f<sub>b</sub>, f<sub>r</sub>, f<sub>f</sub>, f<sub>i</sub>, sl; sl, ci>

$$\exists f_n : f_n = ci, f_n \in \{f_r, f_i, f_j\}$$

$$\forall f_n : f_n \neq ci$$

Rules 29-32: Dividing circulation

R29: <F1, li, be, f<sub>r</sub>, f<sub>i</sub>, st; st, ci>

R30: <F1, f<sub>b</sub>, f<sub>r</sub>, f<sub>f</sub>, f<sub>i</sub>, ci; ci, be>

R31: <F1, f<sub>b</sub>, f<sub>r</sub>, f<sub>f</sub>, f<sub>i</sub>, ci; ci, ba>

R32: <F1, f<sub>b</sub>, f<sub>r</sub>, f<sub>f</sub>, f<sub>i</sub>, ci; ci, y>

$$p_2 = 0$$

$$f_b \in \{h, s\} \wedge \exists f_n : f_n = be, f_n \in \{f_r, f_i, f_j\}$$

$$P_1 \geq 1.0 \text{ m}$$

$$f_n \in \{h, s\} \wedge f_r, f_i \neq st$$

$$P_1 \geq 1.0 \text{ m}$$

$$f_n \in \{h, s\} \wedge f_r = y$$

Rules 33-38: Dissecting nearby zones

R33: <F1, f<sub>b</sub>, f<sub>r</sub>, f<sub>f</sub>, f<sub>i</sub>, se; se, y>

R34: <F1, f<sub>b</sub>, f<sub>r</sub>, f<sub>f</sub>, f<sub>i</sub>, sl; sl, y>

R35: <F1, f<sub>b</sub>, f<sub>r</sub>, f<sub>f</sub>, f<sub>i</sub>, li; li, y>

R36: <F1, f<sub>b</sub>, f<sub>r</sub>, f<sub>f</sub>, f<sub>i</sub>, se; se, li>

R38: <F1, f<sub>b</sub>, f<sub>r</sub>, f<sub>f</sub>, f<sub>i</sub>, y; y, se>

R39: <F1, f<sub>b</sub>, f<sub>r</sub>, f<sub>f</sub>, f<sub>i</sub>, li; li, se>

$$p_1 \geq 1.0 \text{ m}$$

$$\exists f_n : f_n = y, f_n \in \{f_b, f_r, f_i\}$$

$$p_1 \geq 1.0 \text{ m}$$

$$\exists f_n : f_n = y, f_n \in \{f_b, f_r, f_i\}$$

$$p_1 \geq 1.0 \text{ m}$$

$$\exists f_n : f_n = y, f_n \in \{f_b, f_r, f_i\}$$

$$\exists f_n : f_n = li, f_n \in \{f_b, f_r, f_i\}, li \in \{li, lr\}$$

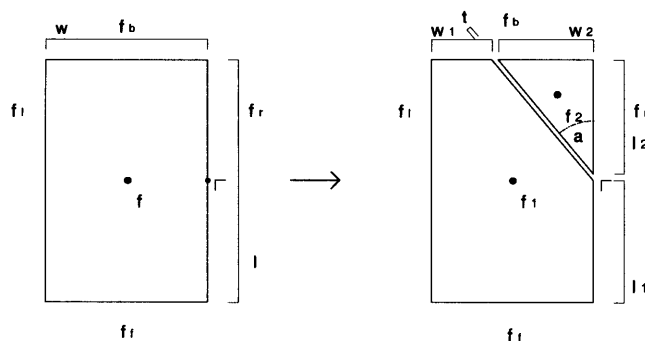
$$\exists f_n : f_n = se,$$

$$f_n \in \{f_b, f_r, f_i\}, se \in \{se, ki, ts\},$$

$$\exists f_n : f_n = li,$$

$$f_n \in \{f_b, f_r, f_i\}, se \in \{se, ki, ts\}$$

Rules 40-47: Dissect living or service zone or rooms with a diagonal wall



On dimension:

$$\angle a \in \{30^\circ, 60^\circ\}$$

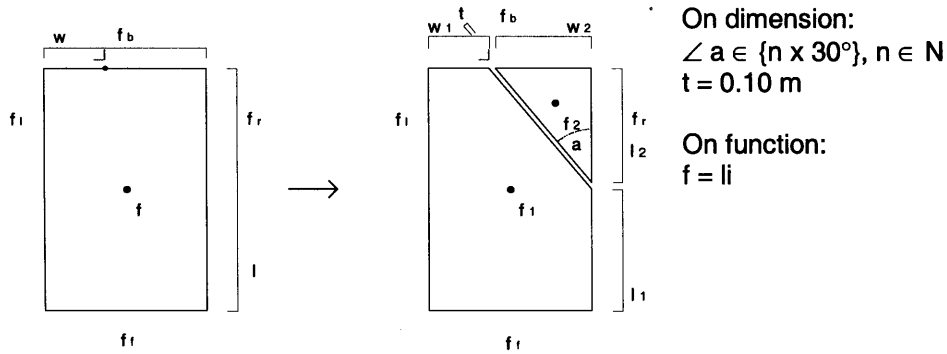
$$t = 0.10 \text{ m}$$

On function:

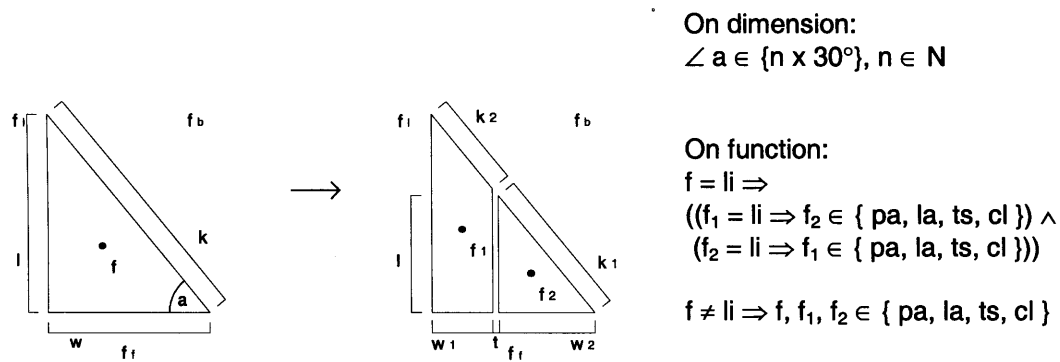
$$f = li$$

R40: <F1, f<sub>b</sub>, f<sub>r</sub>, f<sub>f</sub>, f<sub>i</sub>, li; li, pa>

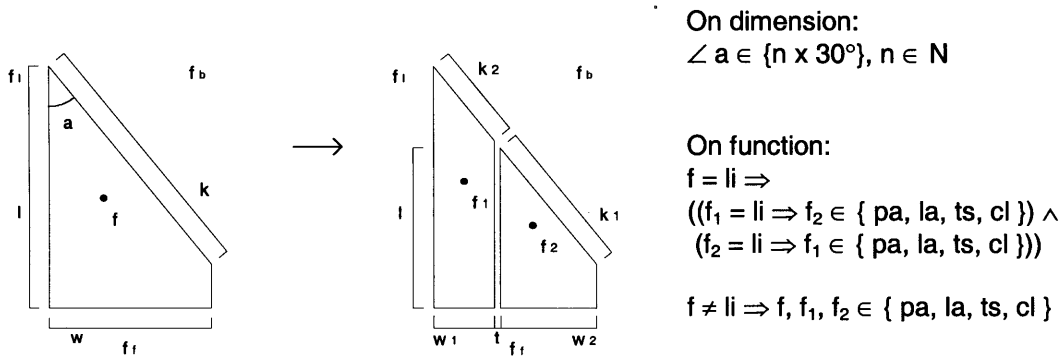
Figure 5.19 (continued) – Shape grammar rules



R41:  $\langle F1, f_b, f_r, f_l, f_i, li; li, ts \rangle$



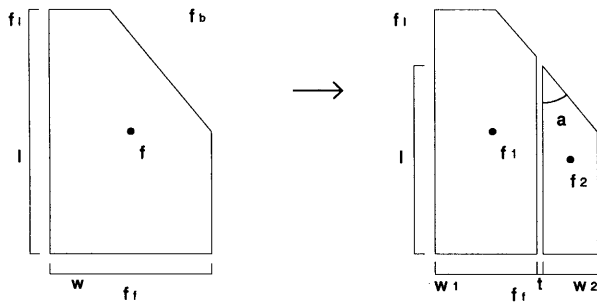
R42:  $\langle F1, f; f_1, f_2 \rangle$



R43:  $\langle F1, f; f_1, f_2 \rangle$

Figure 5.19 (continued) – Shape grammar rules



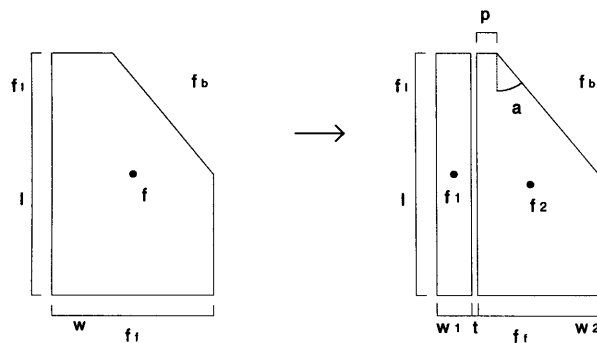


On dimension:  
 $\angle a \in \{n \times 30^\circ\}, n \in \mathbb{N}$

On function:  
 $f = li \Rightarrow$   
 $((f_1 = li \Rightarrow f_2 \in \{pa, la, ts, cl\}) \wedge$   
 $(f_2 = li \Rightarrow f_1 \in \{pa, la, ts, cl\}))$

$f \neq li \Rightarrow f, f_1, f_2 \in \{pa, la, ts, cl\}$

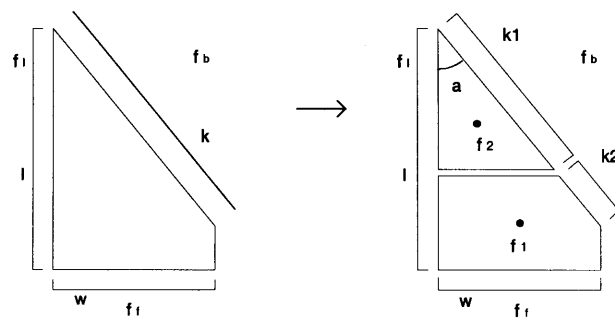
R44:  $\langle F1, f; f_1, f_2 \rangle$



On dimension:  
 $\angle a \in \{n \times 30^\circ\}, n \in \mathbb{N}$

On function:  
 $f = li \wedge p = 0 \Rightarrow$   
 $(f_1 = li \wedge f_2 \in \{pa, la, ts, cl\})$   
 $f = li \wedge p \neq 0 \Rightarrow$   
 $f_1, f_2 \in \{pa, la, ts, cl\}$   
 $f \neq li \Rightarrow f, f_1, f_2 \in \{pa, la, ts, cl\}$

R45:  $\langle F1, f; f_1, f_2 \rangle$

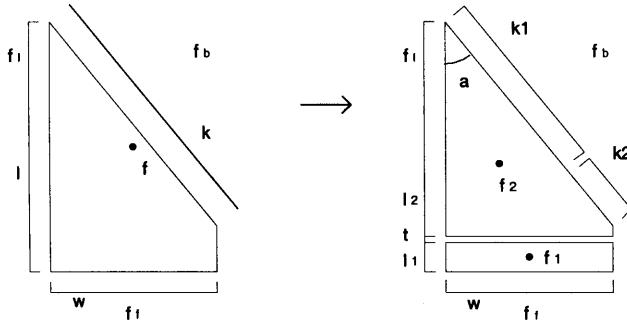


On dimension:  
 $\angle a \in \{n \times 30^\circ\}, n \in \mathbb{N}$

On function:  
 $f = li \Rightarrow (f_1 = li \wedge \{pa, la, ts, cl\})$   
 $f \neq li \Rightarrow f, f_1, f_2 \in \{pa, la, ts, cl\}$

R46:  $\langle F1, f; f_1, f_2 \rangle$

Figure 5.19 (continued) – Shape grammar rules

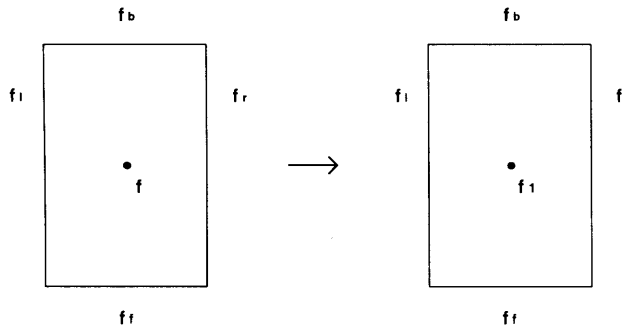


On dimension:  
 $\angle a \in \{n \times 30^\circ\}, n \in \mathbb{N}$

On function:  
 $f = li \wedge p = 0 \Rightarrow$   
 $(f_1 = li \Rightarrow f_2 \in \{pa, la, ts, cl\})$   
 $f \neq li \Rightarrow f, f_1, f_2 \in \{pa, la, ts, cl\}$

R47:  $\langle F1, f, f_1, f_2 \rangle$

Rules 48-57: Terminate division of zones



Rule 48: Terminate division of the yard zone

R48:  $\langle F1, f_b, f_r, f_t, f_l, y; y_a \rangle$

$R'_{se} = \emptyset \wedge R'_{sl} = \emptyset$

Rule 49: Terminate division of the living zone

R49:  $\langle F1, f_b, f_r, f_t, f_l, li; lr \rangle$

Rules 50-53: Terminate division of the service zone

R50:  $\langle F1, f_b, f_r, f_t, f_l, se; ki \rangle$

$R'_{se} = \emptyset$

R51:  $\langle F1, f_b, f_r, f_t, f_l, se; ts \rangle$

$R'_{se} = \emptyset$

R52:  $\langle F1, f_b, f_r, f_t, f_l, se; pa \rangle$

$R'_{se} = \emptyset$

R53:  $\langle F1, f_b, f_r, f_t, f_l, se; la \rangle$

$f_b \in \{ki, li\} \Rightarrow f_r \notin \{ki, li\}$

$f_r \in \{ki, li\} \Rightarrow f_l \notin \{ki, li\}$

$R'_{se} = \emptyset$

R54:  $\langle F1, f_b, f_r, f_t, f_l, se; y \rangle$

$f_b \in \{ki, li\} \Rightarrow f_r \notin \{ki, li\}$

$f_r \in \{ki, li\} \Rightarrow f_l \notin \{ki, li\}$

$R'_{se} = \emptyset$

$\exists f_n: f_n = y, n \in \{b, r, f, l\}$

Rules 55-56: Terminate division of the sleeping zone

R55:  $\langle F1, f_b, f_r, f_t, f_l, sl; ba \rangle$

$R'_{sl} = \emptyset$

R56:  $\langle F1, f_b, f_r, f_t, f_l, sl; cl \rangle$

$R'_{sl} = \emptyset$

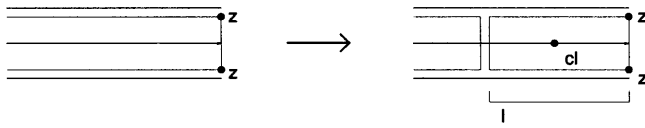
R57:  $\langle F1, f_b, f_r, f_t, f_l, sl; y \rangle$

$R'_{sl} = \emptyset$

$\exists f_n: f_n = y, n \in \{b, r, f, l\}$

Figure 5.19 (continued) – Shape grammar rules

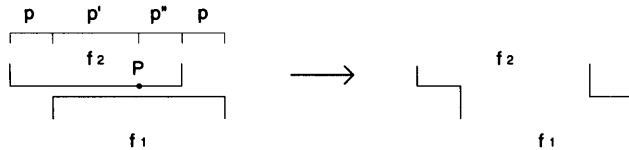
**Rule 58: Create closet under the staircase**



On dimension:  
 $l = 1.75 \text{ m}$

R58:  $\langle F_1, f_b, f_r, f_l, f_i, st; st, cl \rangle$

**Rule 59: Concatenate adjacent rooms**



On dimension:  
 $p \geq 0$   
 $p' \geq 1.20\text{m}$   
 $p'' \geq 0$

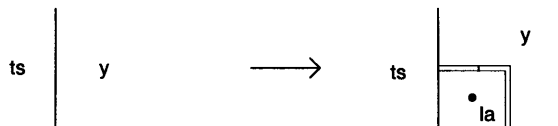
On function:  
 $X = R \vee X = Q$

$F = F_1$   
 $f_1 = f_2$   
 $f_1 \neq f_2 \Rightarrow f_1, f_2 \in R_i \vee \exists f_n : f_n = ci,$   
 $i \in \{y, li, se, sl\}, n \in \{1, 2\}$

$F = F_2$   
 $f_1 = f_2$   
 $f_1 \neq f_2 \Rightarrow \exists f_n : f_n = ci$

R59:  $\langle F_1, f_b, f_r, f_l, f_i, st; st, cl \rangle$

**Rule 60: Locate laundry**



**Rule 61: Turn kitchen into bedroom and transitional space into kitchen**

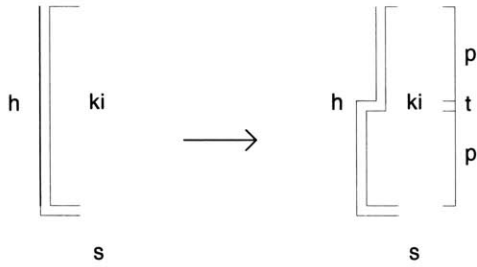
R61:  $\langle F_1, ki, ts; ts, ki \rangle$

$t_n = t_2$

Figure 5.19 (continued) – Shape grammar rules

**Step 1.5: Introduce details**

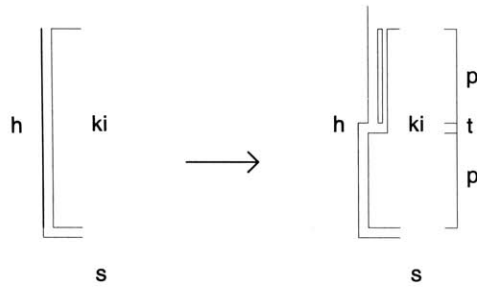
**Rules 62-67: Introduce chimney**



**R62**

On dimension:  
 $t = 0.10 \text{ m}$   
 $1.0 \leq p \leq 1.2 \text{ m}$

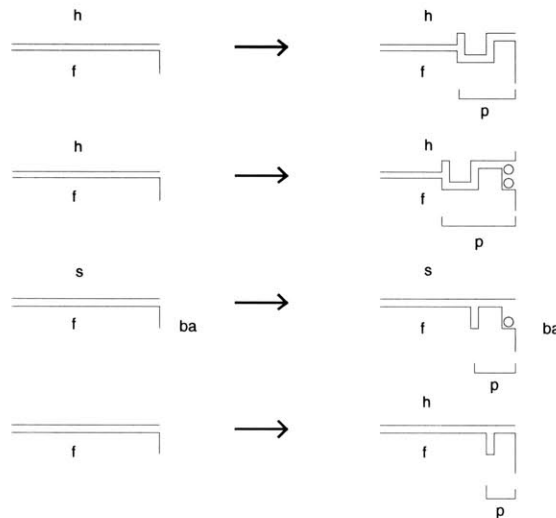
On function:  
 $p(h) \in \{f_l, f_r\}$



**R63**

On dimension:  
 $t = 0.10 \text{ m}$   
 $1.0 \leq p \leq 1.2 \text{ m}$

On function:  
 $p(h) \in \{f_l, f_r\}$



**R64**

On dimension:  $p = 1.00 \text{ m}$   
 On function:  $f = ki \vee f = li$

**R65**

On dimension:  $p = 1.20 \text{ m}$   
 On function:  $f = ki \vee f = li$

**R66**

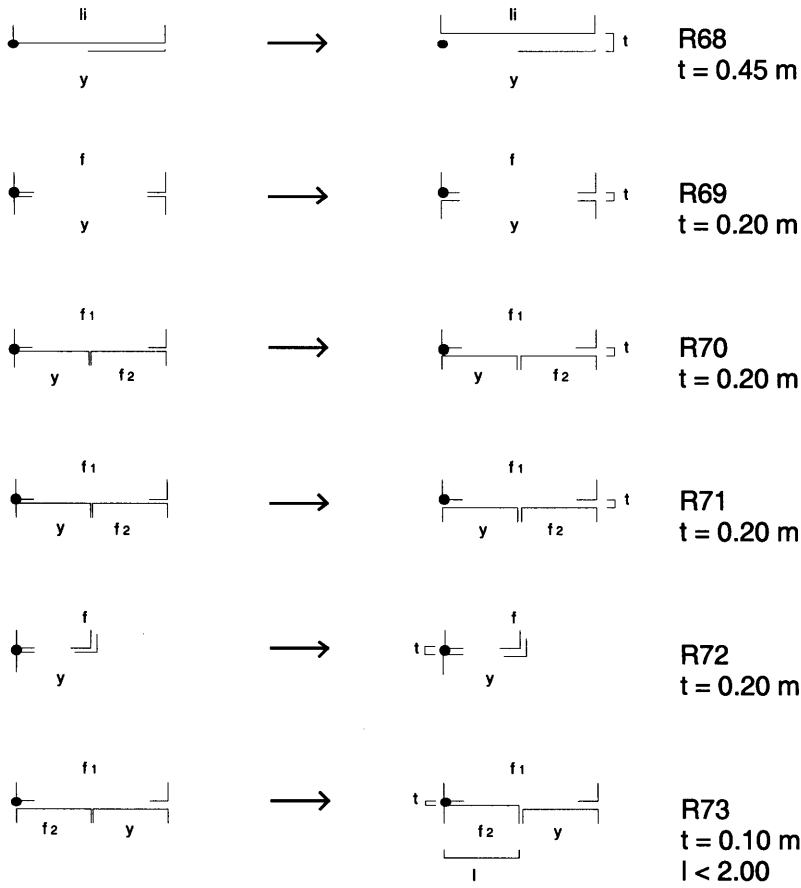
On dimension:  $p = 0.50 \text{ m}$   
 On function:  $f = ki \vee f = li$

**R67**

On dimension:  $p = 0.70 \text{ m}$   
 On function:  $f = ki \vee f = li$

Figure 5.19 (continued) – Shape grammar rules

**Rules 68-73: Adjust wall thickness**



**Rule 74-76: Detail stairs**

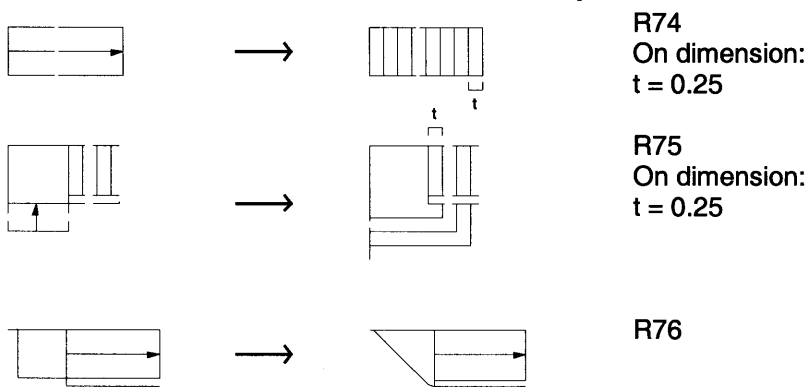
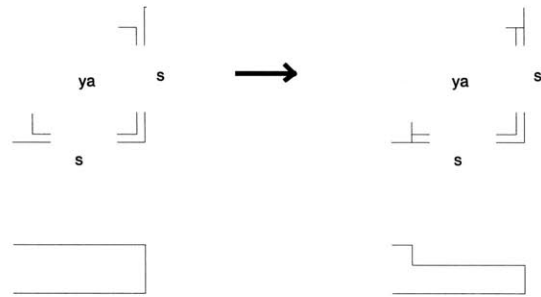
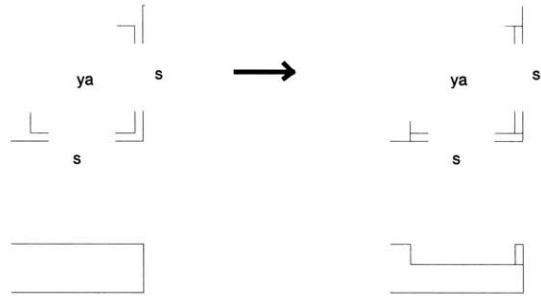


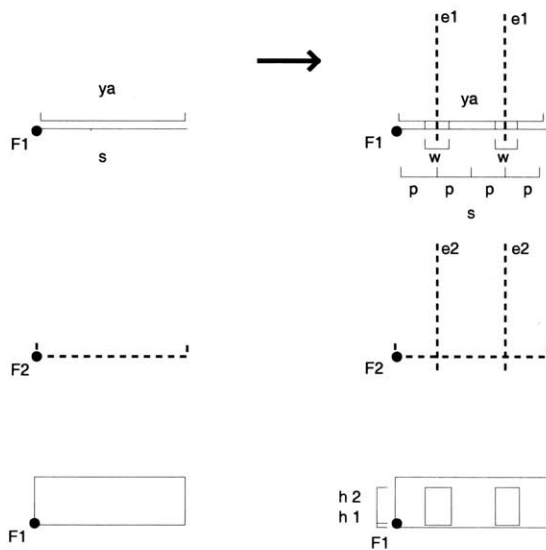
Figure 5.19 (continued) – Shape grammar rules

Rules 77-78: Adjust the patio wall height



Step 1.6: Introduce openings

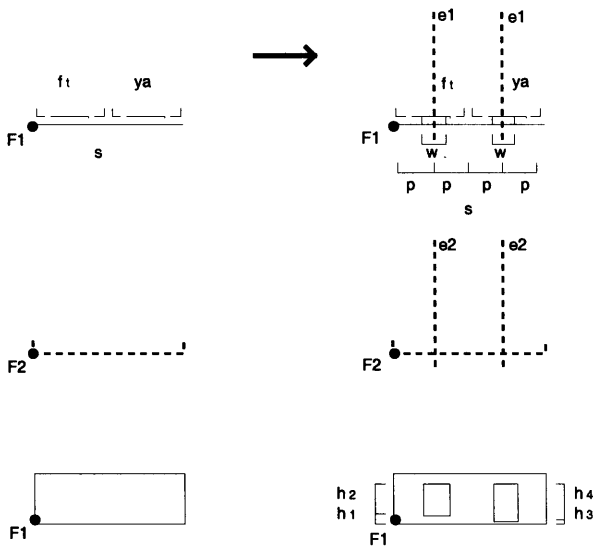
Rule 79: Pierce front facade openings



- On dimension:
- $w \in \{ 1.05, 1.10 \}$
- $h_1 \in \{ 0.00, 0.20 \}$
- $h_2 \in \{ 2.00, 2.07 \}$

Figure 5.19 (continued) – Shape grammar rules

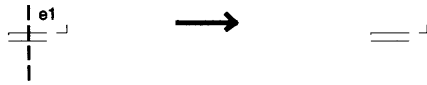
**Rule 80: Pierce front facade openings**



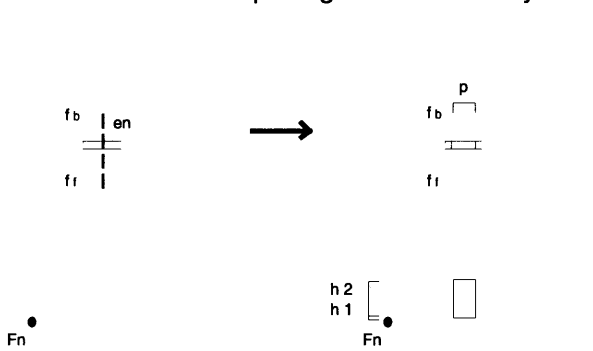
On dimension:

- $w \in \{ 1.05, 1.10 \}$
- $h_1 \in \{ 0.00, 0.20 \}$
- $h_2 \in \{ 2.00, 2.07 \}$
- $h_3 \in \{ 0.20, 0.35, 0.75 \}$
- $h_4 \in \{ 2.00, 2.07 \}$

**Rule 81: Erase axis of symmetry**



**Rule 82: Pierce exterior opening on the axis of symmetry**



On dimension:

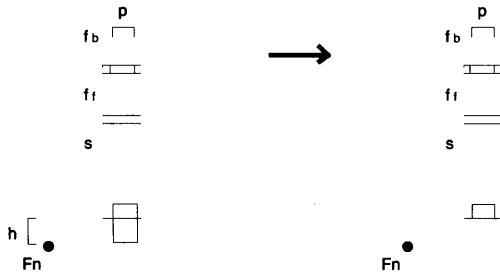
- $w \in \{ 1.05, 1.10 \}$
- $h_1 \in \{ 0.00, 0.20 \}$
- $h_2 \in \{ 2.00, 2.07 \}$
- $h_3 \in \{ 0.20, 0.35, 0.75 \}$
- $h_4 \in \{ 2.00, 2.07 \}$

On function

- $F_t = \{ ya, te, s \}$
- $F_n \in \{ 1, 2 \}$

Figure 5.19 (continued) – Shape grammar rules

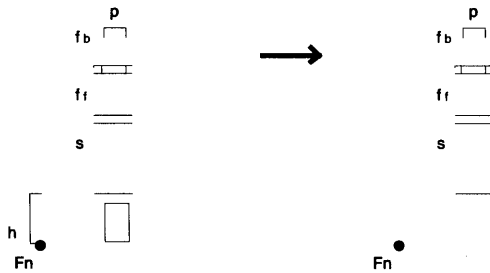
**Rule 83: Erase invisible part of patio opening**



On dimension:  
 $h = 1.50 \text{ m}$

On function  
 $f_i = \{ ya, te \}$   
 $F_n \in \{ 1, 2 \}$

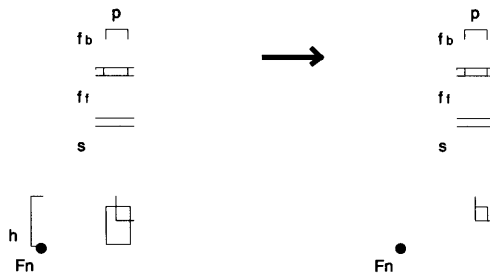
**Rule 84: Erase invisible patio opening**



On dimension:  
 $h = h_{fn}$

On function  
 $f_i = \{ \underline{ya}, te \}$   
 $F_n \in \{ 1, 2 \}$

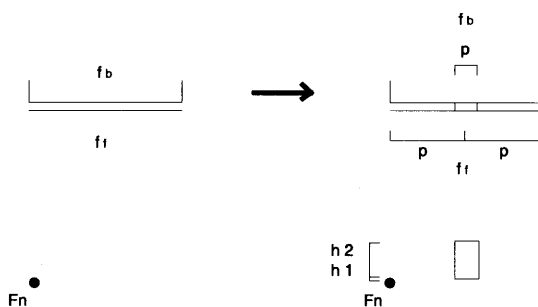
**Rule 85: Erase invisible patio opening**



On dimension:  
 $h = h_{fn}$

On function  
 $f_i = \{ \underline{ya}, te \}$   
 $F_n \in \{ 1, 2 \}$

**Rule 86: Pierce an exterior opening in the middle of a room's wall on the front facade**



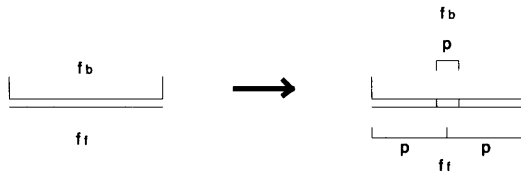
On dimension:  
 $w \in \{ 1.05, 1.10 \}$   
 $h_1 \in \{ 0.00, 0.20 \}$   
 $h_2 \in \{ 2.00, 2.07 \}$   
 $h_3 \in \{ 0.20, 0.35, 0.75 \}$   
 $h_4 \in \{ 2.00, 2.07 \}$

On function  
 $f_i = \{ ya, te, s \}$   
 $p(f_i) = f_i$   
 $F_n \in \{ 1, 2 \}$

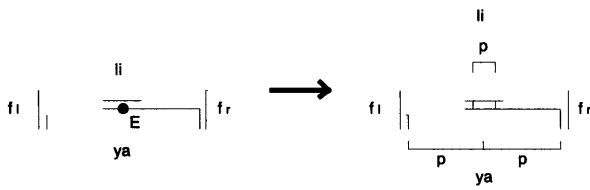
Figure 5.19 (continued) – Shape grammar rules



Rule 87: Pierce an exterior opening in the middle of a room's wall

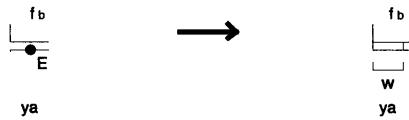


Rule 88: Pierce entrance door in the middle of the patio



Rule 89: Pierce entrance door abutting the living-room's wall

$f_l \in \{ li, st, ci \}$



Rule 90: Pierce an exterior opening abutting a-room's wall

$w = \{ 0.7, 0.8, 0.9 \}$

$f_f \in \{ s, ya, te \}$

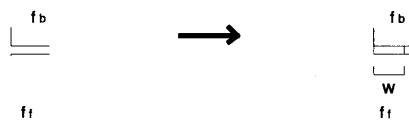


Figure 5.19 (continued) – Shape grammar rules

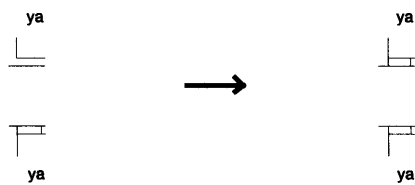
---

Rule 91: Pierce exterior openings in a row on the patio's wall



---

Rule 92: Pierce an exterior opening facing another in the interior



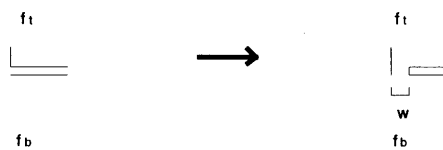
---

Rule 93: Pierce an exterior opening facing another in the exterior



---

Rule 94: Pierce an interior door next to a wall

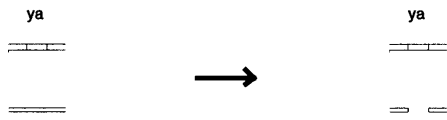


---

Figure 5.19 (continued) – Shape grammar rules

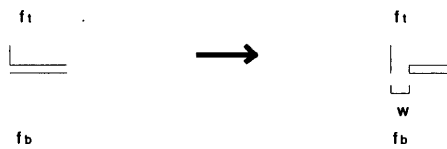
---

**Rule 95: Pierce interior opening facing exterior opening**



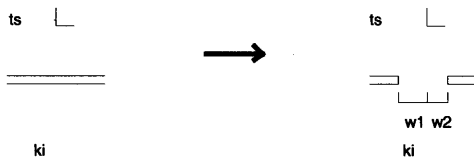
---

**Rule 96: Pierce interior opening in the middle of a wall**



---

**Rule 97: Pierce interior opening between kitchen and transitional space**



---

**Rule 98: Pierce interior opening on diagonal wall**



---

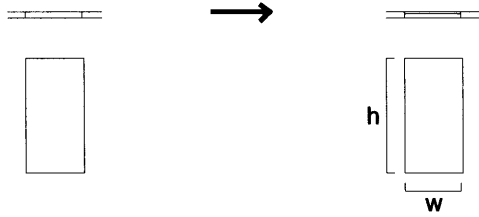
**Rule 99: Pierce interior opening on diagonal wall**



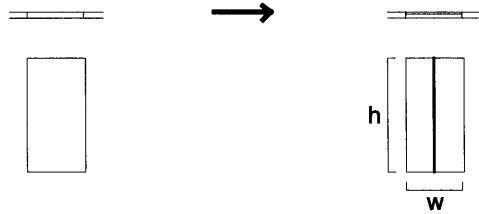
---

Figure 5.19 (continued) – Shape grammar rules

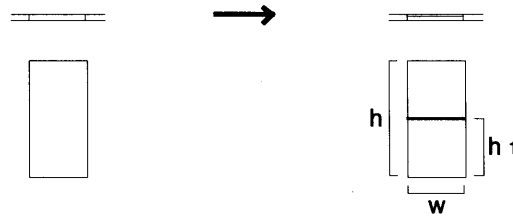
Rules 100-107: Introduce exterior openings mullions



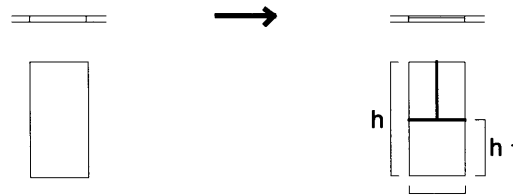
R100  
On dimension:  
 $w \in \{ 0.90, 1.05, 1.10 \}$   
 $h \in \{ 2.00, 2.07 \}$



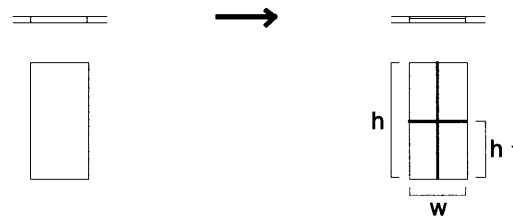
R101  
On dimension:  
 $w \in \{ 1.05, 1.10 \}$   
 $h \in \{ 2.00, 2.07 \}$



R102  
On dimension:  
 $w \in \{ 1.05, 1.10 \}$   
 $h_1 = h / 2 \vee h_1 = w$

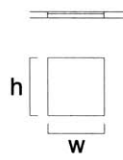
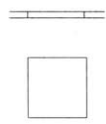


R103  
On dimension:  
 $w \in \{ 1.05, 1.10 \}$   
 $h_1 = h / 2 \vee h_1 = w$

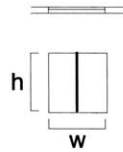
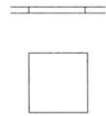


R104  
On dimension:  
 $w \in \{ 1.05, 1.10 \}$   
 $h_1 = h / 2 \vee h_1 = w$

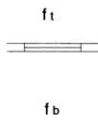
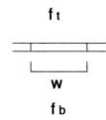
Figure 5.19 (continued) – Shape grammar rules



R105  
On dimension:  
 $w \in \{ 1.05, 1.10 \}$   
 $h = w$



R106  
On dimension:  
 $w \in \{ 1.05, 1.10 \}$   
 $h = w$



R107  
On dimension:  
 $w \in \{ 0.90, 1.05, 1.10 \}$   
On function  
 $f_b \in \{ te, ya, s \}$   
 $p(f_b) \neq f_b$

Stage 1.7: Terminate

Rule 108: Erase unnecessary labelled shape



Rule 109: Change state



$R'_{F1} = \emptyset$



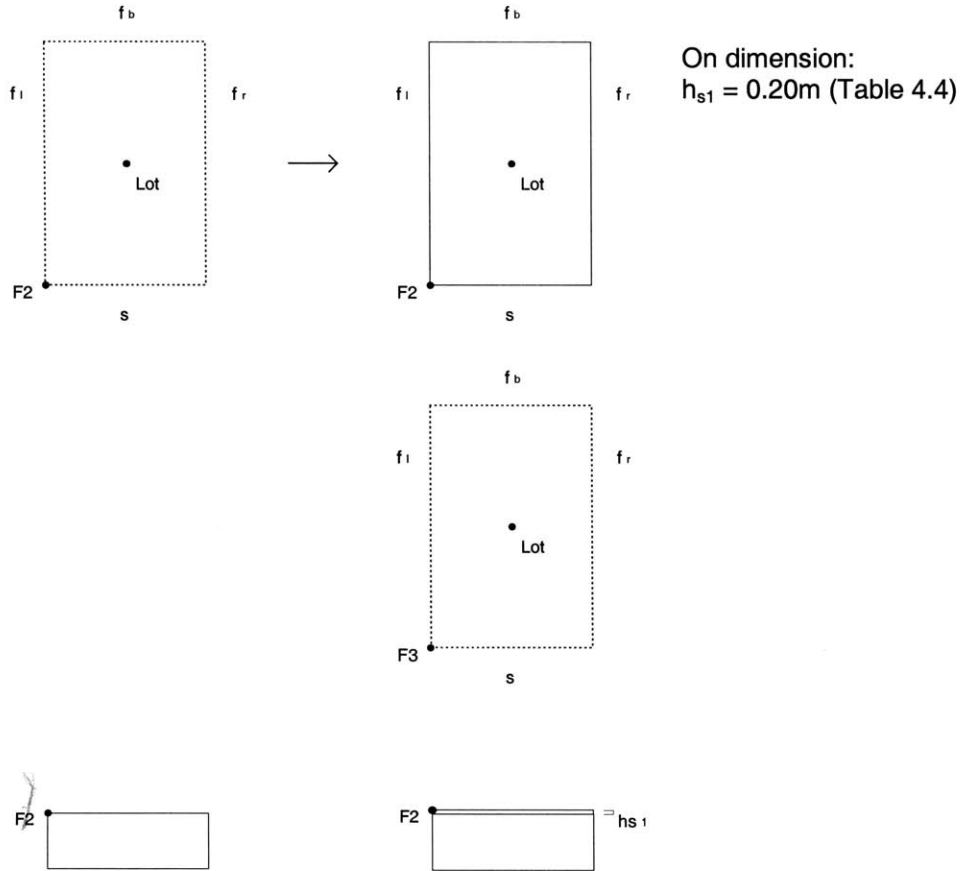
R103:  $\langle F1; F2 \rangle$

Figure 5.19 (continued) – Shape grammar rules

Stage 2: Define second floor

Step 2.1: Start

Rule 110: Introduce slab



(Rule 2 applies): Introduce enclosing walls

Rule 111: Adjust front and back wall thickness

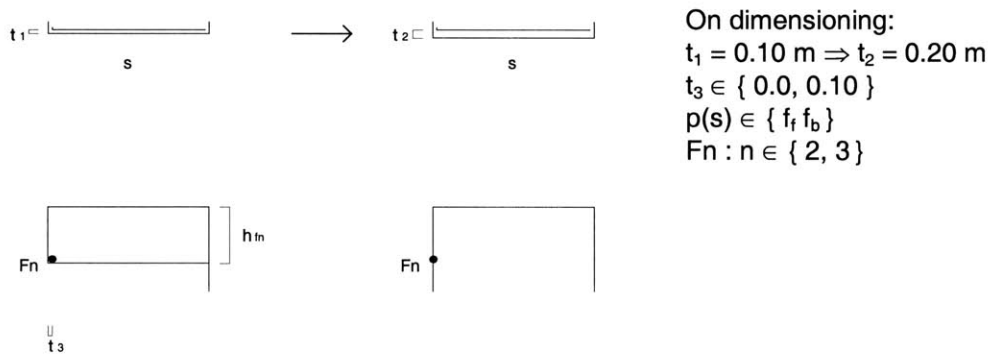
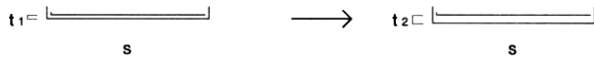
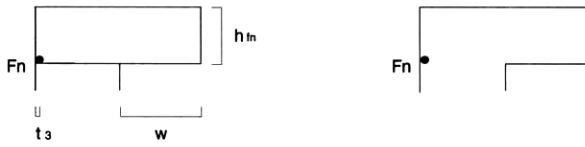


Figure 5.19 (continued) – Shape grammar rules

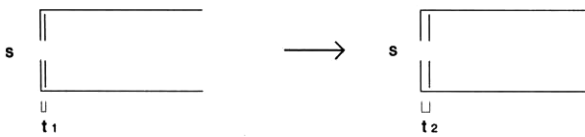
**Rule 112: Adjust front and back wall thickness**



On dimensioning:  
 $t_1 = 0.10 \text{ m} \Rightarrow t_2 = 0.20 \text{ m}$   
 $t_3 \in \{ 0.0, 0.10 \}$   
 $p(s) \in \{ f_l, f_r \}$   
 $F_n : n \in \{ 2, 3 \}$



**Rule 113: Adjust left and right wall thickness**



On dimensioning:  
 $t_1 = 0.10 \text{ m} \Rightarrow t_2 = 0.20 \text{ m}$   
 $t_3 \in \{ 0.0, 0.10 \}$   
 $p(s) \in \{ f_l, f_r \}$   
 $F_n : n \in \{ 2, 3 \}$

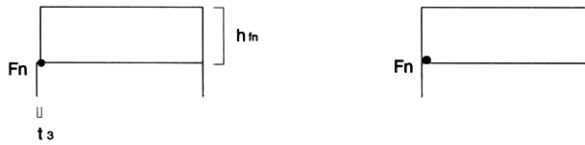
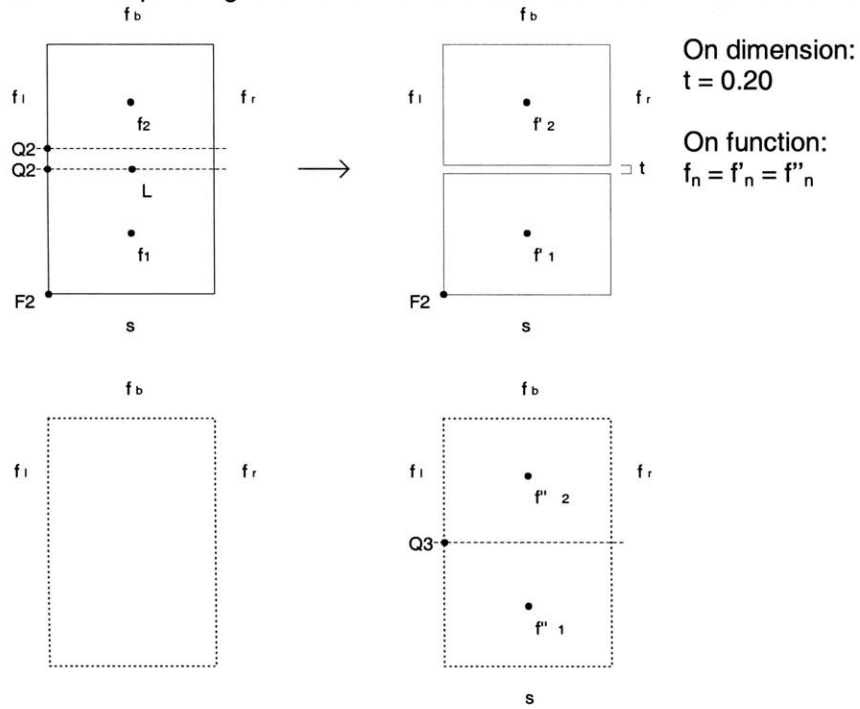


Figure 5.19 (continued) – Shape grammar rules

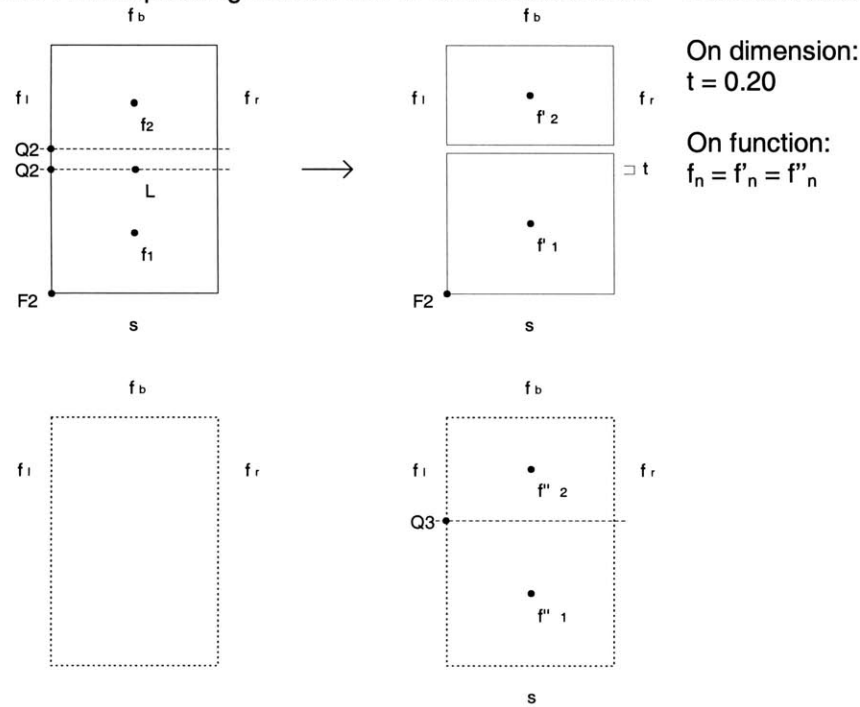
Step 2.2: Replicating first floor's division into functional zones

Rule 114: Replicating division into in- and outside zones First level dissection



R114:  $\langle F2, s, L: f'_1, f'_2 \rangle$

Rule 115: Replicating division into in- and outside zones First level dissection



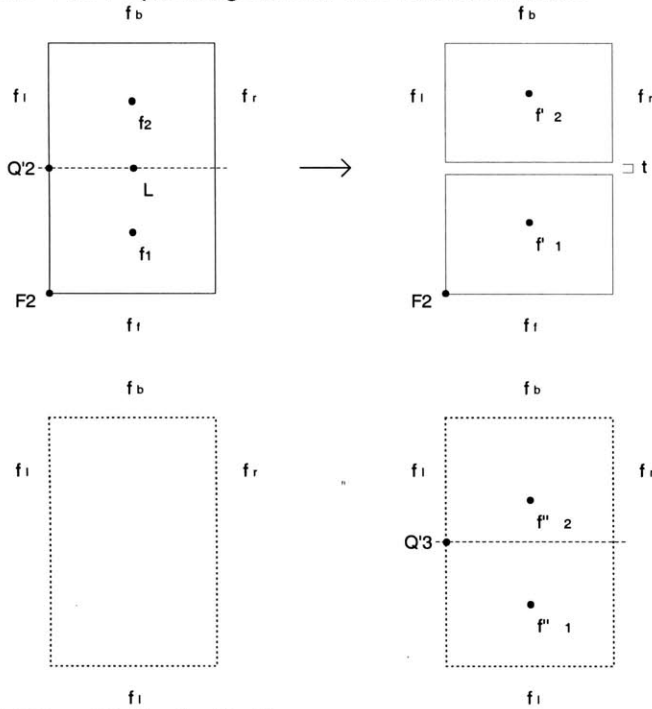
R115:  $\langle F2, s, L: f'_1, f'_2 \rangle$

Figure 5.19 (continued) – Shape grammar rules



Rule 116: Replicating division into functional zones

Second level dissections



On dimension:  
 $t = 0.20 \text{ m}$

On function:

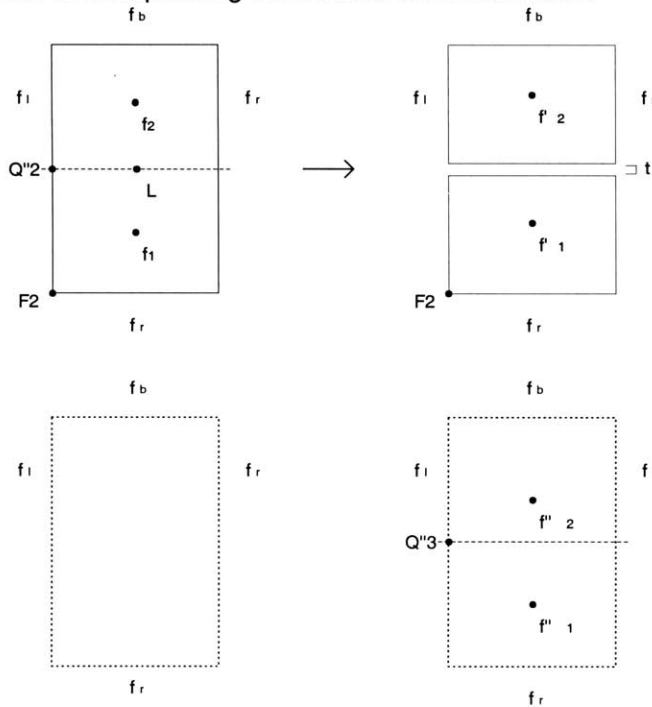
$f_n = o \Rightarrow f'_n = y \wedge f''_n = y$   
 $f_n = i \Rightarrow f'_n = i \wedge f''_n = i$   
 $f_n = i' \Rightarrow f'_n = i' \wedge f''_n = i'$

$n \in \{1, 2\}$

R116:  $\langle F2, f'_1, f'_2, f''_1, f''_2 \rangle$

Rule 117: Replicating division into functional zones

Third level dissections



On dimension:  
 $t = 0.07 \text{ m}$

On function:

$f_n = o \Rightarrow f'_n = y \wedge f''_n = y$   
 $f_n = i \Rightarrow f'_n = i \wedge f''_n = i$   
 $f_n = i' \Rightarrow f'_n = i' \wedge f''_n = i'$

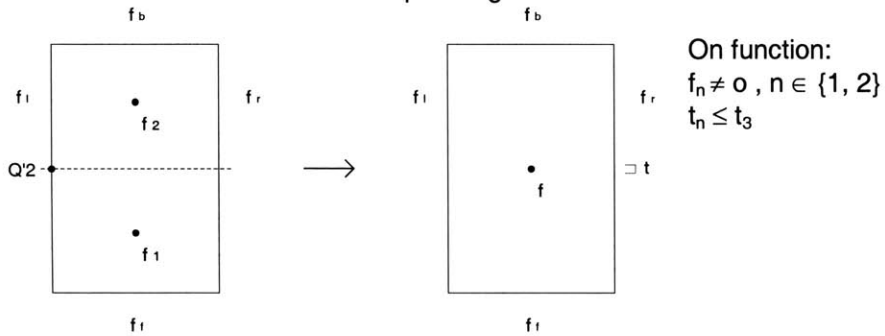
$n \in \{1, 2\}$

$t_n \leq t_3$

R118:  $\langle F2, f'_1, f'_2, f''_1, f''_2 \rangle$

Figure 5.19 (continued) – Shape grammar rules

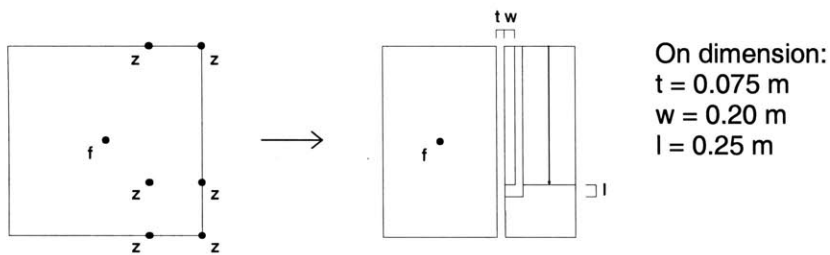
Rule 119: Erase marker to avoid replicating dissection of functional zones



R119:  $\langle F2, f_1, f_2: f \rangle$

Step 2.3: Define circulation

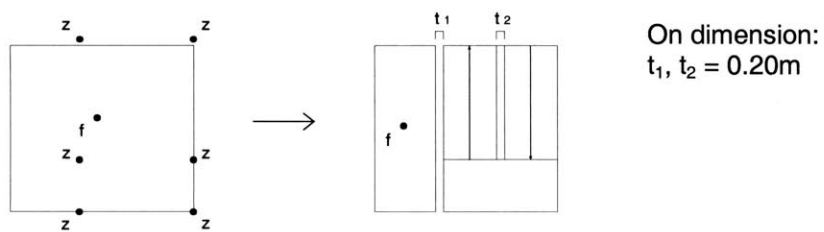
Rule 120: Extend I- or L-shaped staircase



R120:  $\langle F2, i: i, st \rangle$

$f \in \{i, i'\}$

Rule 121: Extend U-shaped staircase

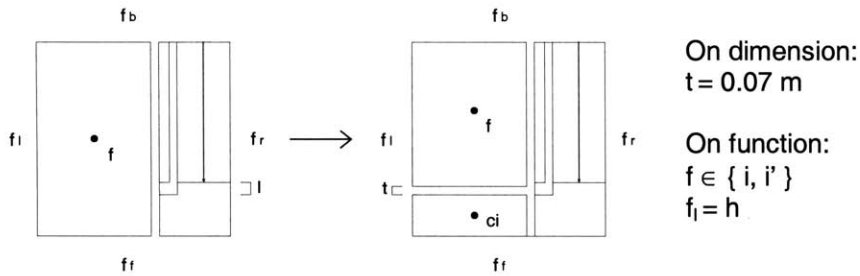


R121:  $\langle F2, f: f, st \rangle$

$f \in \{i, i'\}$

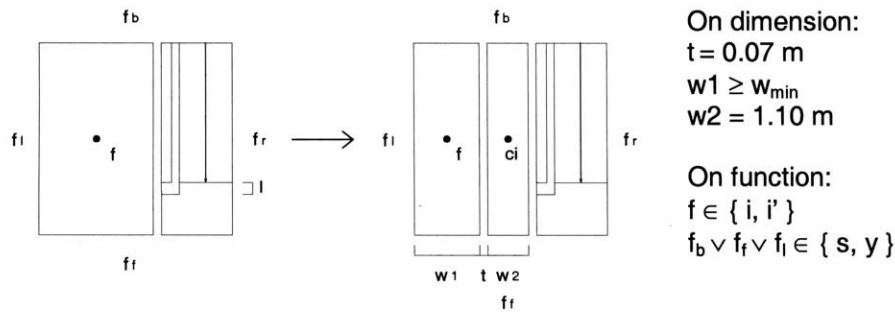
Figure 5.19 (continued) – Shape grammar rules

Rule 122: Create circulation perpendicular to I- or L-shaped staircase



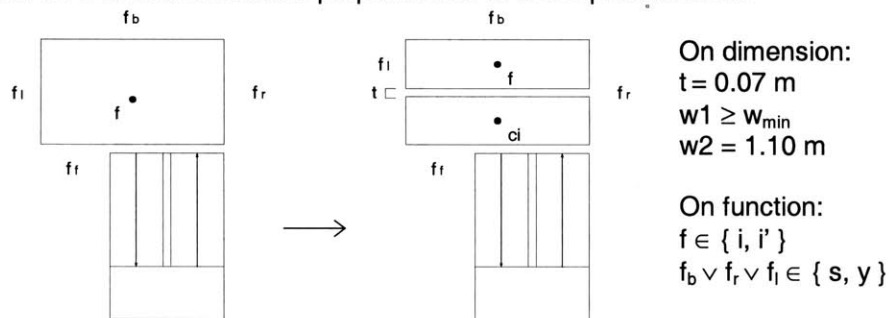
R122:  $\langle F2, f_b, f_r, f_t, f_l, f: f, c_i \rangle$

Rule 123: Create circulation parallel to I- or L-shaped staircase



R123:  $\langle F2, f_b, f_r, f_t, f_l, f: f, c_i \rangle$

Rule 124: Create circulation perpendicular to U-shaped staircase

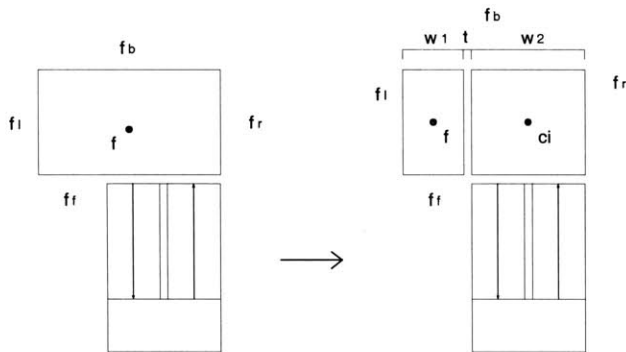


R124:  $\langle F2, f_b, f_r, f_t, f_l, f: f, c_i \rangle$

$f \in \{i, i'\}, f_l = s$

Figure 5.19 (continued) – Shape grammar rules

Rule 125: Create circulation parallel to U-shaped staircase



On dimension:

$$t = 0.07 \text{ m}$$

$$w1 \geq w_{\min}$$

$$w2 = 1.10 \text{ m}$$

On function:

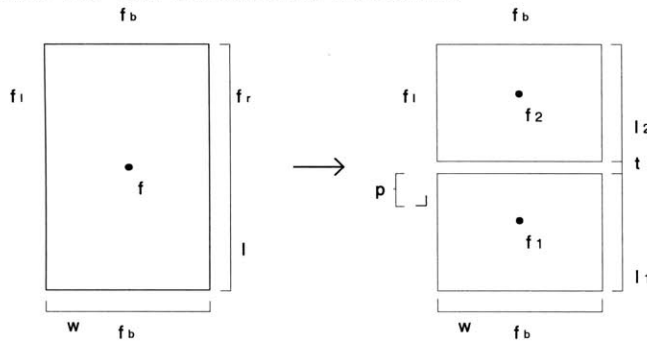
$$f \in \{i, i'\}$$

$$f_b \vee f_f \vee f_i \in \{s, y\}$$

R124:  $\langle F2, f_b, f_r, f_f, f_i, f, ci \rangle$

Step 2.4: Define rooms

Rules 125-131: Divide zones into rooms



Rule 125: Dissect inside zone to create bedroom

R125:  $\langle F2, f_b, f_r, f_f, f_i, i; i, be \rangle$

Rule 126: Dissect inside zone to create bathroom

R126:  $\langle F2, f_b, f_r, f_f, f_i, i; i, ba \rangle$

Rule 127: Dissect bedroom to create circulation

R127:  $\langle F2, f_b, f_r, f_f, f_i, be; be, ci \rangle$

Rule 128: Dissect circulation to create bathroom

R128:  $\langle F2, f_b, f_r, f_f, f_i, ci; ci, ba \rangle$

Rule 129: Dissect bathroom to create circulation

R129:  $\langle F2, f_b, f_r, f_f, f_i, ba; ba, ci \rangle$

Rule 130: Dissect inside zone to create terrace

R130:  $\langle F2, f_b, f_r, f_f, f_i, i; i, te \rangle$

Rule 131: Dissect yard zone to create terrace

R131:  $\langle F2, f_b, f_r, f_f, f_i, y; y, te \rangle$

$$p \geq 1.0 \text{ m}$$

$$\exists f_b : f_b \notin \{i, i'\}, R'_{be} \neq \emptyset$$

$$p \geq 1.0 \text{ m}$$

$$f_b \in \{h, s\} \wedge f_r \neq st \wedge f_i \neq st$$

$$f_b = ci \wedge f_r = be$$

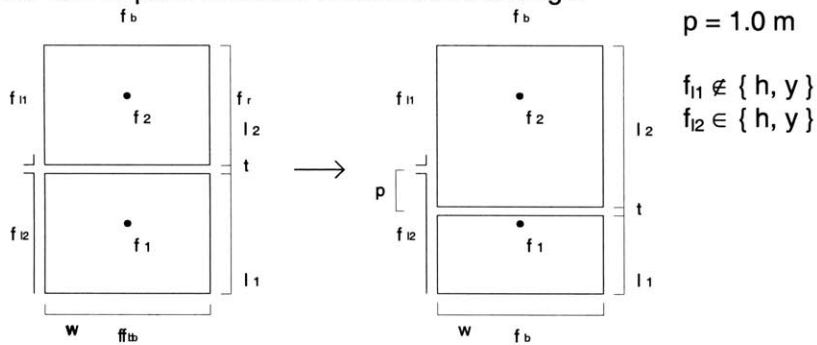
$$p \geq 1.0 \text{ m}$$

$$f_b \in \{h, s\} \wedge f_r \neq st \wedge f_i \neq st$$

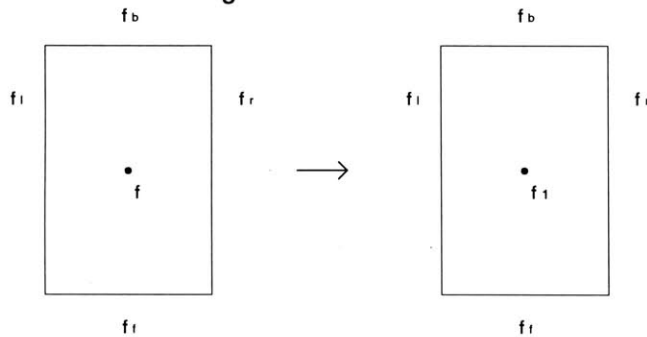
$$R'_{sl} = \emptyset$$

Figure 5.19 (continued) – Shape grammar rules

Rule 132: Expand bedroom to achieve natural light



Rules 133-136: Assign functions



Rule 133: Assigning the bedroom function to an interior room

R133:  $\langle F2, f_b, f_r, f_f, f_i, i: be \rangle$   $f = i, R'_{sl} \neq \emptyset$

Rule 134: Assigning the terrace function to an interior room that resulted from the dissection of the outside zone

R134:  $\langle F2, f_b, f_r, f_f, f_i, i': te \rangle$   $f = i', R'_{sl} = \emptyset$

Rule 135: Assigning the patio function to an exterior room that resulted from the dissection of the outside zone

R135:  $\langle F2, f_b, f_r, f_f, f_i, o: y \rangle$   $f = o$

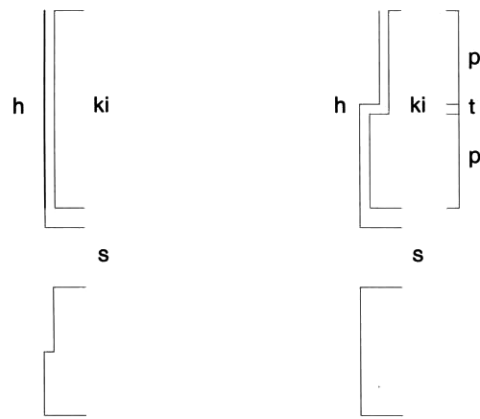
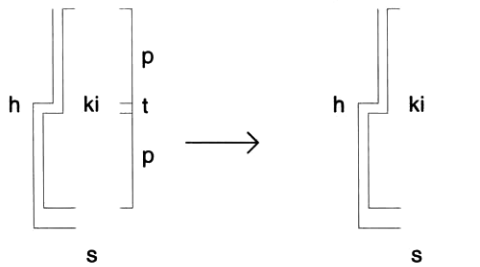
(Rule 98 applies): Concatenate adjacent rooms

Figure 5.19 (continued) – Shape grammar rules

Step 2.5: Introduce details

Rules 137-142: Extend chimneys

R137



R138

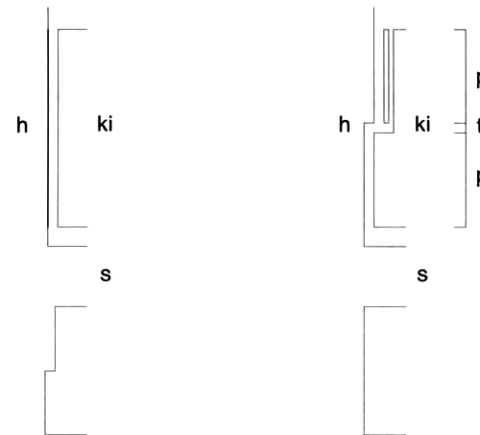
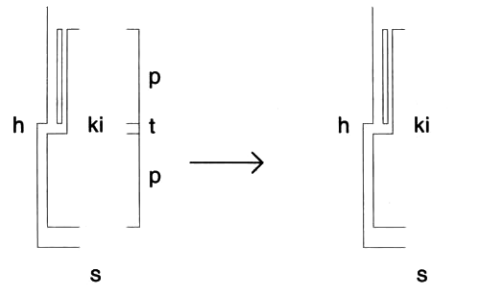


Figure 5.19 (continued) – Shape grammar rules

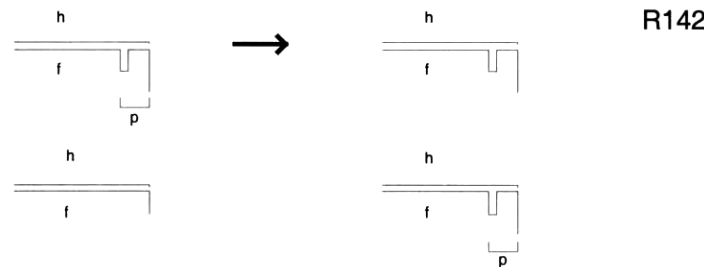
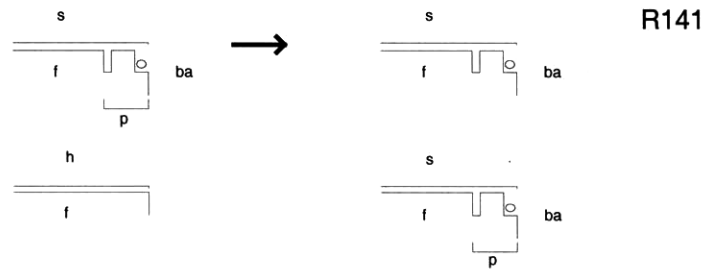
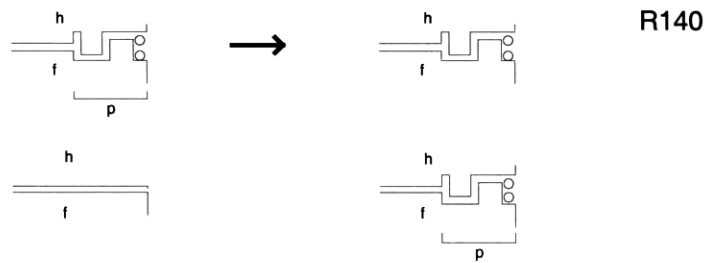
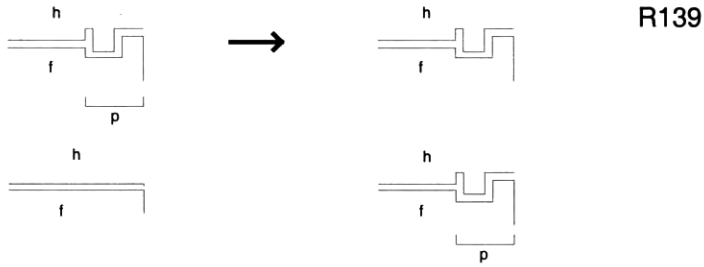
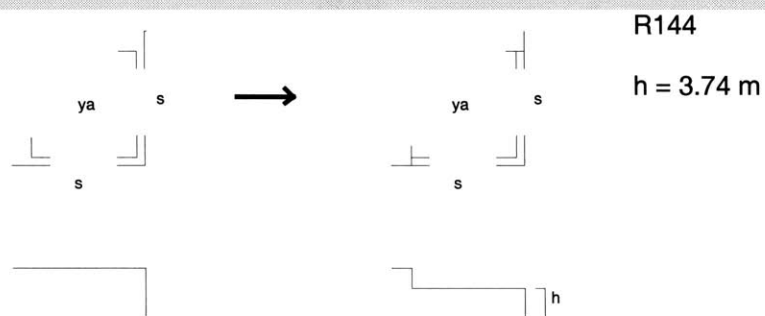
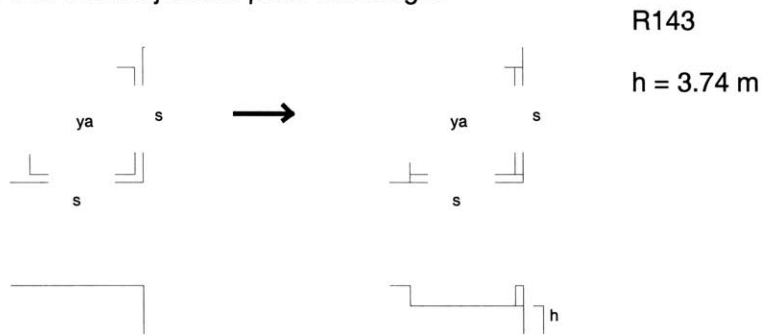


Figure 5.19 (continued) – Shape grammar rules

Rules 143-146: Adjust the patio wall height



Stage 2.6: Introduce openings

- (Rules 81-93 applies): Pierce exterior openings
- (Rules 94-99 applies): Pierce interior openings
- (Rules 100-107 applies): Introduce exterior openings mullions

Stage 2.7: Terminate

Rule 147: Erase unnecessary labelled shape

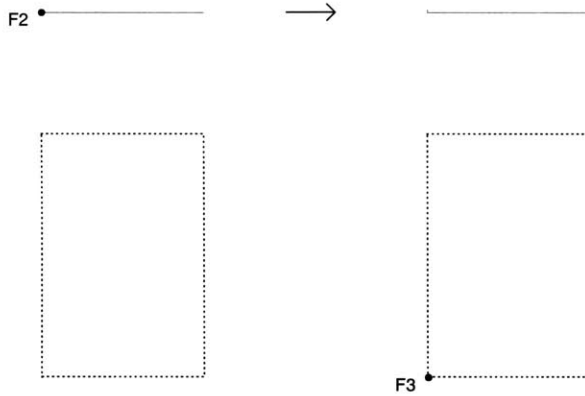


Figure 5.19 (continued) – Shape grammar rules



Rule 148: Change state

$$R'_{F1} = \emptyset$$



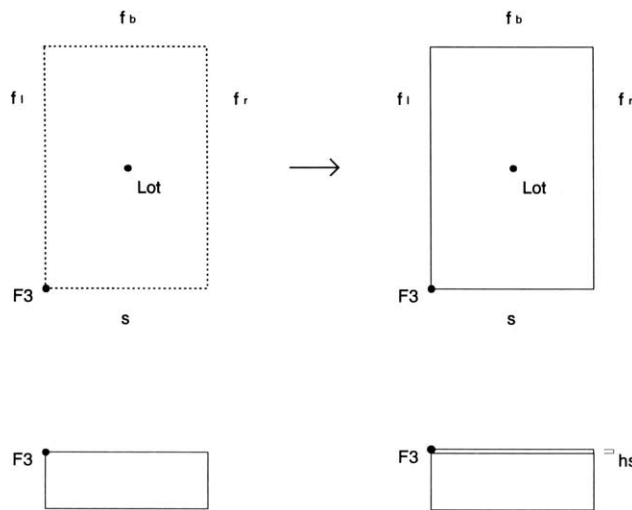
R148:  $\langle F1; F2 \rangle$

Stage 3: Define terrace

Step 3.1: Start

Rule 149: Introduce slab

$$R'_{F2} = \emptyset$$

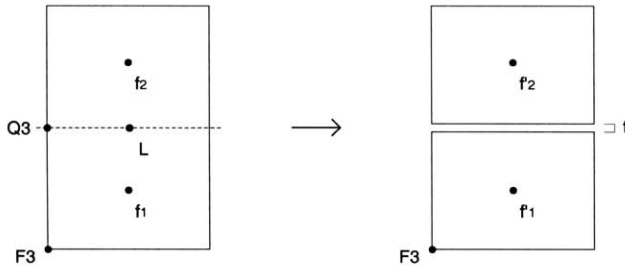


(Rule 2 applies): Introduce enclosing walls  
 (Rules 3-4 apply): Adjust wall thickness

Figure 5.19 (continued) – Shape grammar rules

**Step 3.2: Replicating second floor's division into functional zones**

**Rule 150: Replicating dissection into in- and outside z. First level dissection**



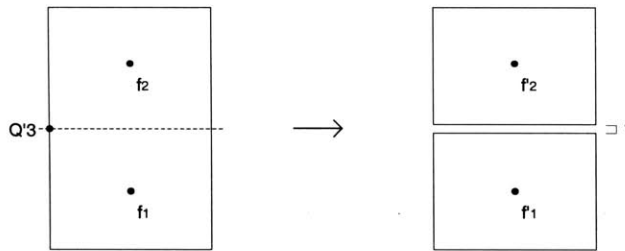
On dimension:  
 $T = 0.20$

R150:  $\langle F3, L, f_1, f_2: f'_1, f'_2 \rangle$

$f'_n = f_n, n \in \{1, 2\}$

**Rule 151: Replicating division into functional zones**

**Second level dissection**



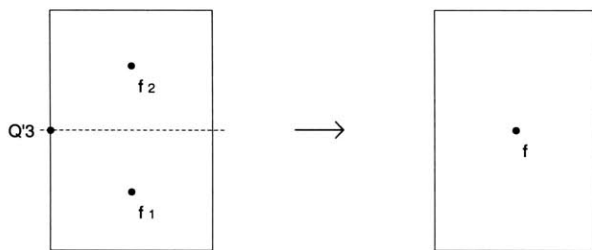
On dimension:  
 $T = 0.20$

On function:  
 $f_n = y \Rightarrow f'_n = y$   
 $f_n = te \Rightarrow f'_n = y$   
 $f_n \notin \{y, te\} \Rightarrow f'_n = te$

$n \in \{1, 2\}$

R151:  $\langle F3, f_1, f_2: te, te \rangle$

**Rule 152: Erase marker to avoid replicating dissection**



R152:  $\langle F3, f_2, f_3: f \rangle$

$\forall f_n = i \Rightarrow f_n = te$

**Rule 153: Pierce patio**

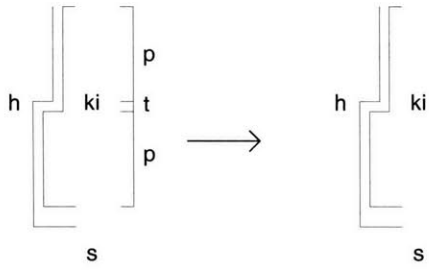
R153:  $\langle F_n, y: ya \rangle$

$F_n \in \{2, 3\}$

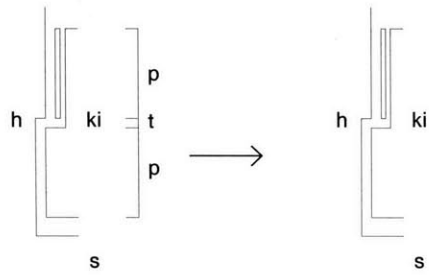
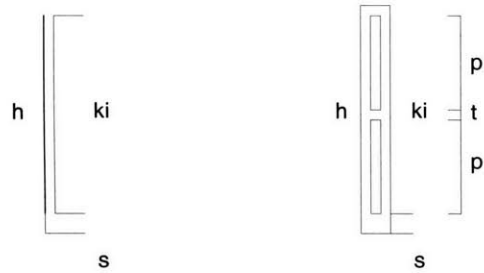
Figure 5.19 (continued) – Shape grammar rules

Step 3.3: Introduce details

Rules 154-159: Extend chimney



R154



R155

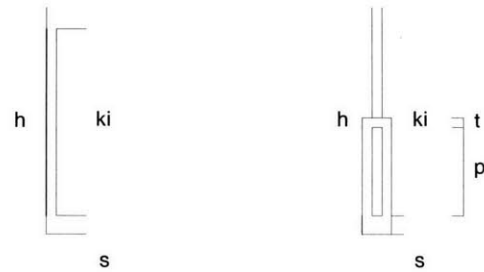


Figure 5.19 (continued) – Shape grammar rules

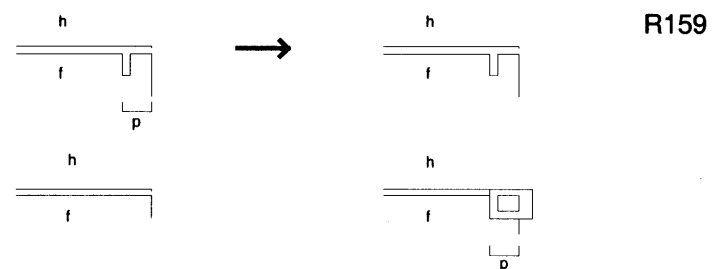
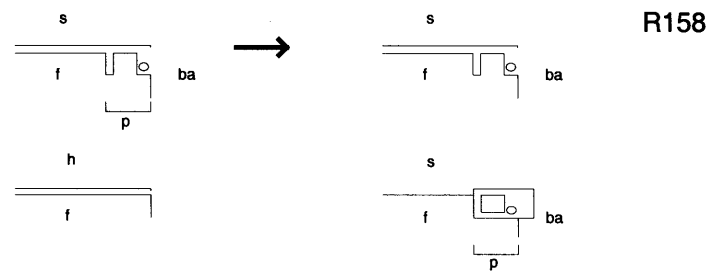
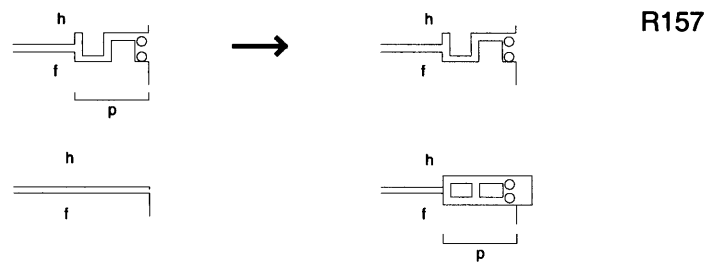
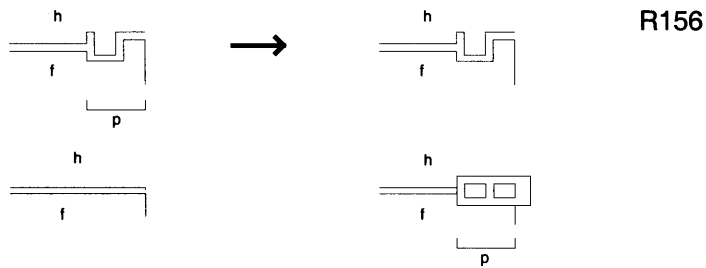
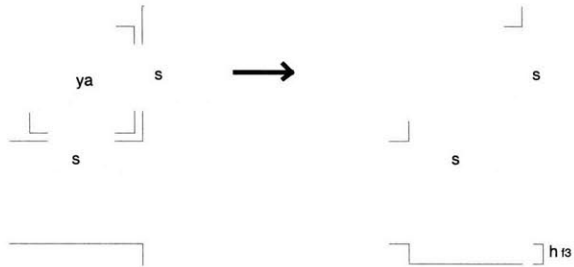


Figure 5.19 (continued) – Shape grammar rules

---

Rule 160: Adjust the height of the patio wall



---

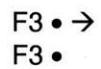
Step 3.4: Terminate

Rule 161: Erase unnecessary labelled shape



---

Rule 162: Erase state labels



---

Figure 5.19 (continued) – Shape grammar rules

## **5.10 Derivation of an existing design**

The generation of a subtype Ab, five-bedroom house is provided below (Figure 5.19.)

This house was selected because it corresponds to the first mature design produced in the language, constituting the prototypical design for frontyard houses. The rules applied during the derivation are shown below the arrows between design states.

0: Introduce initial shape

1: Define 1st floor

1.1: Starting

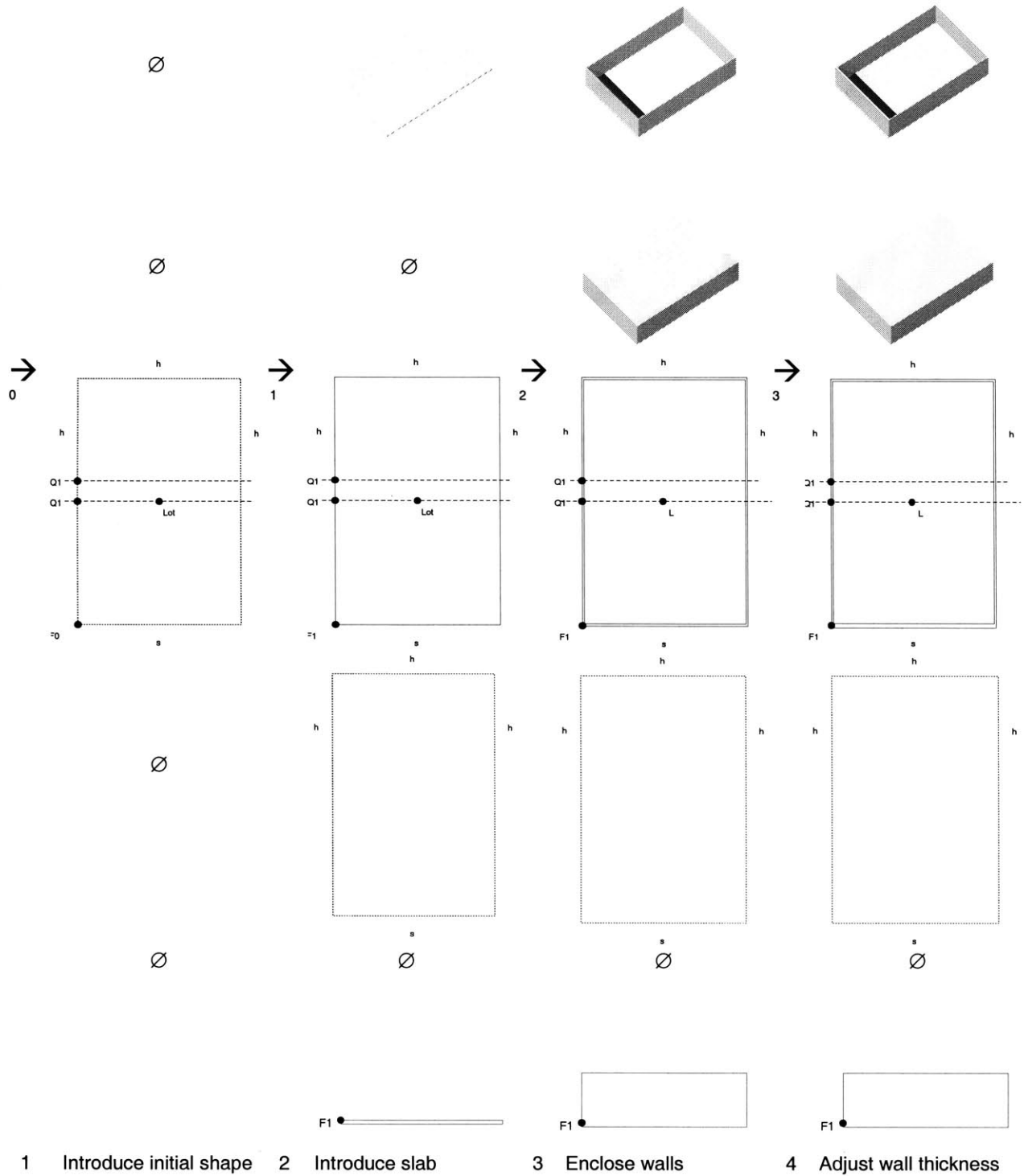


Figure 5.20 – Derivation of an existing design (Tabt5.)

1.2: Locate functional zones      1.3: Define circulation

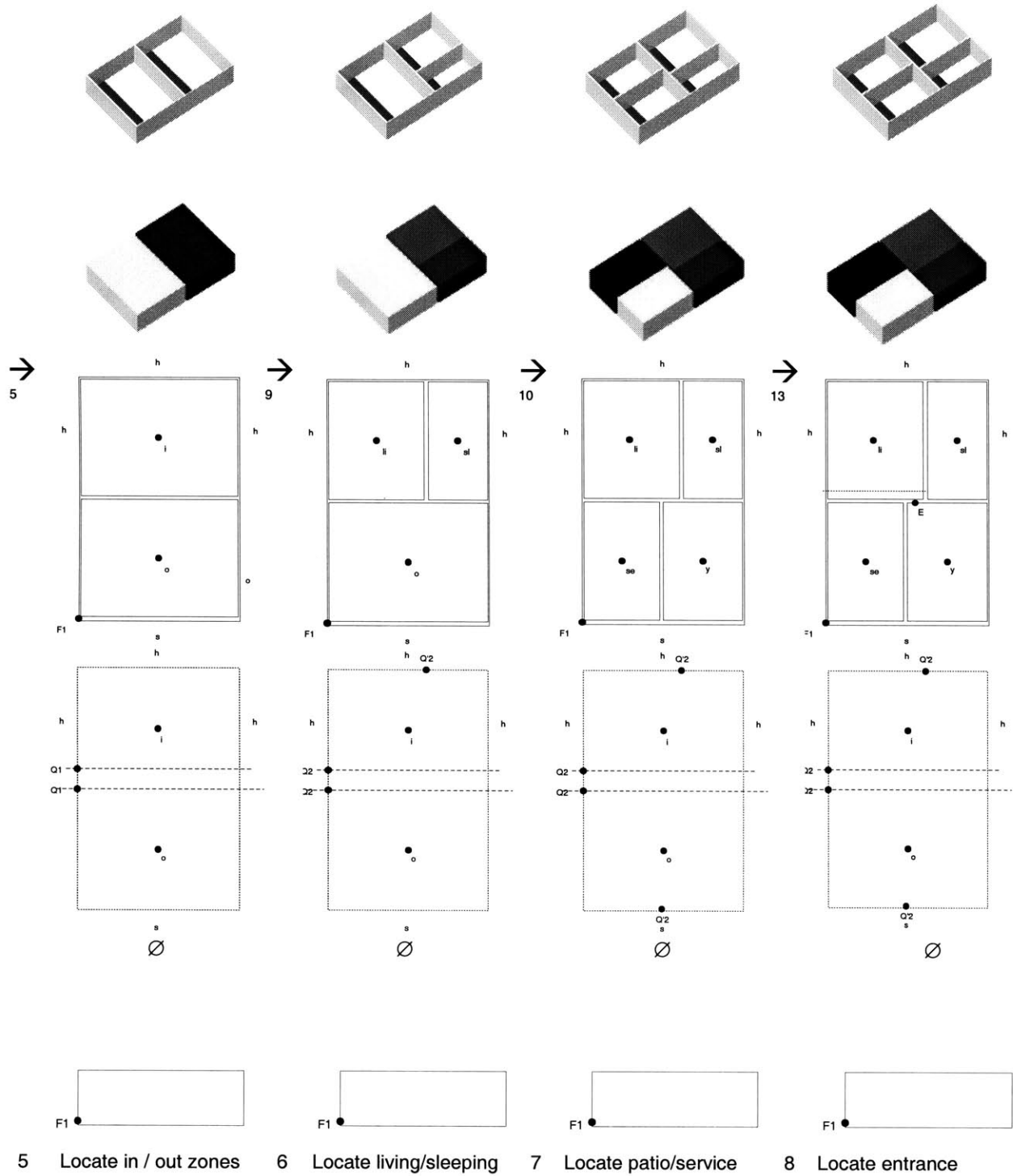


Figure 5.20 (continued) – Derivation of an existing design (Tabt5.)





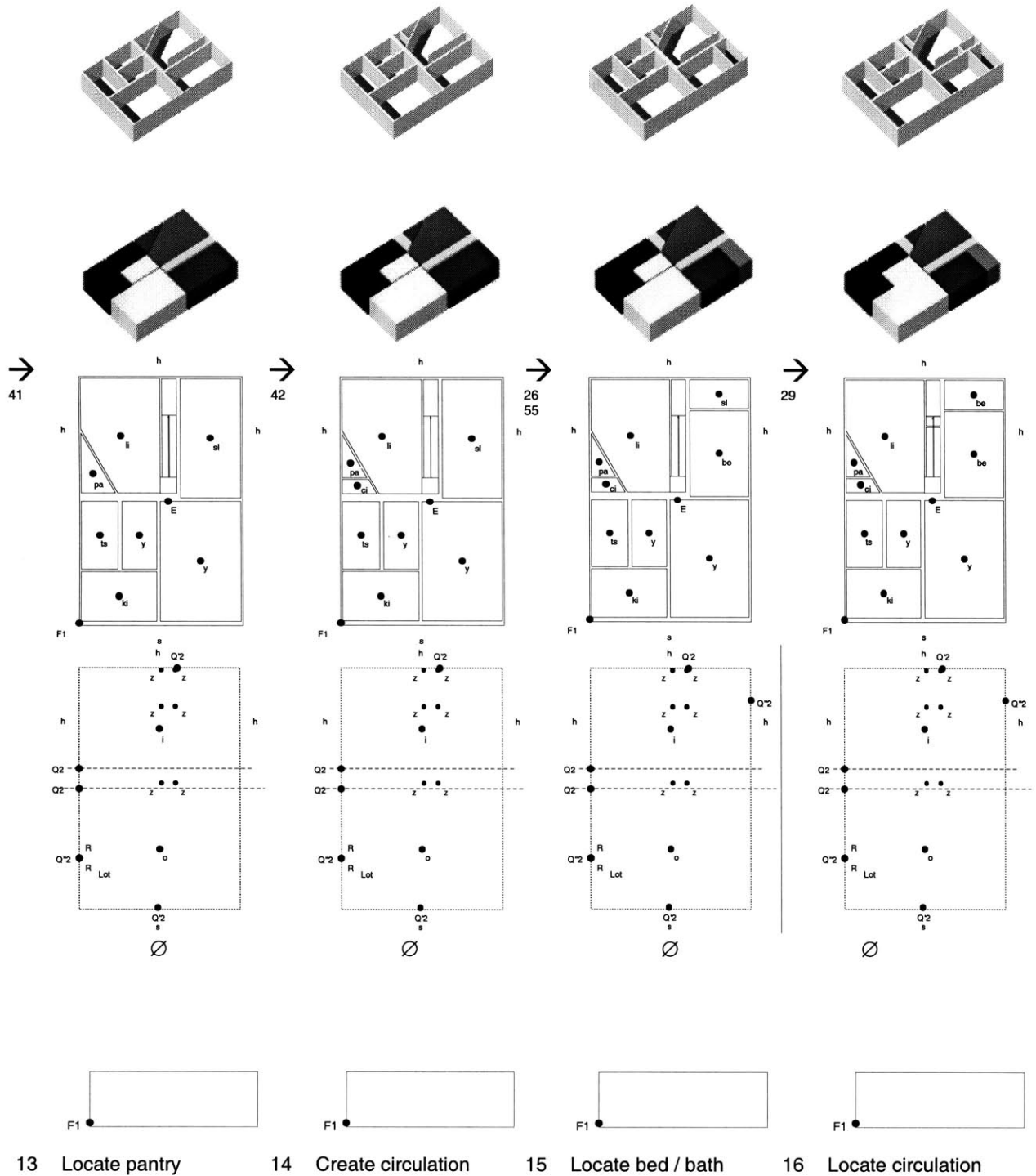


Figure 5.20 (continued) – Derivation of an existing design (Tabt5.)

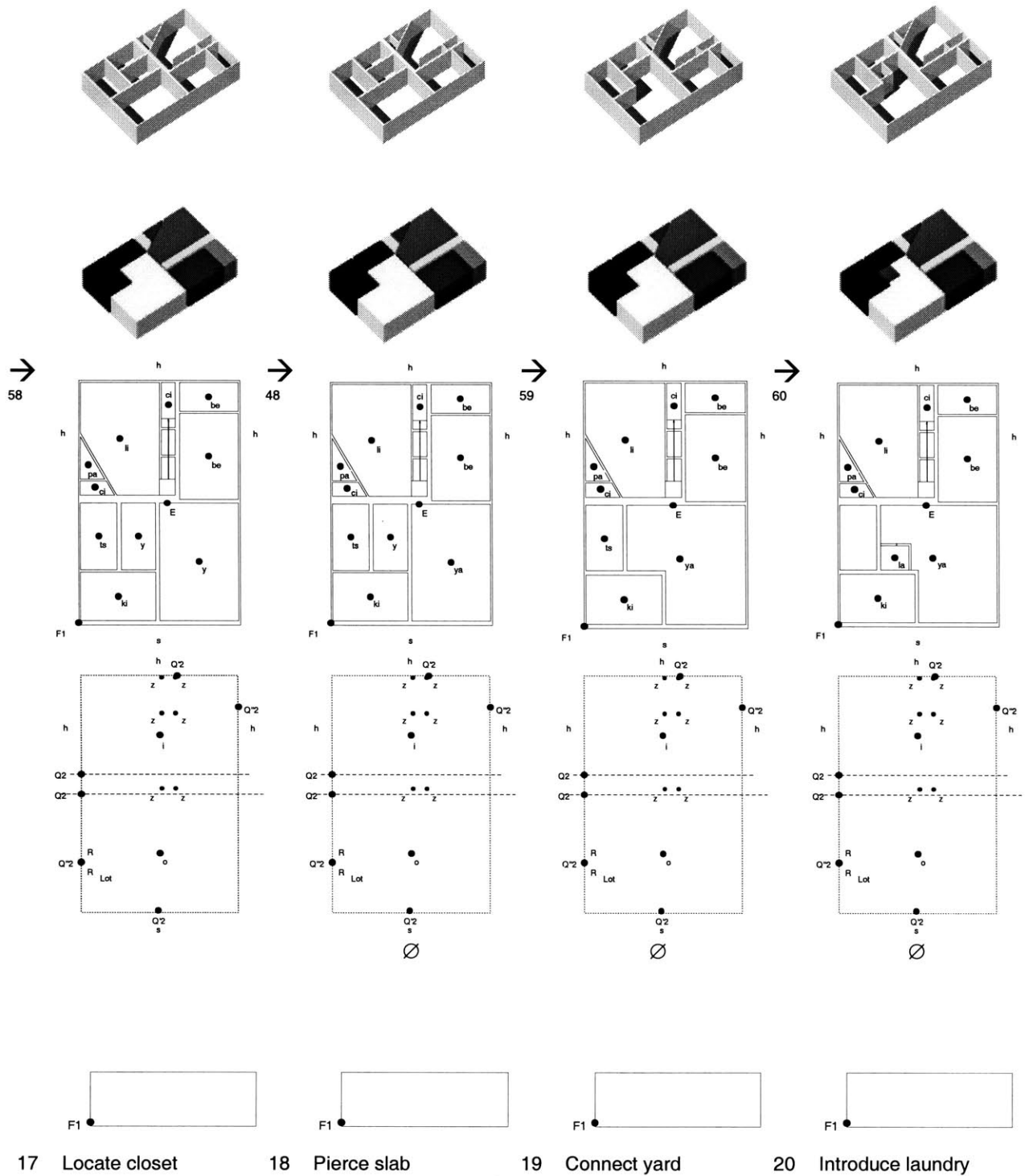


Figure 5.20 (continued) – Derivation of an existing design (Tabt5.)

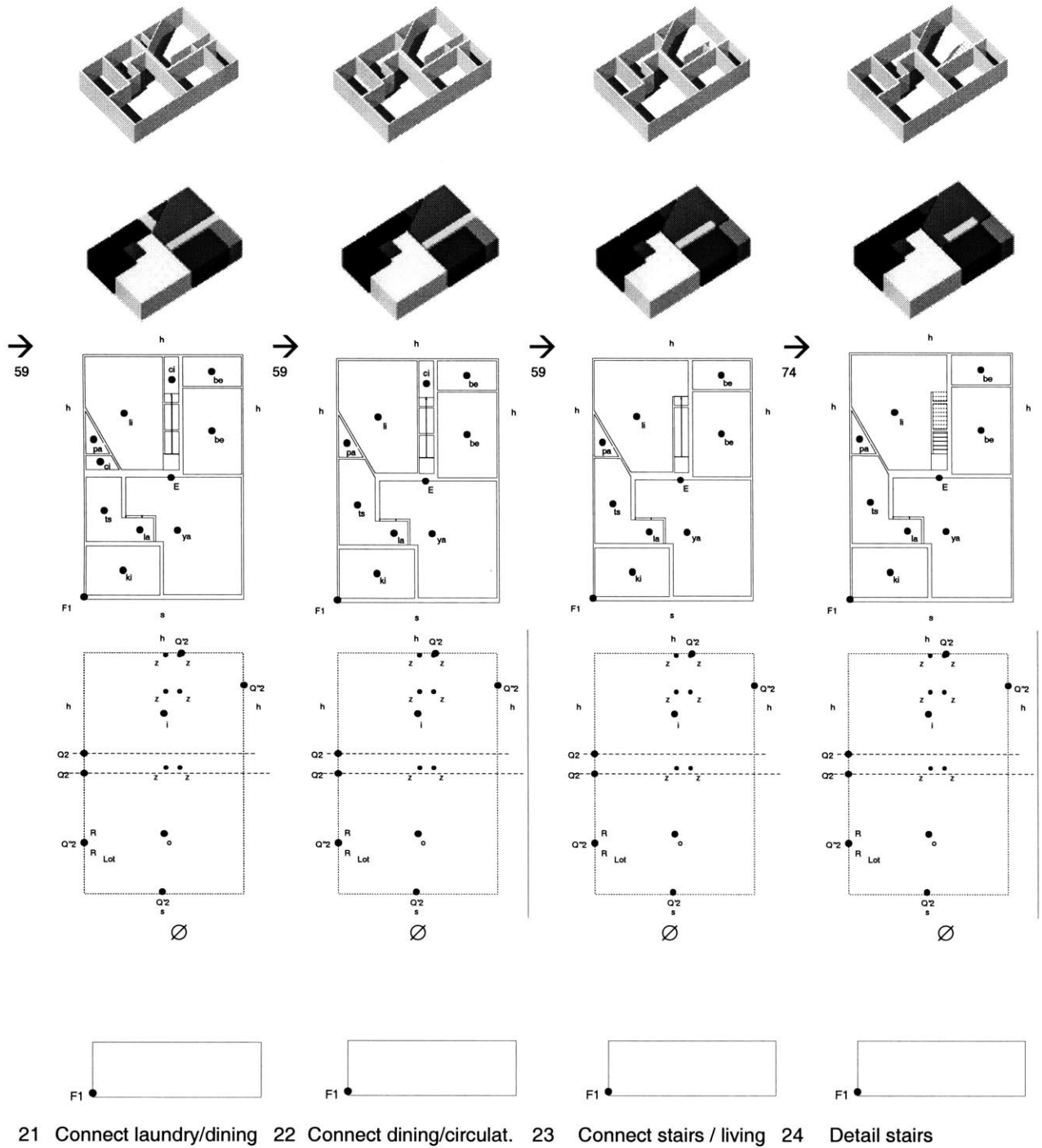


Figure 5.20 (continued) – Derivation of an existing design (Tabt5.)

1.6: Introduce openings

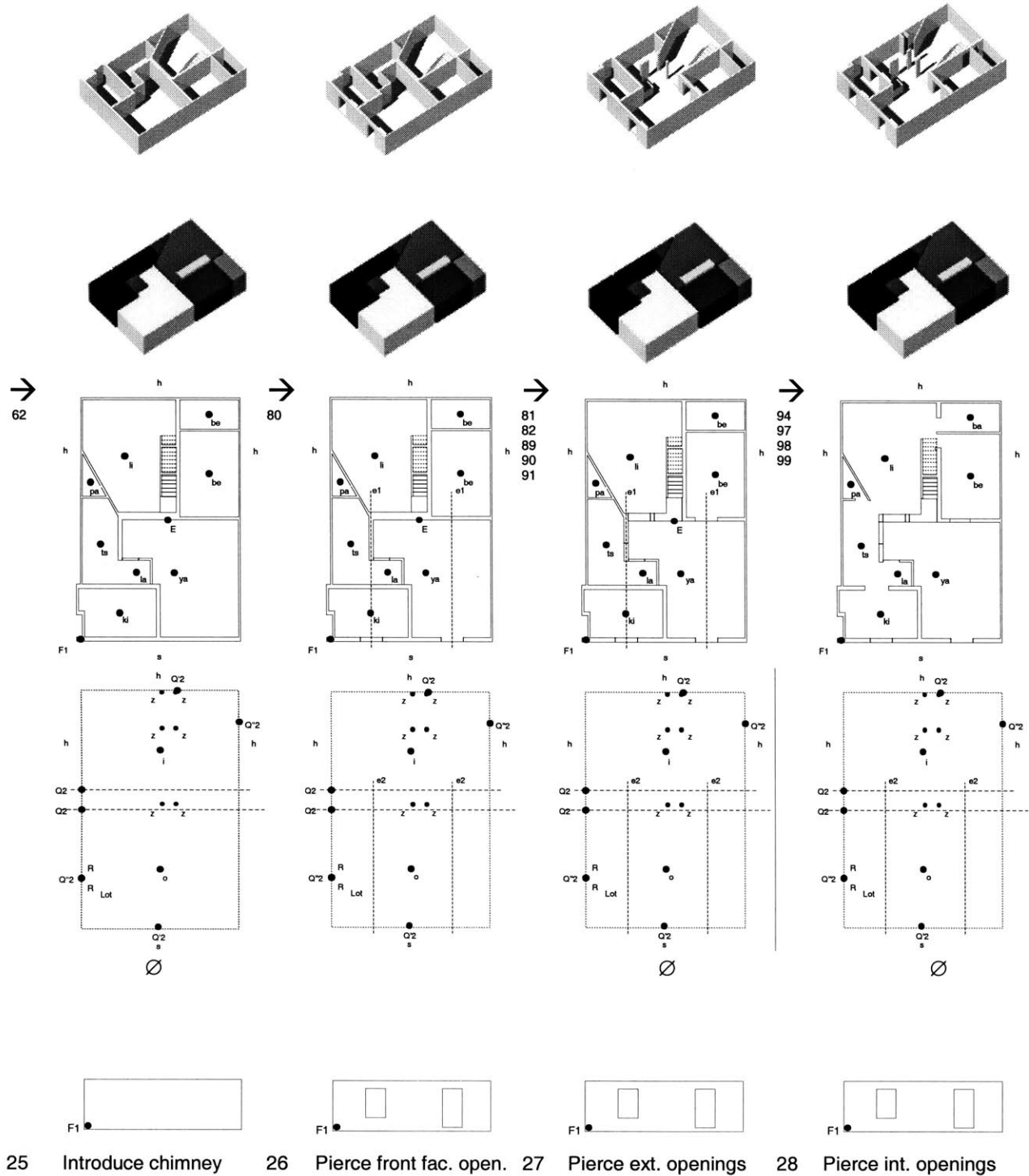


Figure 5.20 (continued) – Derivation of an existing design (Tabt5.)

1.7: Terminating      2: Define 2nd floor      2.1: Introduce slab

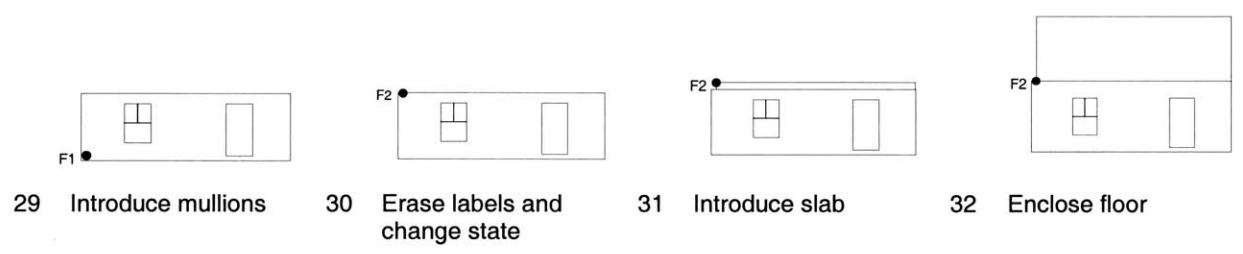
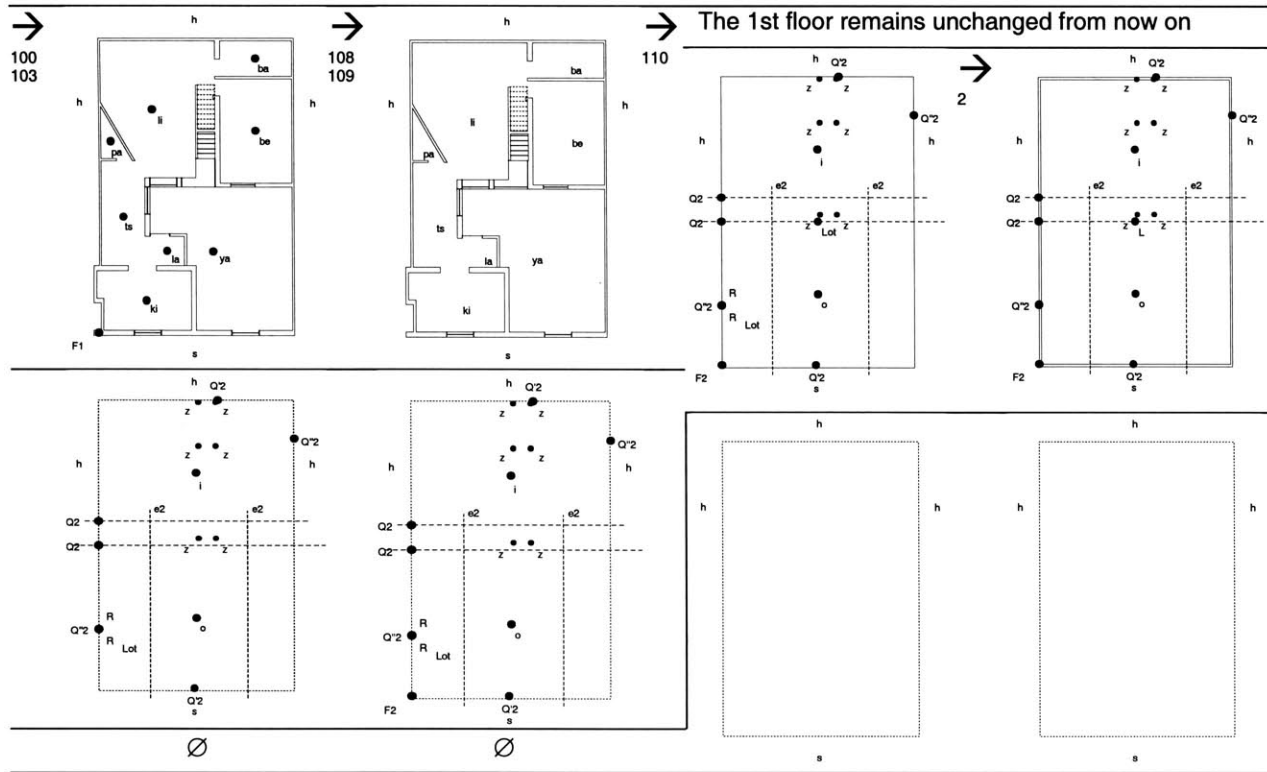
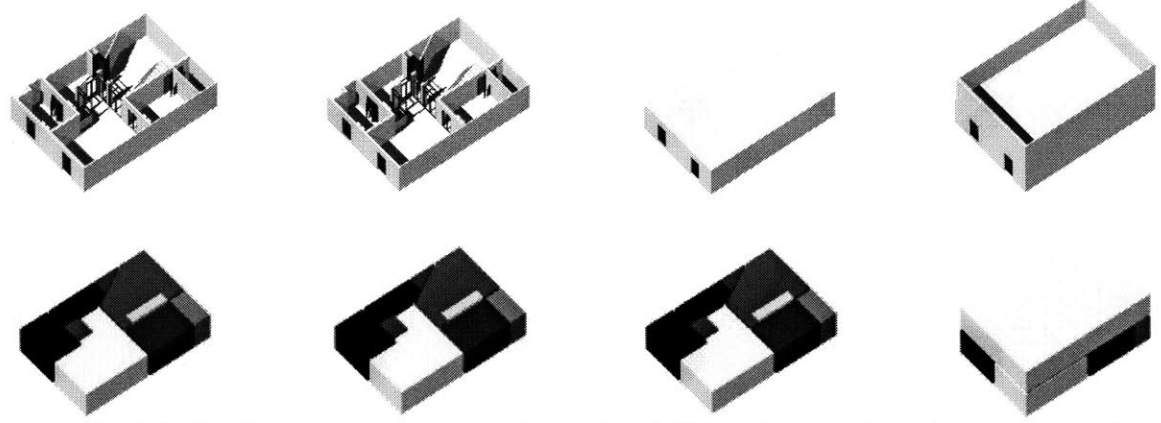


Figure 5.20 (continued) – Derivation of an existing design (Tabt5.)

## 2.2: Replicate 1st floor divisions

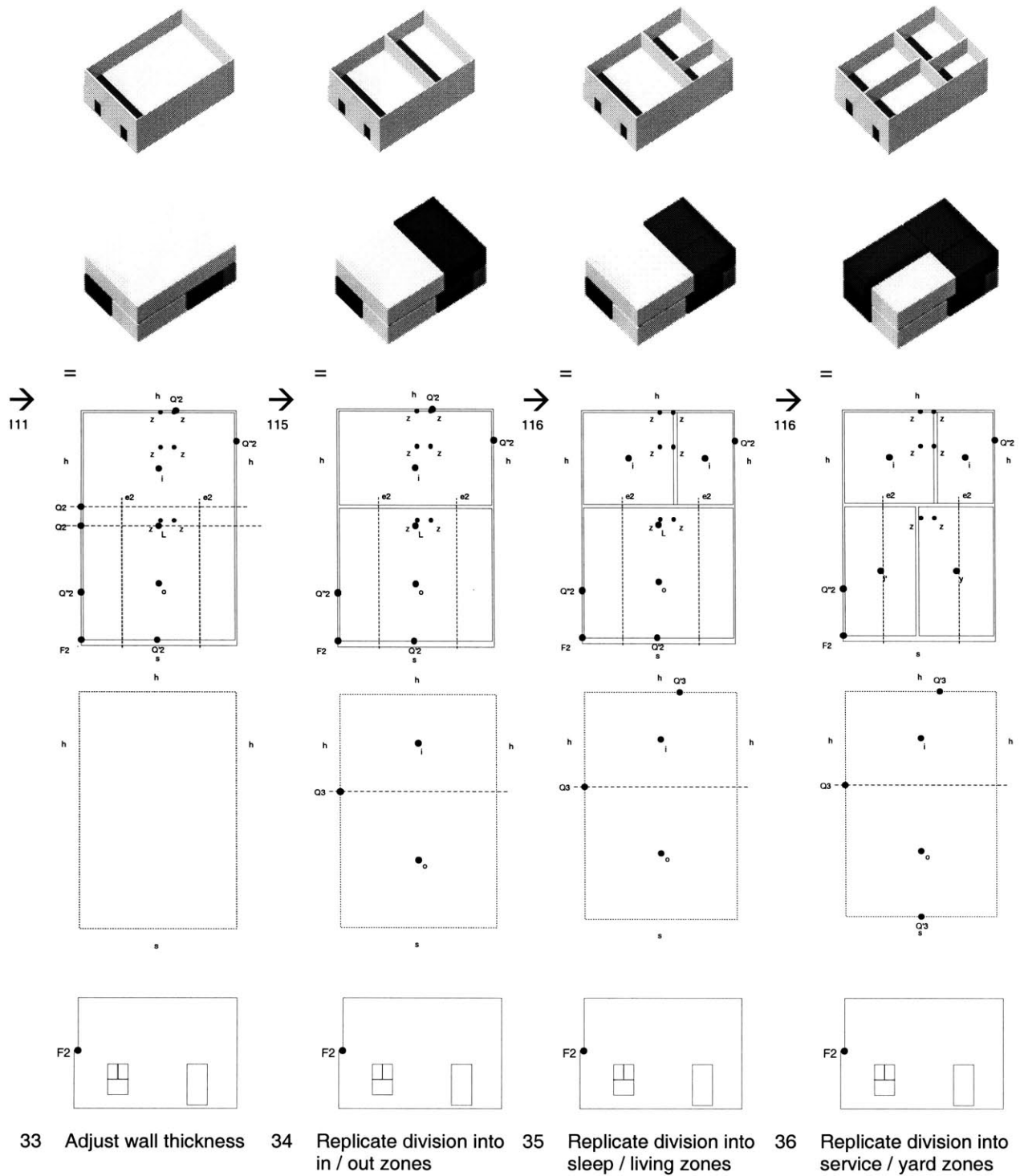


Figure 5.20 (continued) – Derivation of an existing design (Tabt5.)

2.3: Define circulation

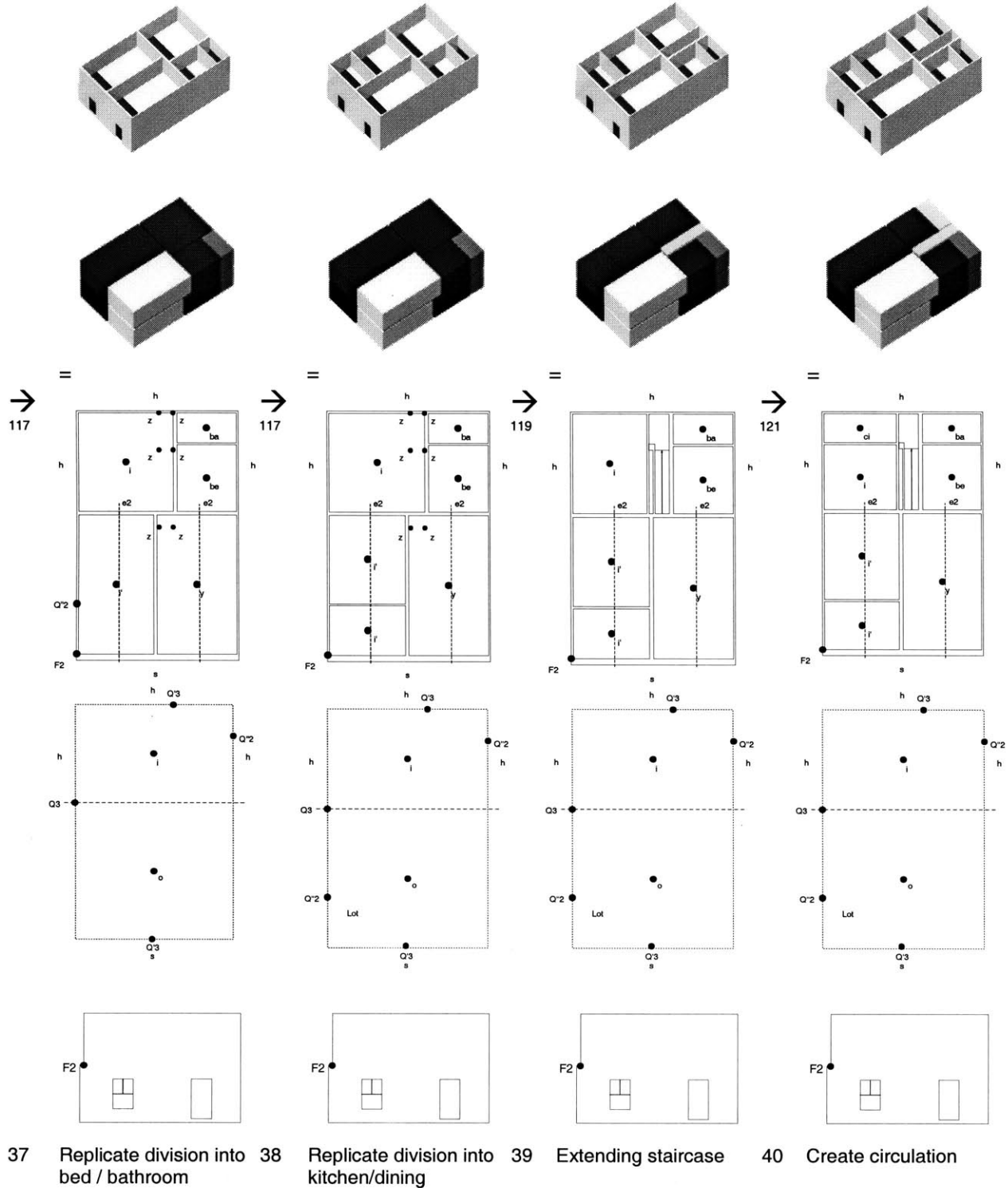


Figure 5.20 (continued) – Derivation of an existing design (Tabt5.)



## 2.4 Define rooms

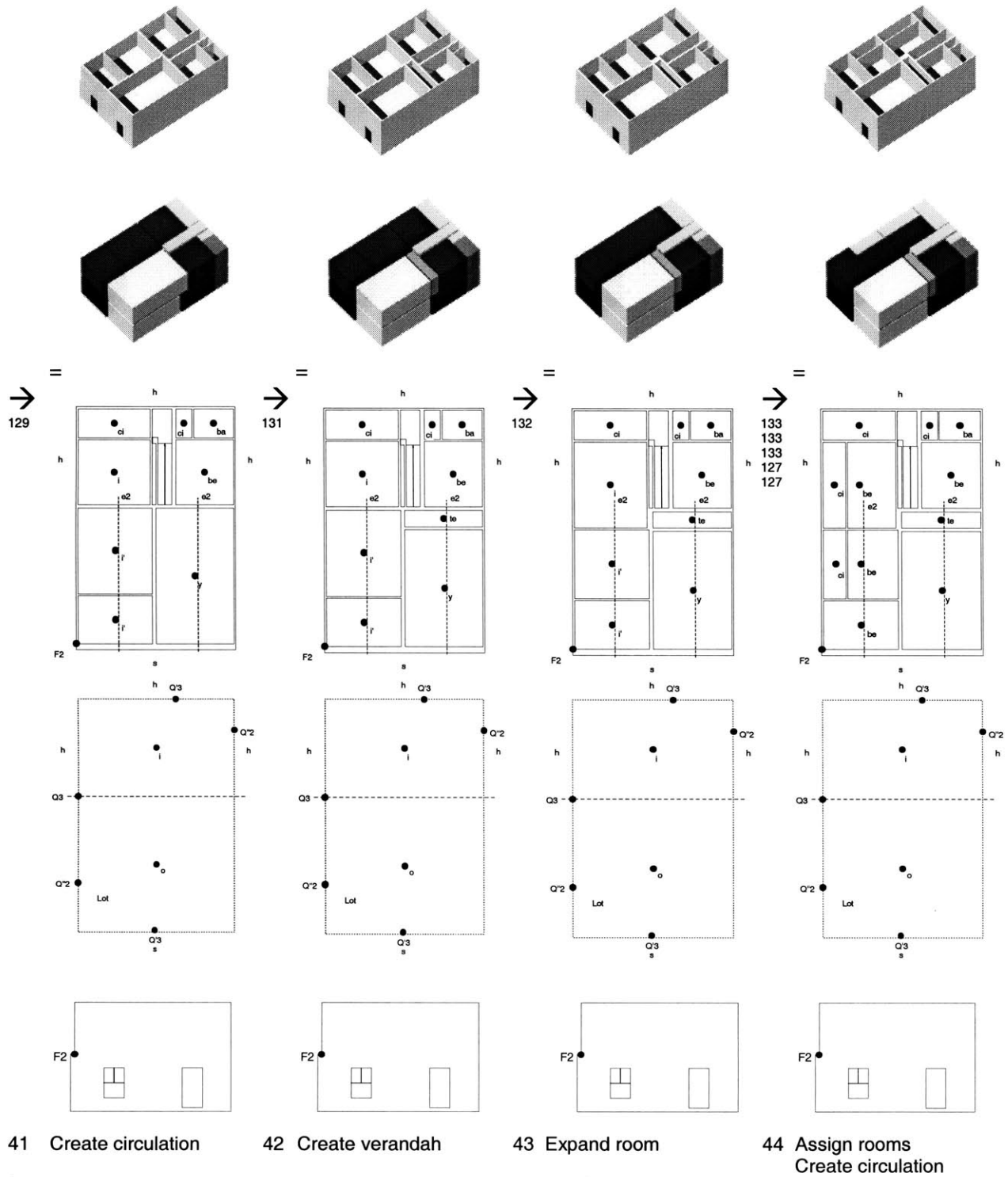


Figure 5.20 (continued) – Derivation of an existing design (Tabt5.)

2.5: Add details

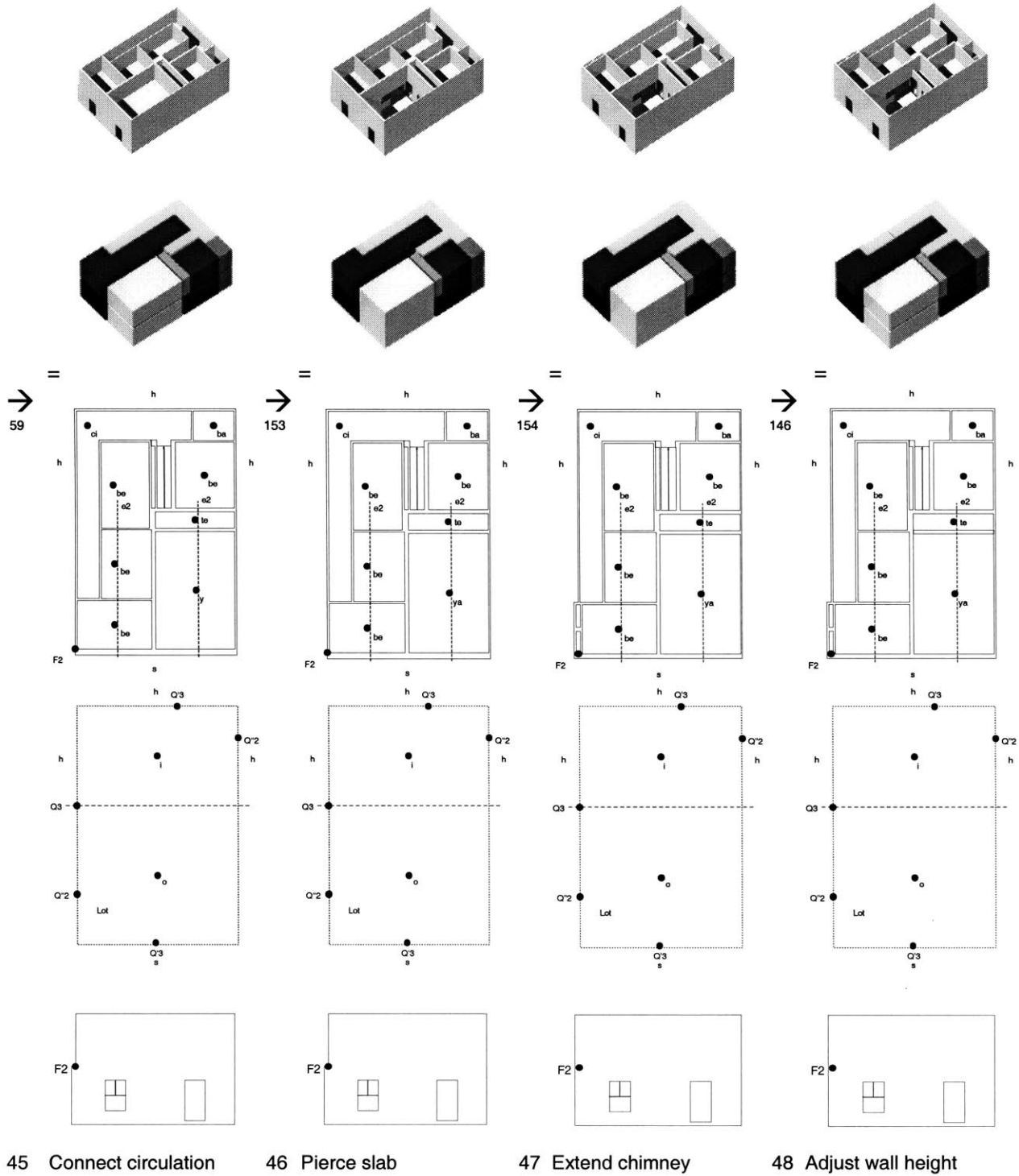


Figure 5.20 (continued) – Derivation of an existing design (Tabt5.)

2.6: Introduce openings

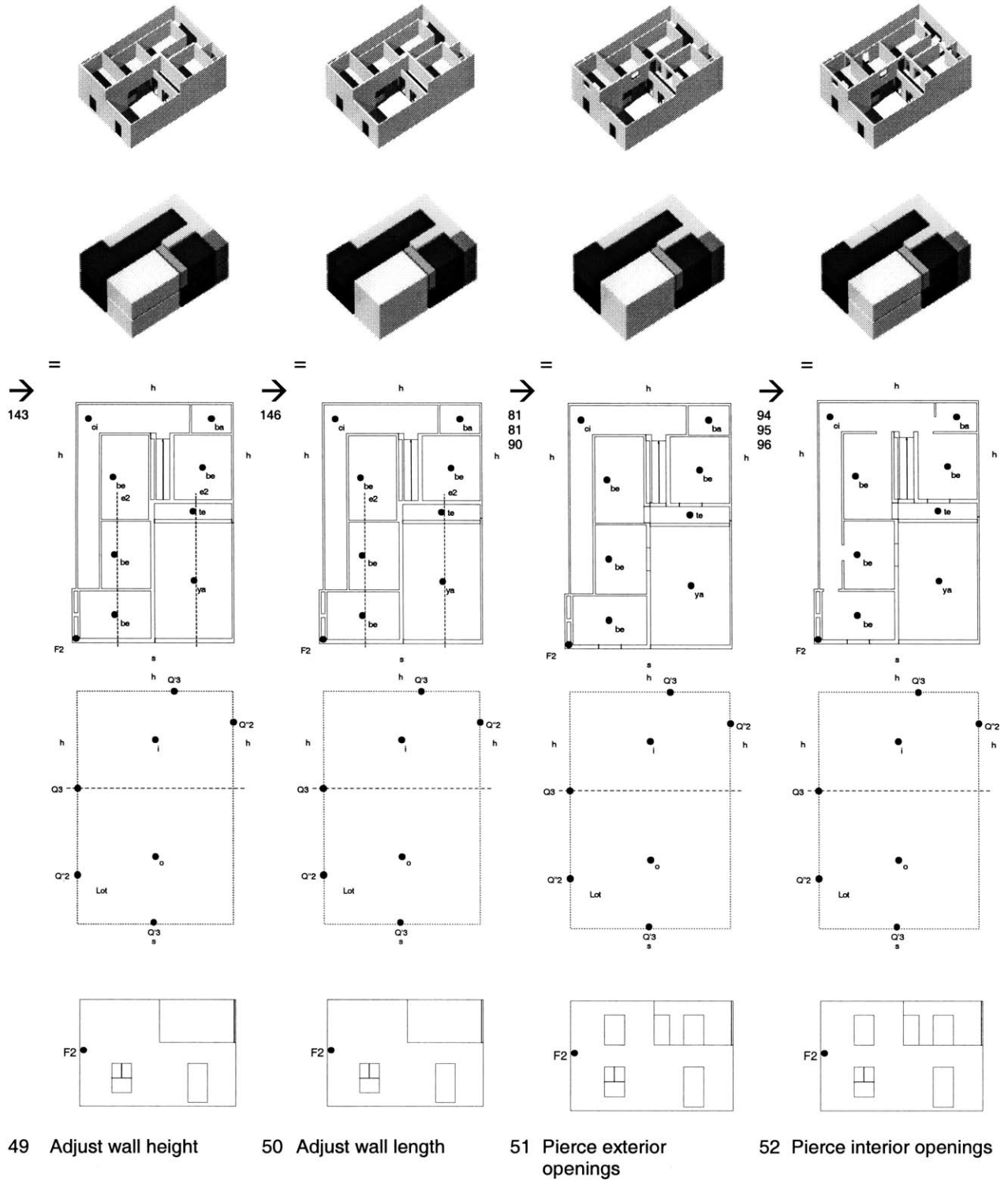


Figure 5.20 (continued) – Derivation of an existing design (Tabt5.)

2.7: Terminating

3: Define terrace  
3.1: Introduce slab

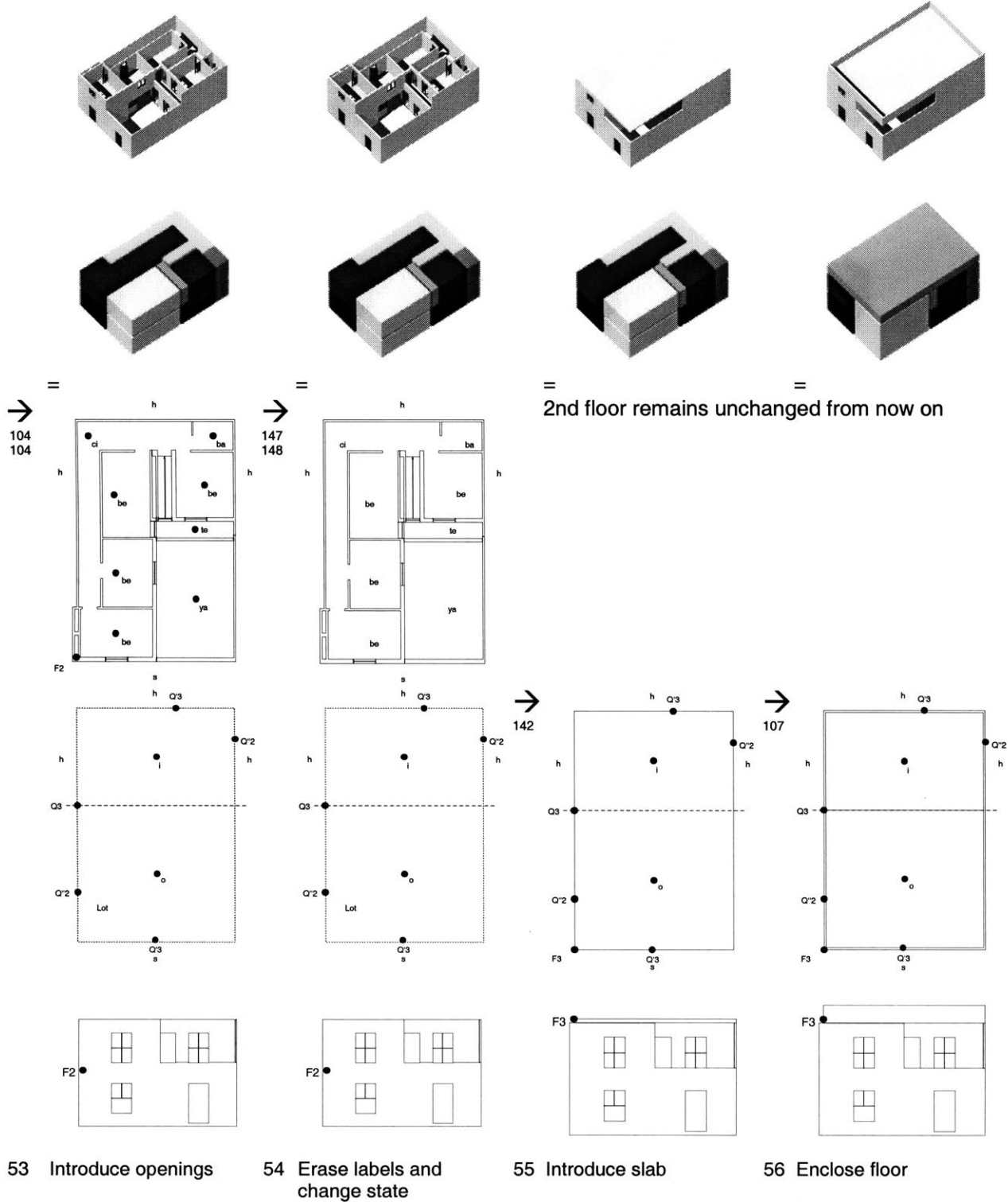


Figure 5.20 (continued) – Derivation of an existing design (Tabt5.)

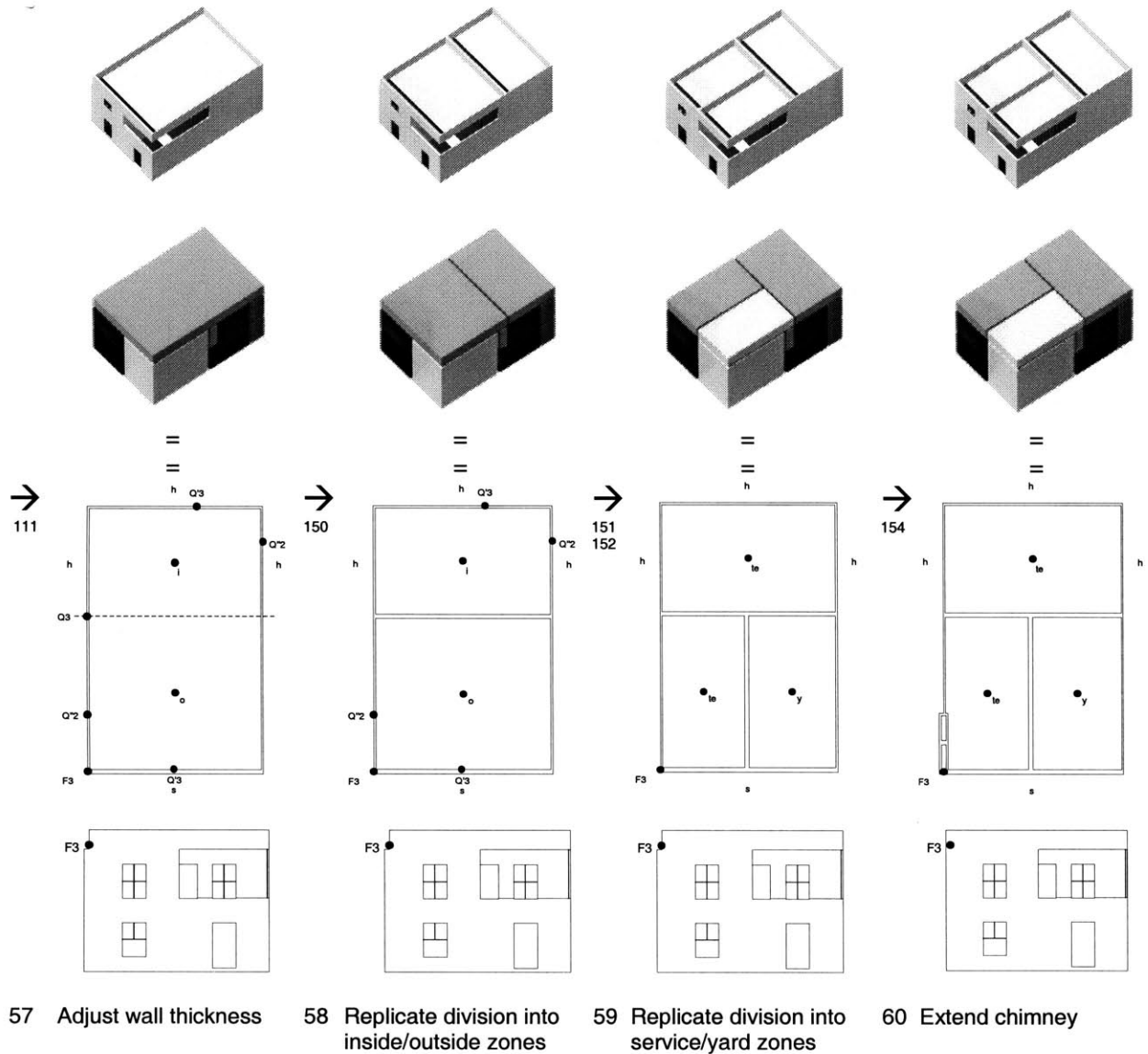


Figure 5.20 (continued) – Derivation of an existing design (Tabt5.)

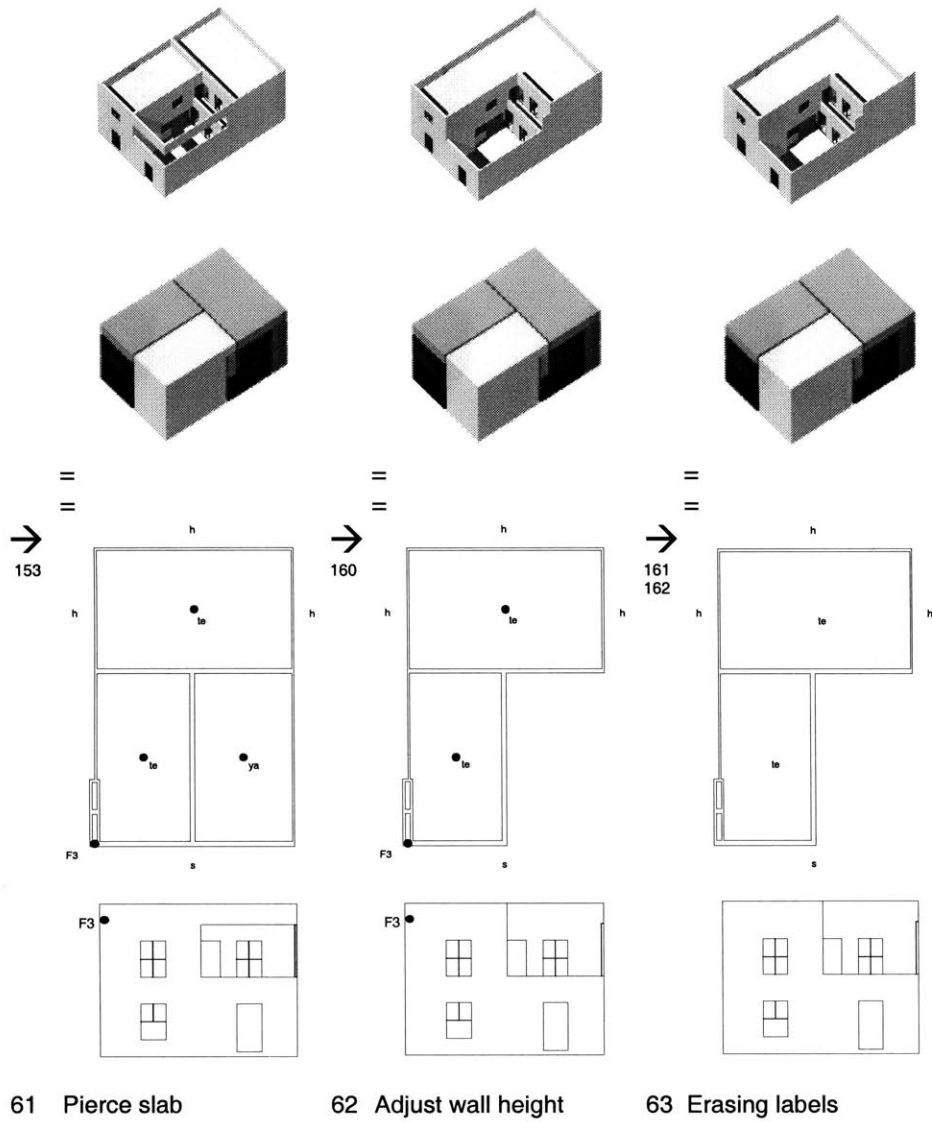


Figure 5.20 (continued) – Derivation of an existing design (Tabt5.)

## 5.11 Summary

The use of simple compositional rules consisting of the dissection of rectangles determines the style of the courtyard houses design by Álvaro Siza at Malagueira. These rules are then coupled with a set of constraints that specify functional requirements and limit the ways in which compositional rules can be applied. A shape grammar encoding both sets of rules is presented and discussed. The grammar accounts for the generation of the 35 houses considered in the corpus, thus fulfilling the requirements of the analytic test. The next chapters will address the generation of random new designs (synthetic test), and the generation of designs that match criteria (goal test.)

## References

MITCHELL W J, Steadman J P, Liggett R S, 1976, "Synthesis and optimisation of small rectangular floor plans" *Environment and Planning B* 3 37-70

TESTA, Peter "The Architecture of Álvaro Siza." S.M.Arch.S. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1984.





## **6. Experiments**

### **6.1 Introduction**

This chapter presents a set of design experiments undertaken with two goals in mind. The first goal was to generate new designs and to test the grammar by performing analytical, synthetic, and goal tests, as described in Chapter 2. In Experiment 1, the grammar rules presented in the previous chapter are applied to a new design not in the original corpus to verify their capability to account for its generation (analytic test.) Experiment 2 addressed the generation of a random new house (synthetic test.) In Experiments 3 and 4, experimental subjects derived designs for given clients out of the grammar rules (goal test.) The second goal was to find how designers used the grammar to generate such designs by undertaking a protocol study of Experiment 3. The results showed that, with minor changes, the grammar could successfully account for the generation of such designs, but that some changes were required to improve its capability to generate custom-tailored houses.

### **6.2 Experiment 1: existing design (analytic test)**

#### **6.2.1 Goal**

The goal of this experiment was to check whether the grammar could account for the generation of a patio house designed by Siza after the grammar had been developed.

### **6.2.2 Subjects**

Siza, the original author of the Malagueira houses, the author of the grammar, and one of the author's collaborators were the subjects of this experiment. The latter was an architect who became acquainted with Siza's work at Malagueira and the grammar while participating in the development of an interactive tool to teach the grammar. She had the main role in the experiment, and so she is referred to as the main subject in the discussion below.

### **6.2.3 Setting**

No particular setting was used in the experiment.

### **6.2.4 Task**

The experiment included two tasks. The first, assigned to Siza, was to design a new Malagueira house. This task was implicit because Siza did not design the new house on purpose for the experiment, which made it necessary to contextualize the results for their appropriate analysis. The second task, assigned to the main subject, was to use the grammar rules to reconstruct the generation of the new design, introducing new rules, only when necessary.

### **6.2.5 Procedure**

After becoming acquainted with grammar while participating in the development of the tool mentioned above, the main subject built the 2D and the 3D models of the new design to acquire a better understanding of the design. She then colored the layout, using tones of the same color for related rooms until decomposing the design into functional zones and then into the initial shape, following the procedure described in Figure 2.5. Next, she sketched the derivation of the design in 2D using paper and

pencil. After corrections had been made by the author of the grammar, she modeled the derivation in 3D using the computer.

### **6.2.6 Results**

The location of the new design is shown in Figure 6.1. The photos and plans are shown in Figure 6.2, and the 3D model is shown in Figure 6.3. The added rules are shown in Figure 6.4, and the derivation of the design is shown in Figure 6.5. To facilitate visual understanding, only the 3D part of the derivation is shown. Interested readers can reconstruct the 2D derivation by applying rules in Figures 5.19 and 6.4.

### **6.2.7 Discussion and conclusions**

The grammar can account for the most of the new design generation, despite the need to introduce new rules to derive some features that are not present in the previous Malagueira designs. These features are due to the non-standard land plot, and it can be shown that the new rules adjust the design scheme of the standard houses to this plot. The new design is, thus, a new housetype in the scheme, rather than a new scheme. It was, therefore, named Type E, following the conventions introduced in Chapter 5. The plots on which the three houses of this type were built had been initially destined for commerce, and later changed to housing, at the request of one of the cooperatives operating at Malagueira. This change in use explains the different lot shape. Unlike the standard 8 x 12 m rectangular plot, the new plots are trapezoids with the same width but longer.

The major differences of the new design relative to the previous designs are a big portico in the patio, at the front of the house, and a row of outdoor service rooms -- pantry, laundry, and gas cabinet -- adjacent to the non-orthogonal side of the plot, facing

the street. Despite these differences, a careful analysis of the design reveals that the layout is in the grammar. The dimensions of the house are exactly those that would result from one of the two possible dissections of the standard plots into inside and outside zones -- 6.0 m away from the back wall, as if the house had been placed against the back of the plot. This dissection permitted the inside zone to be manipulated as in a standard design. The resulting deeper outside zone, however, needed to be manipulated differently, leading to the design of the portico and the outdoor service rooms. The portico helps to protect the houses from the bright south sunlight. The outdoor service rooms are allowed by the R.T.H.S., the Portuguese regulations for social housing, although they are indoors in all the previous houses.

As a way of illustrating how the grammar can be enlarged to generate Type F, part of the rules that need to be added to the grammar are shown in Figure 6.4. The shown rules apply to the generation of the first floor functional organization. The addition of the remaining rules is left to the reader. Rules N0 through N6 are variations of rules R1-R6, adapted to the trapezoidal lot shape. Rule N1 introduces the initial shape representing the lot. Rule N2 introduces the floor slab. Rule N2 encloses the floor with walls. Rules N3 and N4 adjust the thickness of the enclosing walls when they are adjacent to the street. Rules 5 and 6 divide the lot into inside and outside zones. Rules 7 through N13 apply to generate the portico and the storage spaces. Rule N7 creates a gallery by dissecting the patio with a line perpendicular to its orthogonal sides, when the house faces south and the patio is longer than 7.0 m. Rule N8 creates a service area next to the street by dissecting the patio with a line parallel to its diagonal side, when the patio is longer than 6.0 m. This rule creates a parallelogram, a shape that Siza avoided in the previous designs because it lacked orthogonal sides. In fact, Siza had no choice. Had he used a rule like Rule 47 to dissect the patio, he would have ended up with storage

rooms bigger than permitted by the R.T.H.S. Rule N9 creates an entrance hall on the axis of symmetry of the street facade, by dissecting the outdoor service area with two lines perpendicular to the street. Notice how Rule N9, in fact, prevents further development of non-orthogonal shapes. Had Siza placed the portico by dissecting the outdoor service area with lines parallel to the bigger sides of the plot, we would have ended up with parallelogram shapes and rooms that had no orthogonal sides, which he did not consider acceptable. By using lines perpendicular to the street he got rid of the parallelogram created by rule N8, and created acceptable, trapezoidal shapes instead. This rule is, thus, consistent with the previous set of rules. Rule N10 opens the gallery to the yard, and places a column to support its covering. Rules N11-13 assigns functions to rooms with trapezoidal shapes. Some of the new rules are created just by changing the conditions under which previous rules apply. For instance, Rule N7 is similar to Rule 47, except in the conditionals on function. Some other rules are entirely new shape rules. For instance, Rules N8 introduces a new shape in the vocabulary, as mentioned above.

The similar width of the plot and the fact that it resulted from a change in land use, might partially explain why Siza's strategy was to adjust the standard design scheme to the new plot, rather than to devise a new scheme, as he had done in other non-standard plots. Nevertheless, our conversations with Siza also might have influenced his strategy. Siza designed the new type after we had shown him the grammar and the generation of a new type following the rules (see Section 6.3.) In this conversation, we stressed how the staircase seemed to be key in the definition of the Malagueira types, and how we had placed an I-shaped staircase in the living room to define the new backyard type (the existing backyard type -- Type B -- had an U-shaped staircase.) Our conversation might have increased his desire to devise a new housetype. As there were

no more standard plots available for housing, the modified plots represented an opportunity for experimentation. Moreover, in the new type he placed an U-shaped staircase in the service zone, which never happened in the previous frontyard types. Therefore, the new design supports our previous assumptions regarding the placement of staircases and their role in the definition of types, as shown by comparing the derivation tree in Figure 6.6, which includes the new types, with the one in Figure 5.9, which presents the previous types.



Figure 6.1 – The location of type F houses on plots initially destined for commerce.

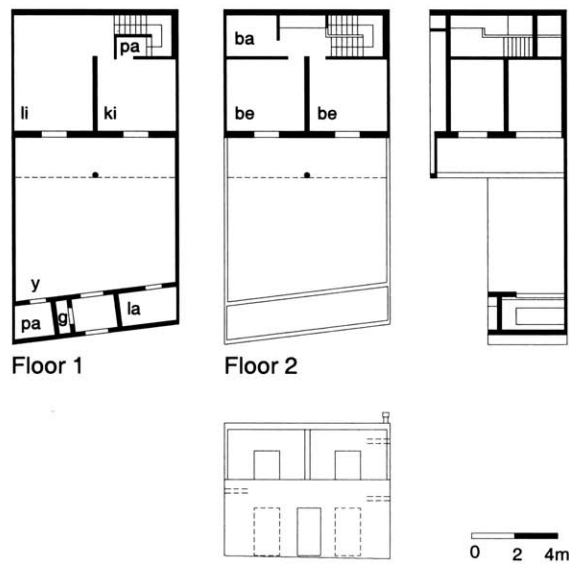


Figure 6.2 – Plans and photos of type F, the new design designed by Siza (1999) after the grammar had been developed.

# Housetype F - 1999

Malagueira - Alvaro Siza Vieira

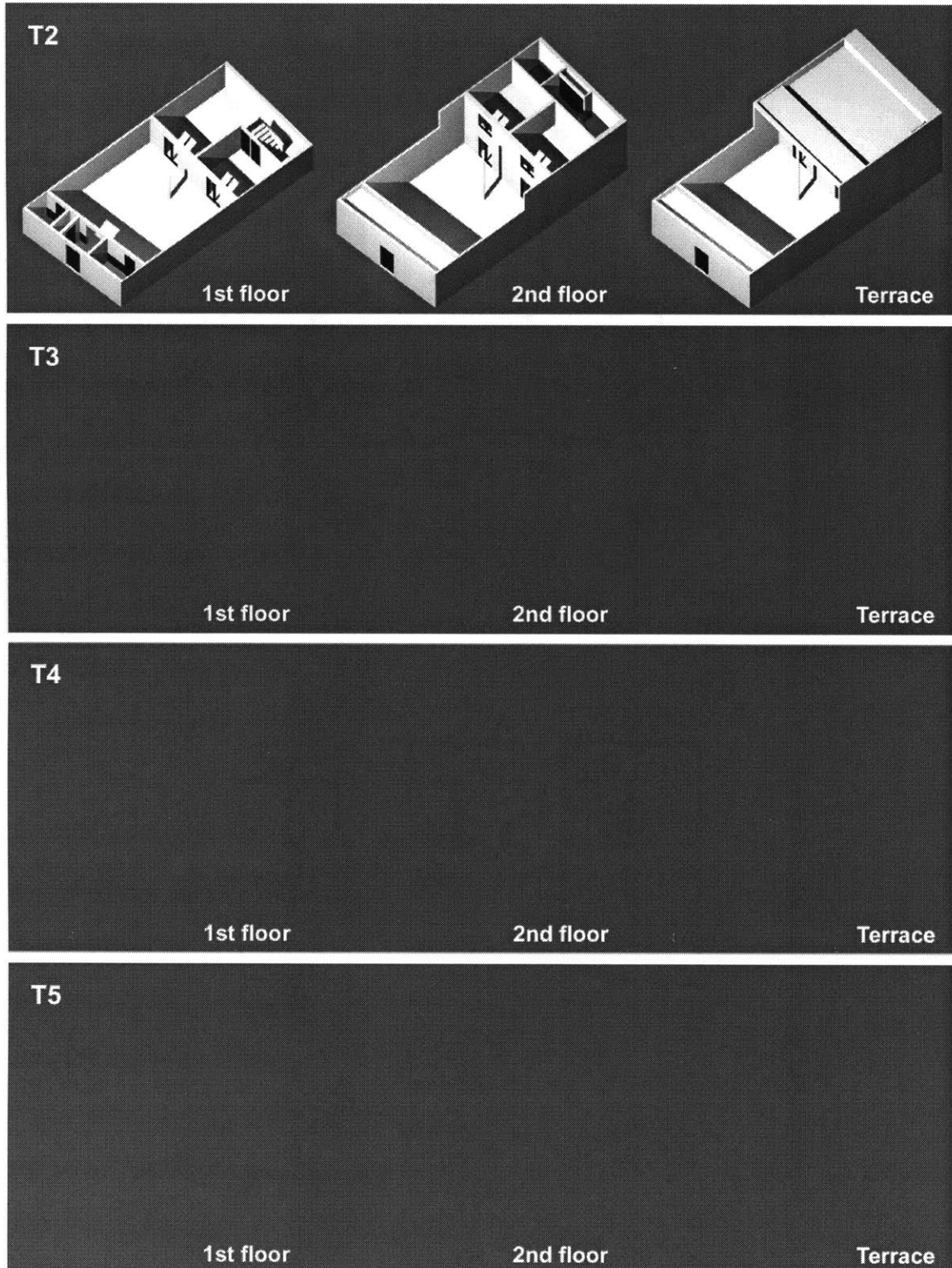
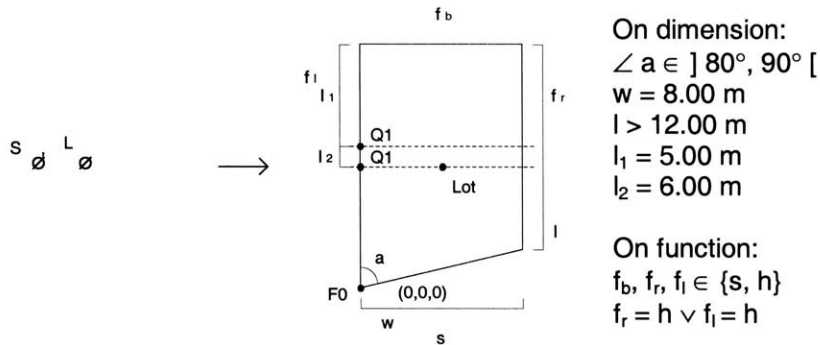


Figure 6.3 – Digital model of Type F, the new design designed by Siza after the grammar had been developed.



**Stage 0: Introduce initial shape and grammar labels**

Rule N0: Introduce initial shape



**Stage 1: Define first floor**

**Step 1.1: Start**

Rule N1: Introduce slab

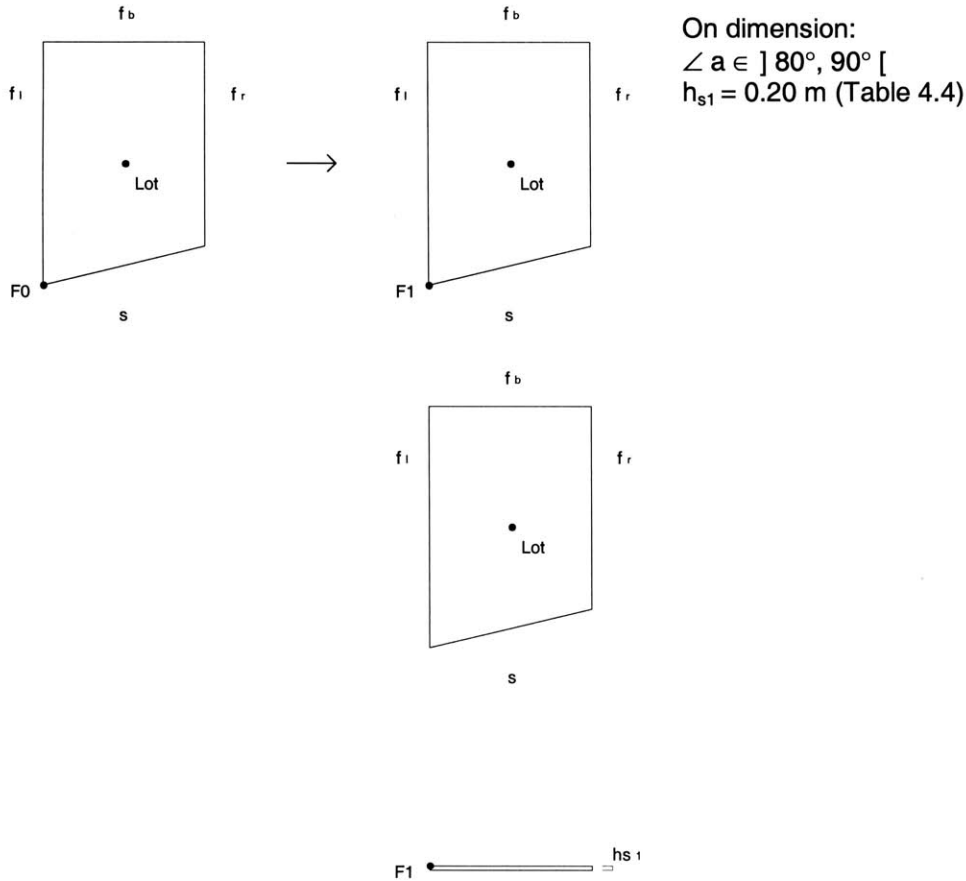
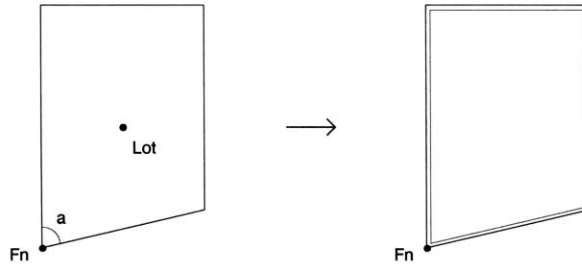


Figure 6.4 – The set of rules introduced with the design of Type F, the new design designed by Siza after the grammar had been developed. The new rules are mainly used to adjust the standard design scheme encoded by the grammar to the non-standard land plot.

**Rule N2: Introduce enclosing walls**

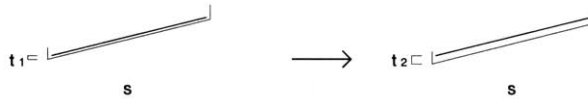


On dimension:  
 $\angle a \in ] 80^\circ, 90^\circ [$   
 $h_{\min} < h_{f1} < h_{\max}$  (Table 4.4)  
 $t = 0.1 \text{ m}$

On function:  
 $n \in \{1, 2\}$



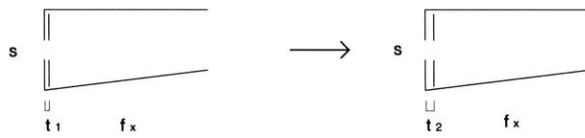
**Rule N3: Adjust front and back wall thickness**



On dimension:  
 $t_1 = 0.10 \text{ m} \Rightarrow t_2 = 0.20 \text{ m}$

On function  
 $p(s) \in \{f_f, f_b\}$

**Rule N4: Adjust left and right wall thickness**



On dimension:  
 $t_1 = 0.10 \text{ m} \Rightarrow t_2 = 0.20 \text{ m}$

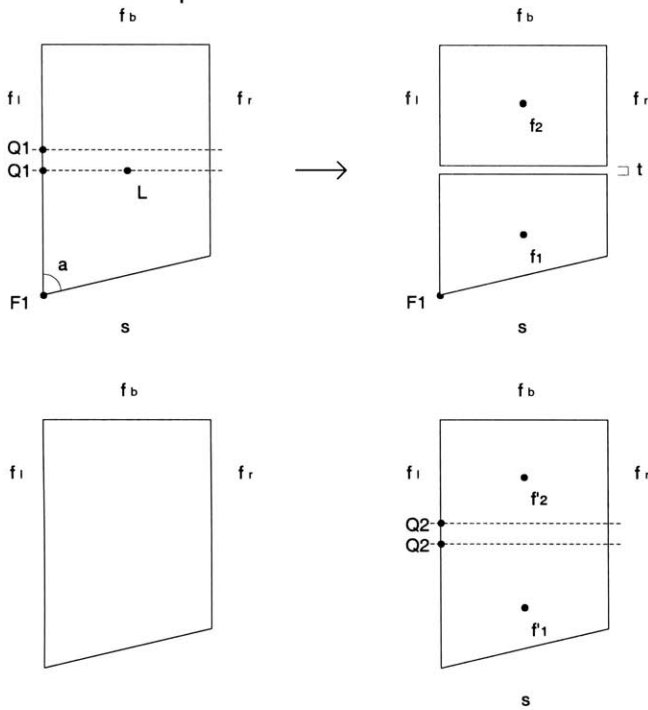
On function:  
 $p(f_x) \in \{f_f, f_b\}$



Figure 6.4 (continued) – Part of the set of rules introduced with the design of Type F.

Step 1.2: Locate functional zones

Rule N5: Locate patio

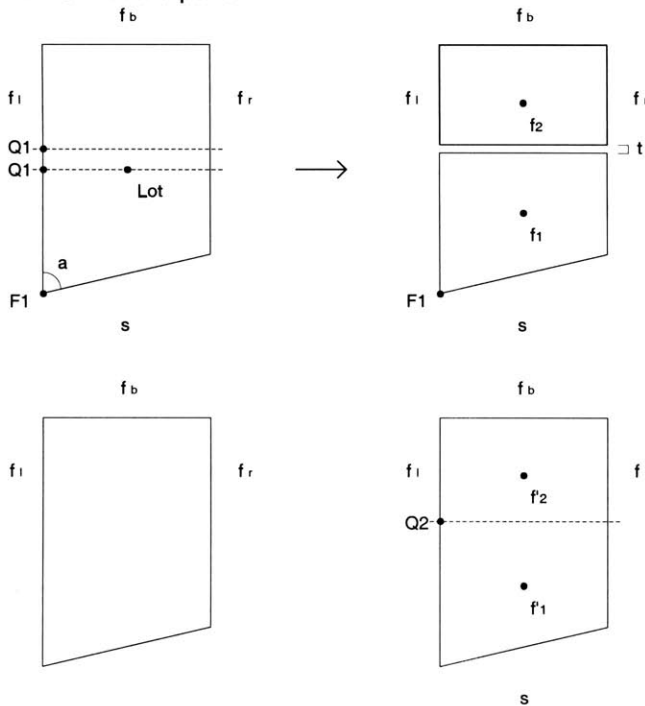


On dimension:  
 $t = 0.20 \text{ m}$

On function:  
 $f_1, f_2 \in \{i, o\} \wedge f_1 \neq f_2$   
 $f_1 = f'_1, f_2 = f'_2$

R4:  $\langle f_1, f_t, f_r, s, f_i, L: f_1, f_2 \rangle$

Rule N6: Locate patio



On dimension:  
 $t = 0.20 \text{ m}$

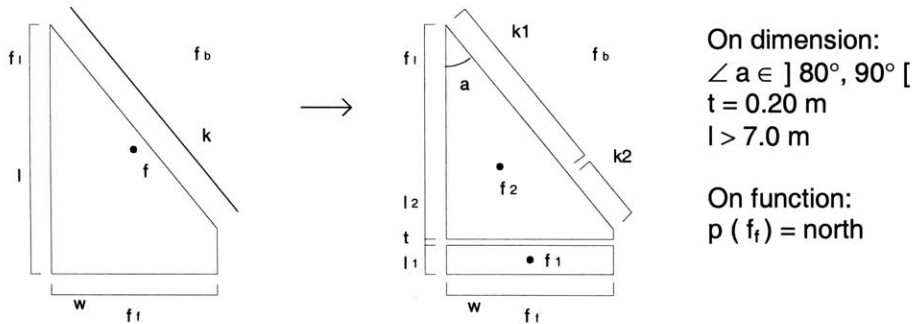
On function:  
 $f_1, f_2 \in \{i, o\} \wedge f_1 \neq f_2$   
 $f_1 = f'_1, f_2 = f'_2$

R4:  $\langle f_1, f_t, f_r, s, f_i, L: f_1, f_2 \rangle$

Figure 6.4 (continued) – Part of the set of rules introduced with the design of Type F.

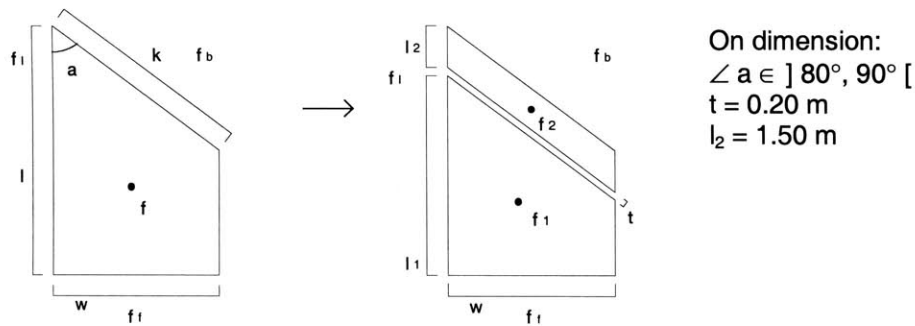
**Step 1.4: Divide functional zones into rooms**

**Rule N7: Dividing the yard to create a gallery**



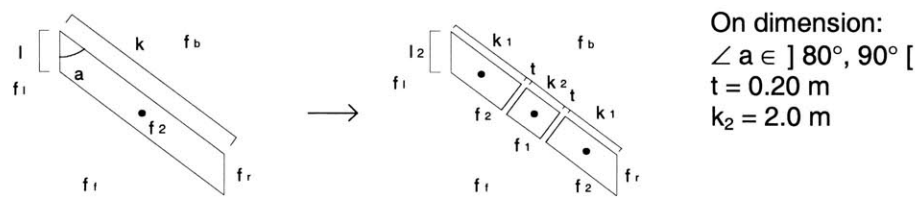
RN7: < F1, y: y, ga >

**Rule N8: Dividing the yard to create an outdoor service space**



RN8: < F1, y: y, so >

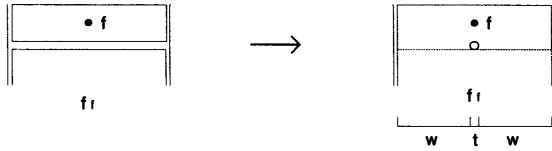
**Rule N9: Dividing the yard to create an outdoor service space**



RN9: < F1, so: ha, so, so >

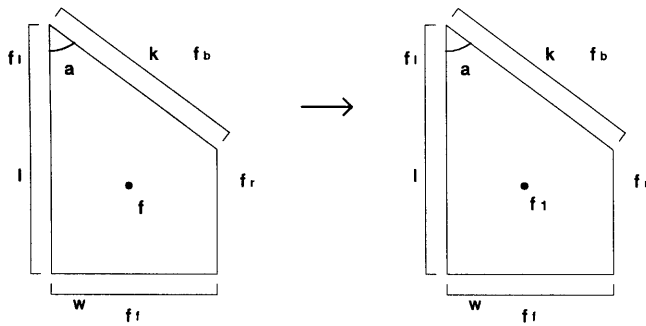
Figure 6.4 – Part of the set of rules introduced with the design of Type F.

Rule N10: Open the gallery to the patio



RN10: < F1, y: y, ga >

Rule N11-13: Assign functions to outdoor service rooms



On dimension:  
 $\angle a \in ] 80^\circ, 90^\circ [$   
 $l_{\min} \leq l \leq l_{\max}$   
 $w_{\min} \leq w \leq w_{\max}$   
 $a_{\min} \leq a \leq a_{\max}$

RN11: < F1, so: la >

RN12: < F1, so: gs >

RN13: < F1, so: pa >

Figure 6.4 (continued) – Part of the set of rules introduced with the design of Type F.

## Derivation of TF

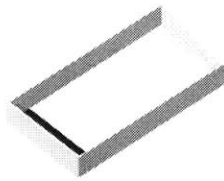
## Malagueira - Alvaro Siza Vieira

Stage 1: Define 1<sup>st</sup> floor

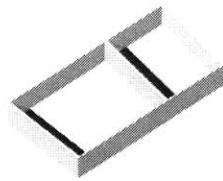
Step 1.1: Start floor



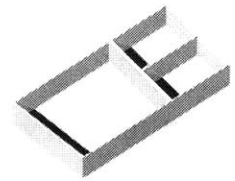
1. Introduce initial shape



2. Introduce initial shape



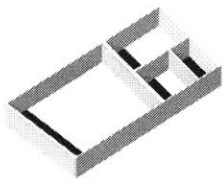
3. Locate in-/outside zones



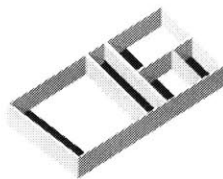
4. Locate living/services zones

Step 1.3: Define circul.

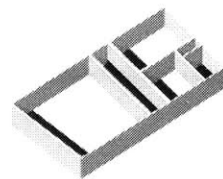
Step 1.4: Divide zones into rooms



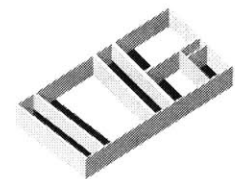
5. Locate staircase



6. Locate gallery



7. Create circulation



8. Locate yard/services

Figure 6.5 – Derivation of Type F, the new design designed by Siza after the grammar had been developed.

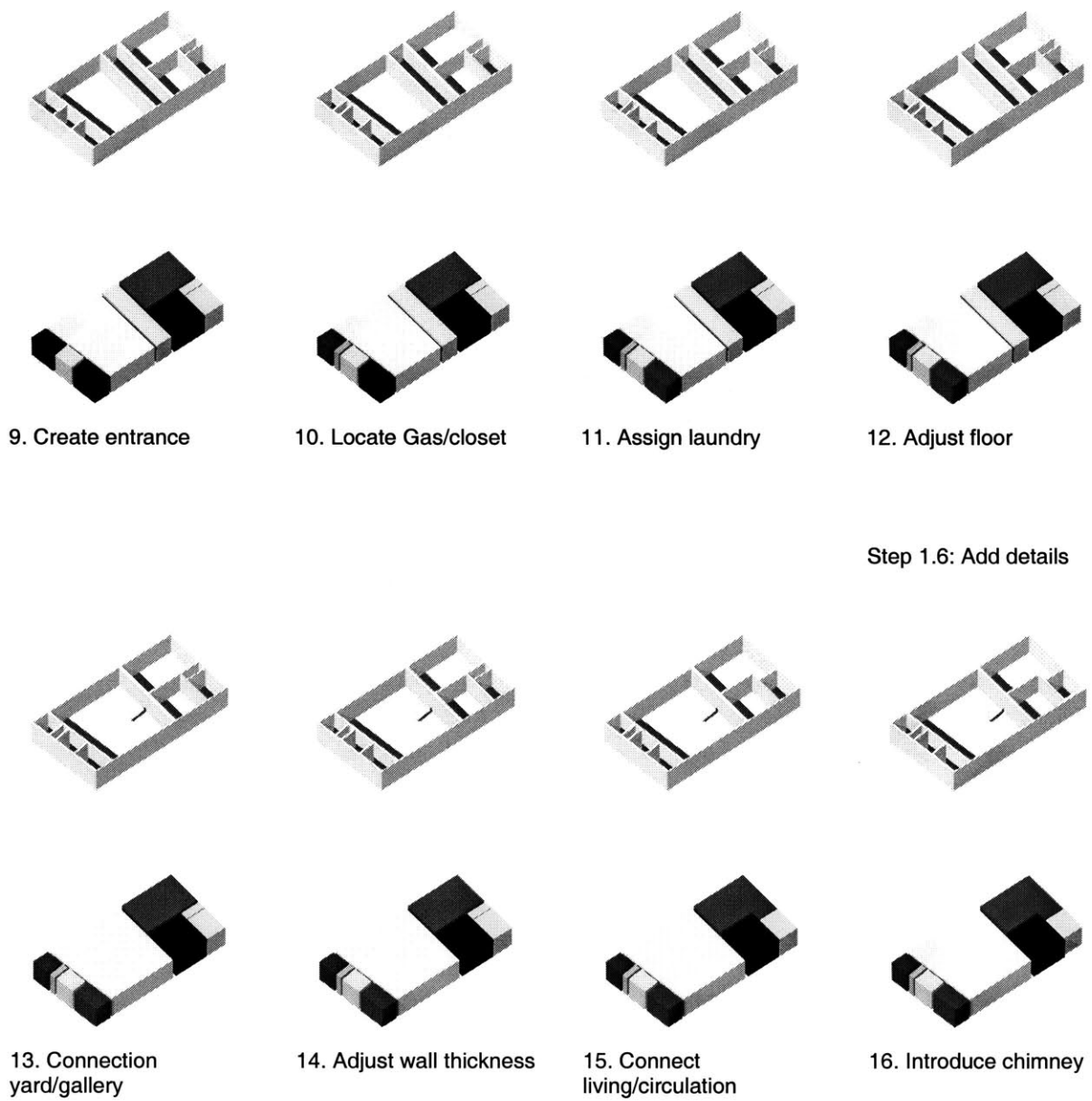
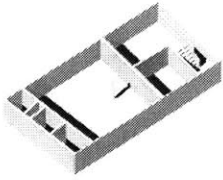
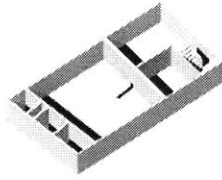


Figure 6.5 (continued) – Derivation of Type F, the new design designed by Siza after the grammar had been developed.

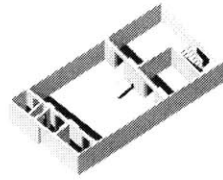
1.7: Create Openings



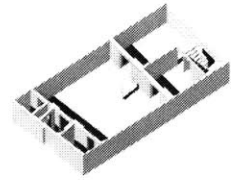
17. Detailing stairs



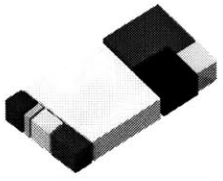
18. Assign pantry



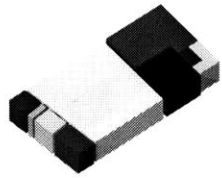
19. Pierce external openings



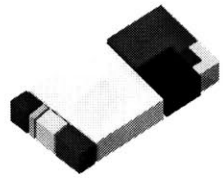
20. Pierce internal openings



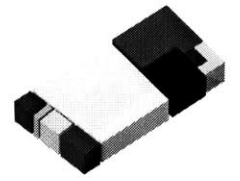
17. Detailing stairs



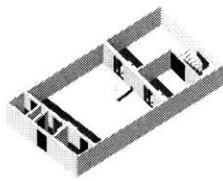
18. Assign pantry



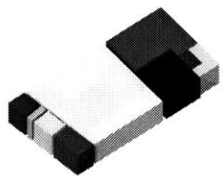
19. Pierce external openings



20. Pierce internal openings



21. Introduce openings



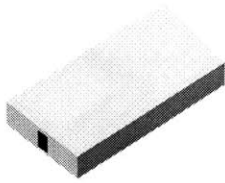
21. Introduce openings

Figure 6.5 (continued) – Derivation of Type F, the new design designed by Siza after the grammar had been developed.

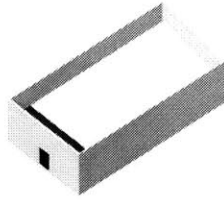


Stage 2: Define 2<sup>nd</sup> floor

Step 2.1: Start floor

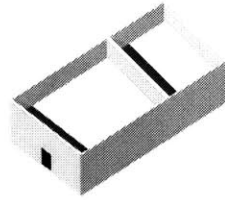


22. Introduce slab

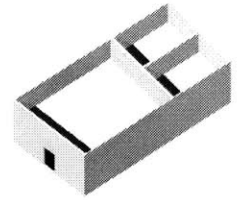


23. Enclosing floor

Step 2.2: Replicate 1st floor divisions



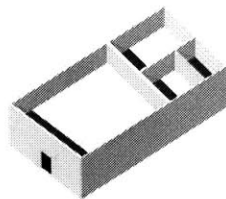
24. Replicate division into in-/outside zones



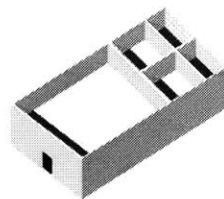
25. Replicate division into living/services zones

Step 2.3: Define circul.

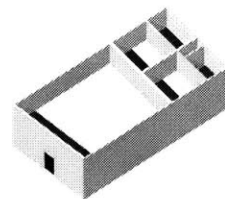
2.7: Define rooms



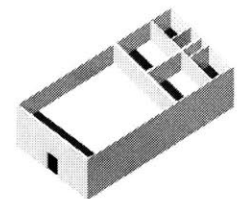
26. Extend staircase



27. Locate bathroom

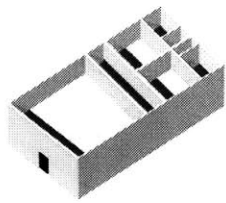


28. Pierce slab

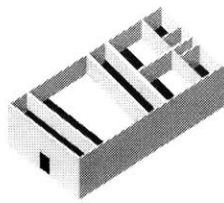


29. Create circulation

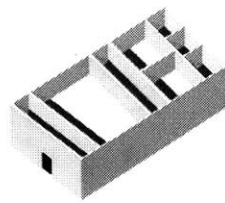
Figure 6.5 (continued) – Derivation of Type F, the new design designed by Siza after the grammar had been developed.



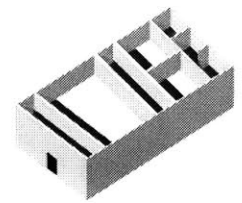
30. Replicate division into yard/gallery



31. Replicate division into yard/services

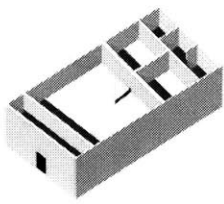


32. Connect circulation

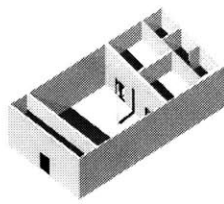


33. Connect circulation

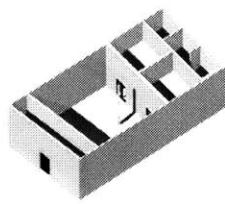
Step 2.8: Add details



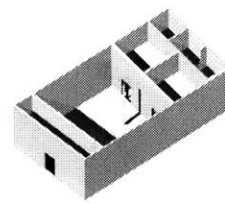
34. Adjust wall thickness



35. Pierce slab



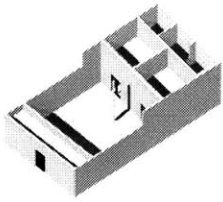
36. Extend chimney



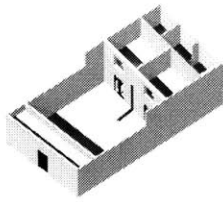
37. Adjust wall height interior

Figure 6.5 (continued) – Derivation of Type F, the new design designed by Siza after the grammar had been developed.

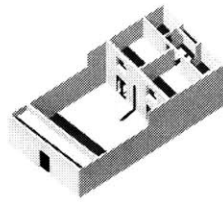
Step 2.9: Create Openings



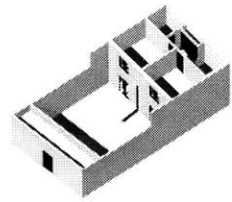
38. Adjust wall height/thickness exterior



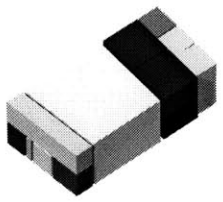
39. Pierce external openings



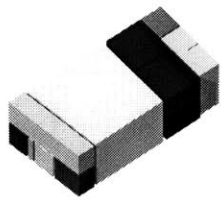
40. Pierce internal openings



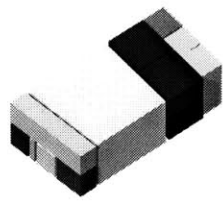
41. Introduce openings



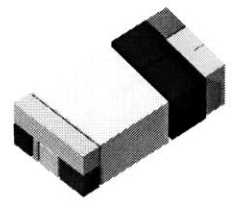
38. Adjust wall height/thickness exterior



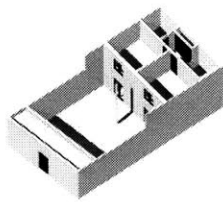
39. Pierce external openings



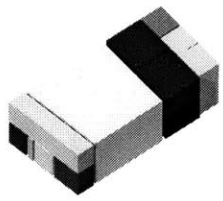
40. Pierce internal openings



41. Introduce openings



42. "Stretch"

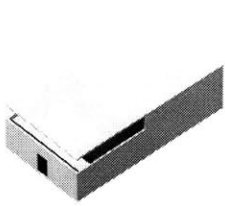


42. "Stretch"

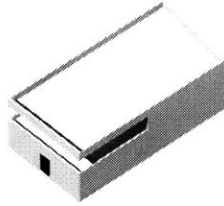
Figure 6.5 (continued) – Derivation of Type F, the new design designed by Siza after the grammar had been developed.

Stage 3: Define terrace

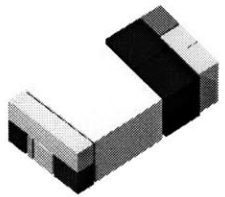
Step 3.1: Start floor



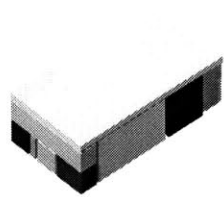
43. Introduce slab



44. Enclosing floor

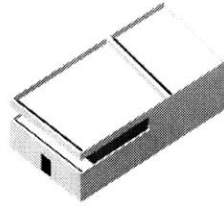


43. Introduce slab

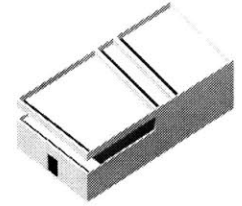


44. Enclosing floor

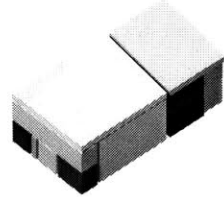
Step 3.2: Replicate 1<sup>st</sup> floor divisions



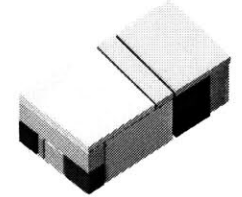
45. Replicate division into in-/outside zones



46. Replicate division into yard/gallery

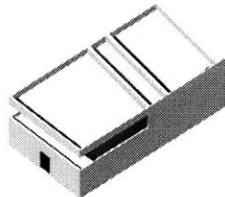


45. Replicate division into in-/outside zones

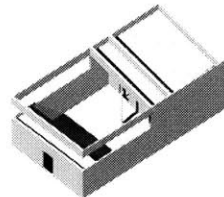


46. Replicate division into yard/gallery

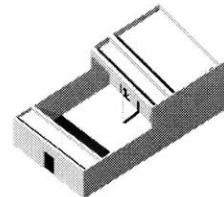
Step 3.3: Add details



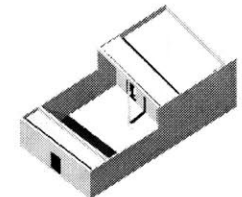
47. Extend chimney



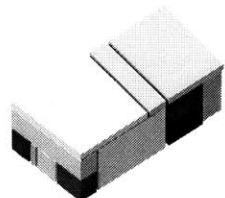
48. Pierce slab



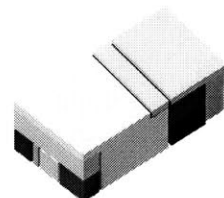
49. Adjust wall height



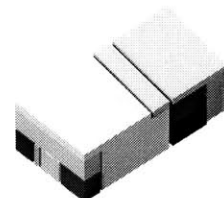
50. "Stretch"



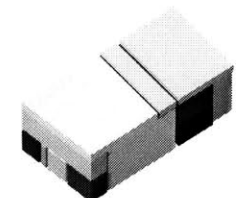
47. Extend chimney



48. Pierce slab



49. Adjust wall height



50. "Stretch"

Figure 6.5 (continued) – Derivation of Type F, the new design designed by Siza after the grammar had been developed.

DEFINE 1st FLOOR

LOCATE FUNCTIONAL ZONES

Locate outside /inside zones

Locate passage

Locate living / sleeping zones

Locate remaining zones  
*Basic pattern defined*

LOCATE STAIRCASE  
*Type defined*

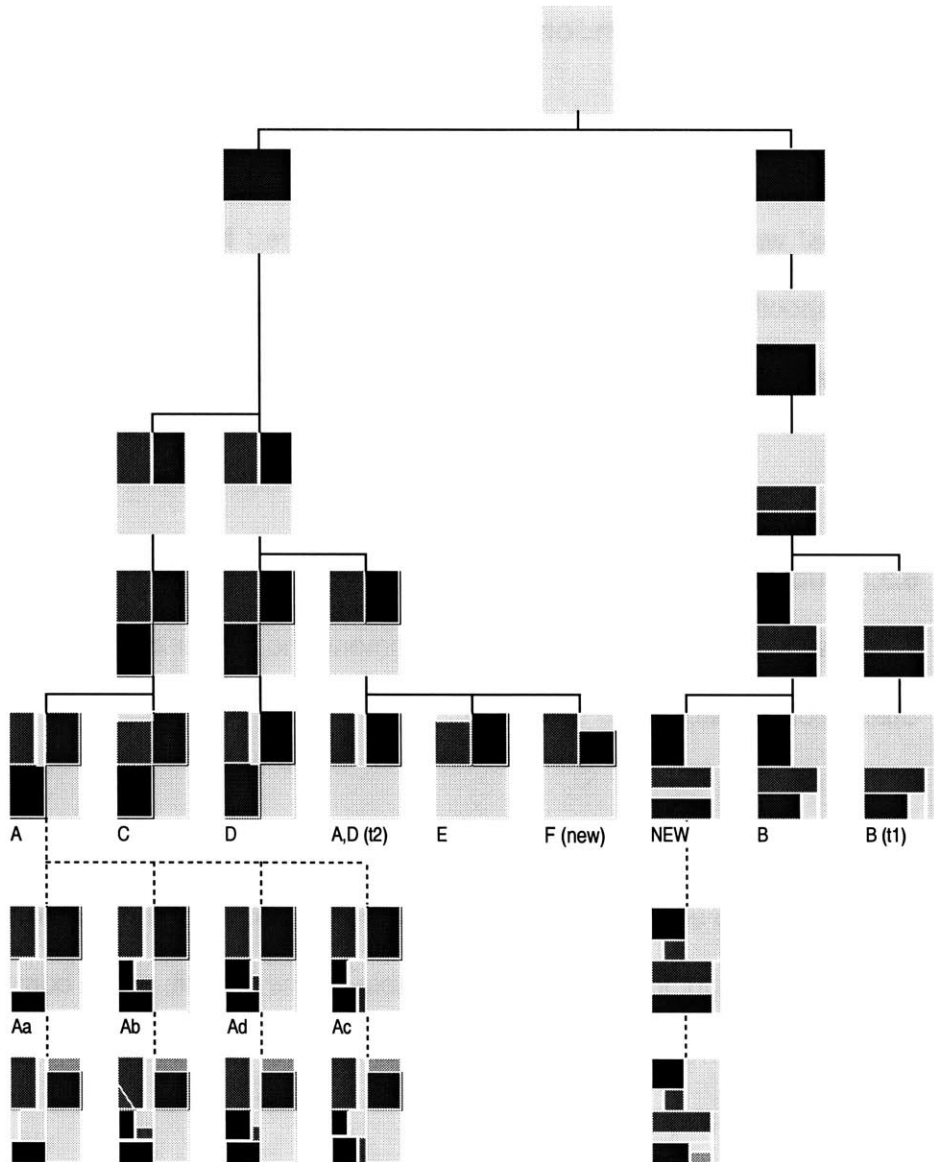


Figure 6.6 – Partial tree diagram showing the derivation of the new types presented in Sections 6.2 and 6.3.

## **6.3 Experiment 2: random design (synthetic test)**

### **6.3.1 Goal**

The goal was to verify the third criterion required for defining a successful grammar, which specifies that "it should provide the compositional machinery needed to design new buildings that are instances of the style" (Stiny and Mitchell, 1978) – the synthetic test.

### **6.3.2 Subjects**

Three subjects participated in the experiment: the author of the grammar, Siza, and Siza's collaborator Nuno Lopes.

### **6.3.3 Setting**

There were two settings. The first, used by the author of the grammar in the generation of the new design, included a chair, a table, paper, pencil, and a computer. The second, used to show the new design to Siza and Lopes, included two chairs, a table, and a computer to display the results.

### **6.3.4 Task**

The experiment included two tasks. The first, assigned to the author of the grammar, was to generate a new design out of the Malagueira grammar rules with no particular program. The second, assigned to Siza and Nuno Lopes, was to verify whether the new design was in the grammar.

### **6.3.5 Procedure**

The procedure followed in this experiment was as follows. First, the experimental subject sketched the derivation of the design in 2D using pencil to apply the rules by hand on a paper with a drawn rectangle representing the 8 x 12 m Malagueira lot. This rectangle had overlaid a metric grid to facilitate rule application and give a sense of scale. Then, he modeled the derivation and the design in 3D using the computer. Next, 2D and 3D digital and physical models of the new design – a backyard type -- were placed amidst original Malagueira designs and shown to Siza and Nuno Lopes separately. The only backyard type designed by Siza was omitted to prevent him from noticing that there was an additional backyard type.

### **6.3.6 Results**

The plans and views of the new design are shown in Figure 6.7, the 3D model is shown in Figure 6.8, and the derivation is shown in Figure 6.9. As in the previous section, the derivation is shown only in 3D.

### **6.3.7 Discussion and conclusion**

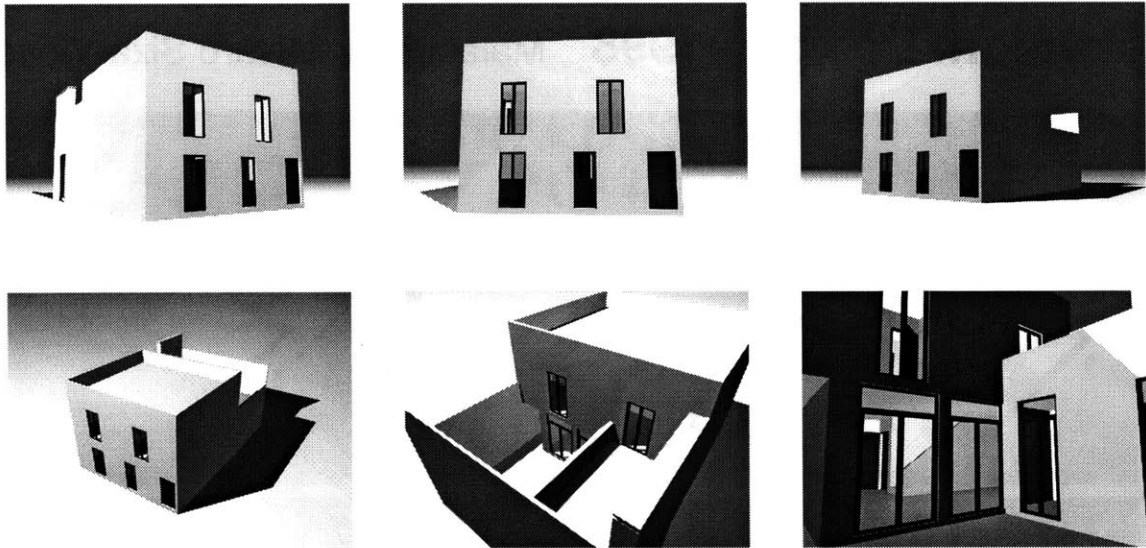
The new design is a backyard type. The decision to design a backyard type emerged after a conversation with Siza during which he expressed regret for having designed only one backyard type due to the lack of demand. Designing a backyard type represented a challenge because it was easier to spot stylistic mistakes, as there was only such type. On the other hand, if the experiment succeeded it would prove the validity of the grammar with fewer doubts.

The analysis of the corpus to infer the grammar showed that the differences among designs were due to different rule applications. It also showed that the key-difference

among housetypes was the placement of the staircase. Therefore, the new backyard type was defined first by a different placement of the staircase, (see Figure 6.6) and then by the using the existing set of rules to derive the remaining of the design. The backyard type designed by Siza had an U-shaped staircase placed in the sleeping zone. In the new type an I-shaped staircase was placed in the living zone.

When the new design was shown to Siza amidst other Malagueira designs, he did not notice that it was not his own design. At some point, he seemed confused because he did not remember such a placement of the staircase. But then, he acknowledged its validity and dismissed his doubts, and validated the design. When he was told that it was a his design he was truly surprised. Then, after careful analysis, he acknowledged its validity again. A similar experiment was undertaken with Nuno Lopes with the same result. The results of these experiments showed that the grammar did capture Siza's design rules at Malagueira and cleared the way for the undertaking of more ambitious experiments, explained, in the next sections.





New Design



Figure 6.7 – Plans and views of the new Malagueira design designed by the author of the grammar out of its rules, and which Siza considered stylistically correct.

# Random Design - 1998 Malagueira - Alvaro Siza Vieira

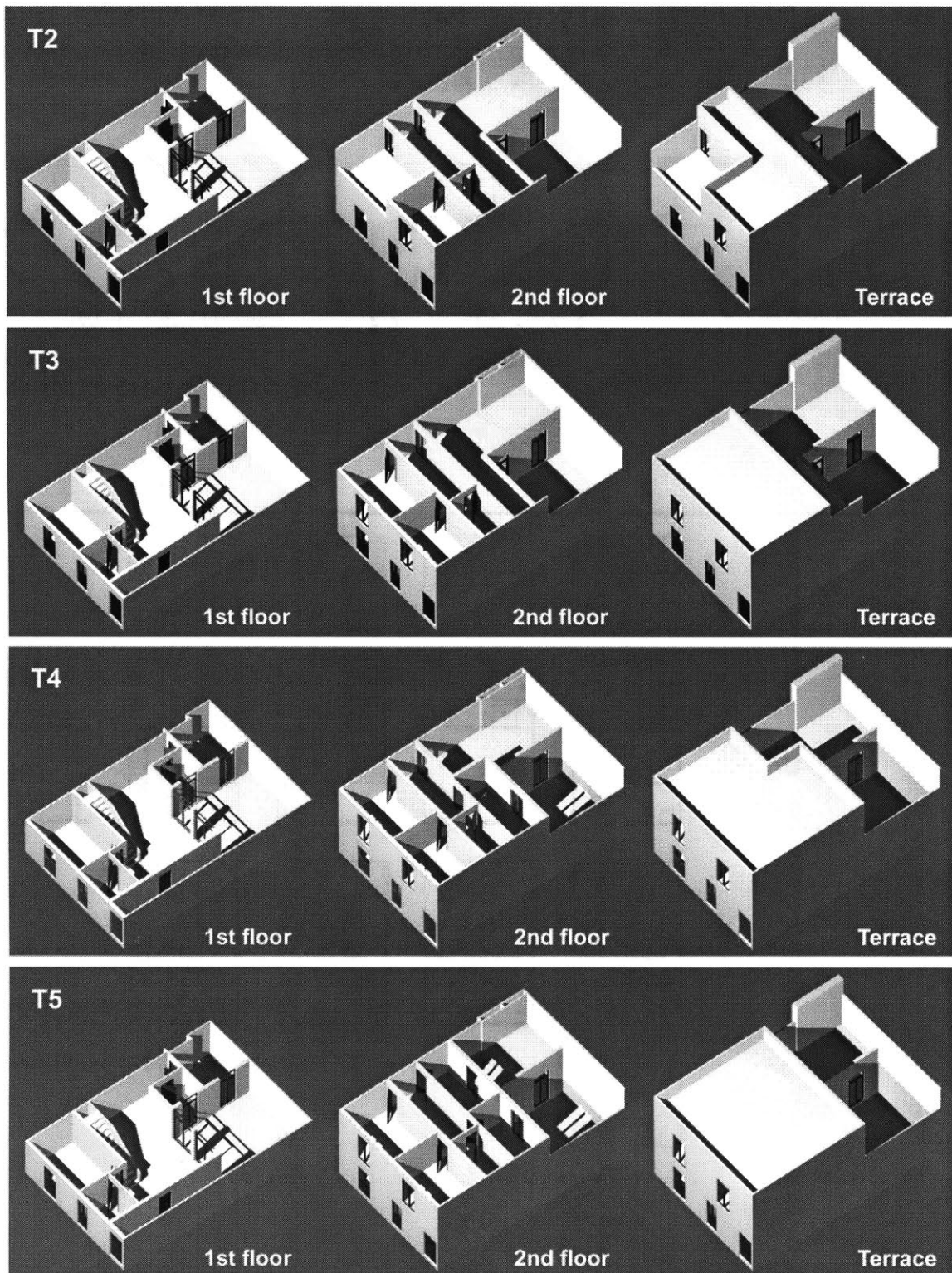


Figure 6.8 – Digital model of the new Malagueira design designed by the author of the grammar out of its rules in Experiment 2, and which Siza considered stylistically correct.

## Derivation of New Design

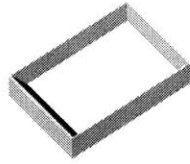
## Malagueira - Alvaro Siza Vieira

Stage 1: Define 1<sup>st</sup> floor

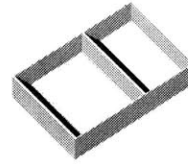
Step 1.1: Start floor



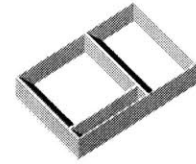
1. Introduce initial shape



2. Enclose floor



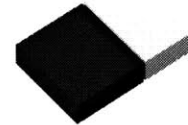
3. Locate in-/outside zones



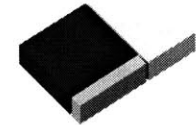
4. Create passageway zones



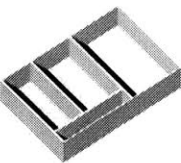
5. Locate sleeping/living zones



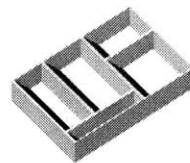
6. Locate service/yard zones



7. Locate staircase



8. Locate bedroom/bathroom



Step 1.3: Define circul.

Step 1.3: Define rooms

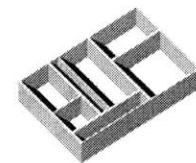
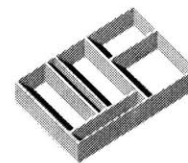


Figure 6.9 – Derivation of the new Malagueira design designed by the author of the grammar out of its rules, and which Siza considered stylistically correct.

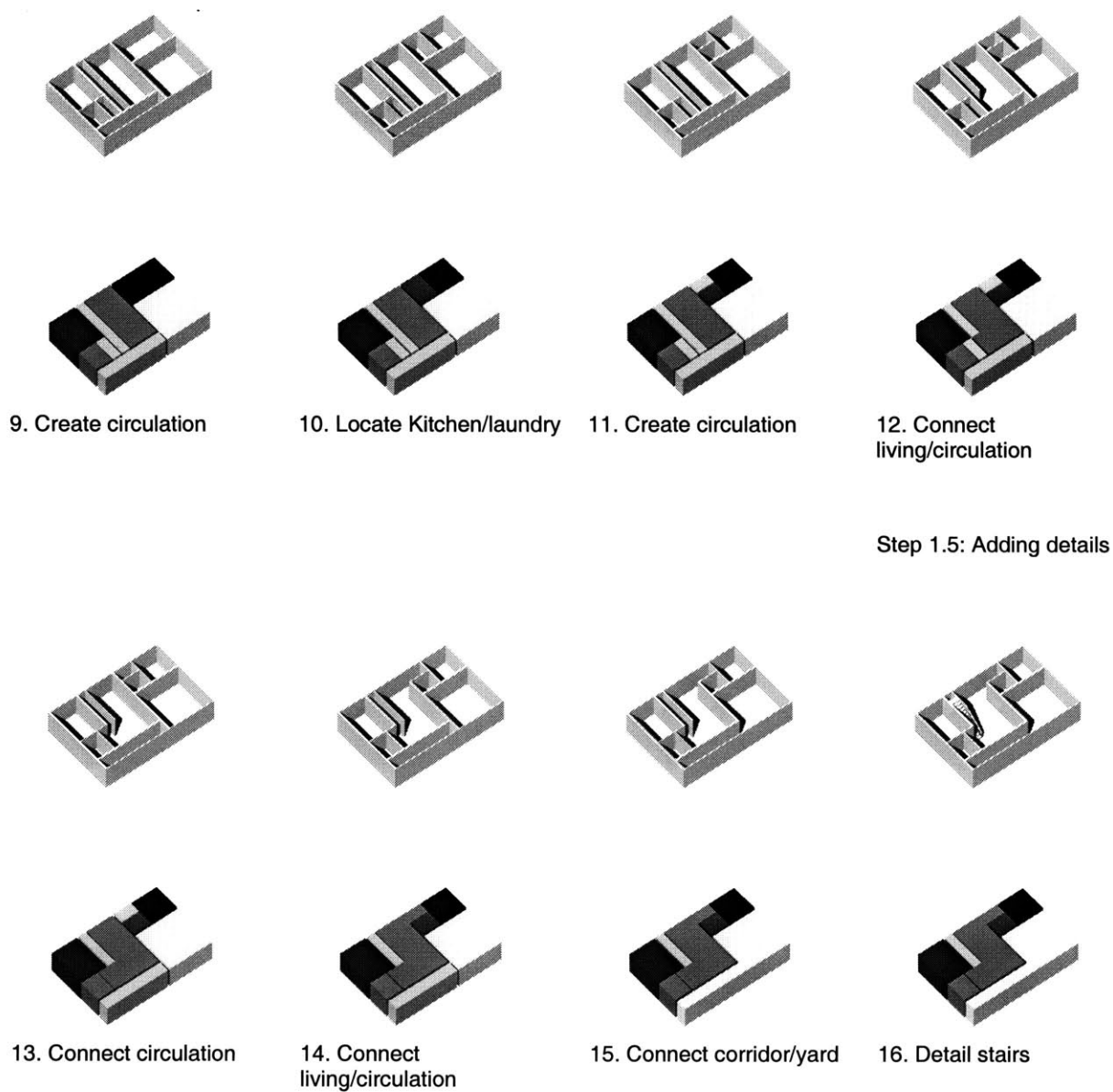


Figure 6.9 – Derivation of the new Malagueira design designed by the author of the grammar out of its rules, and which Siza considered stylistically correct.

Step 1.6: Create Openings

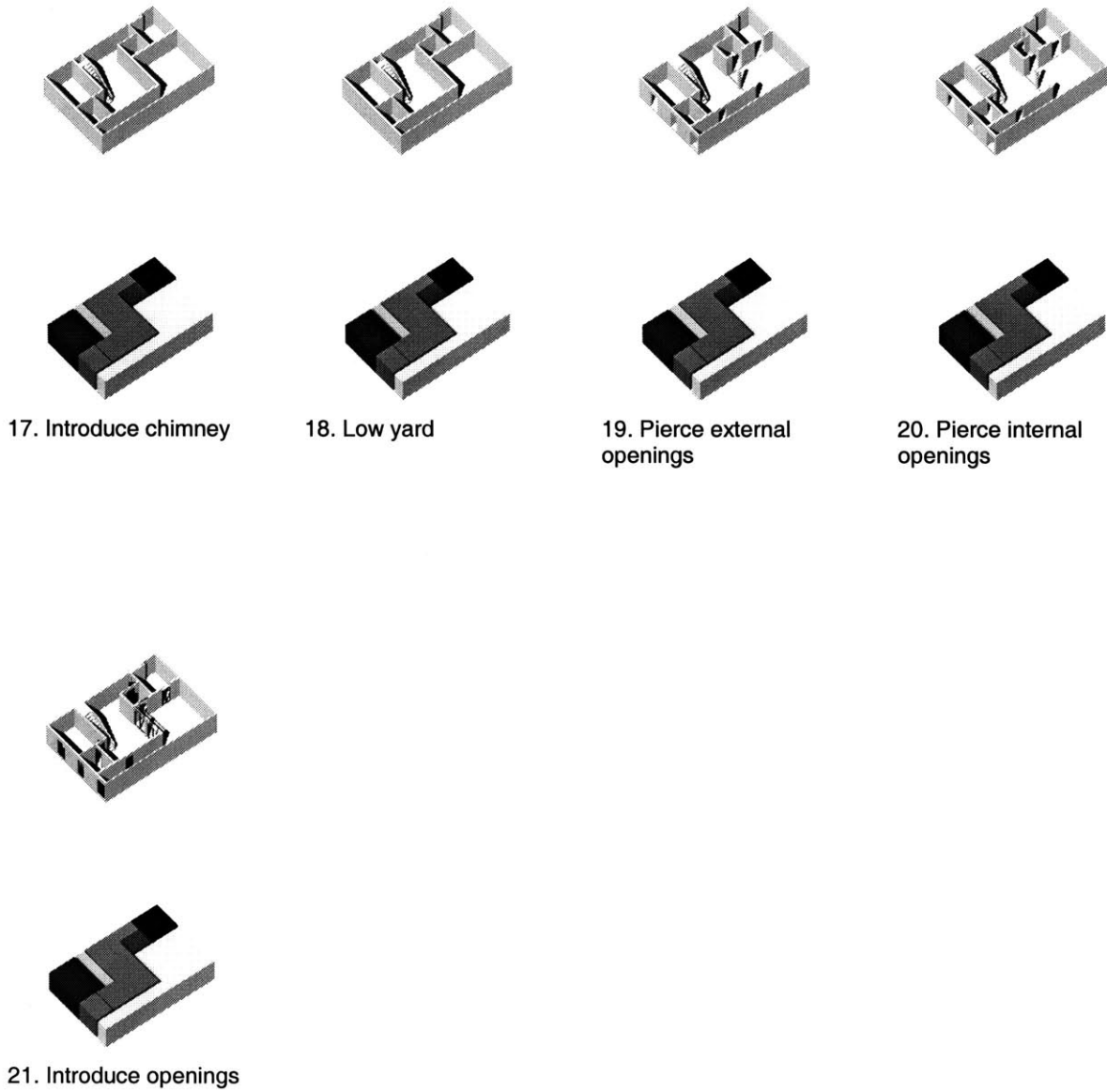
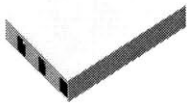


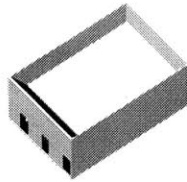
Figure 6.9 – Derivation of the new Malagueira design designed by the author of the grammar out of its rules, and which Siza considered stylistically correct.

Stage 2: Define 2<sup>nd</sup> floor

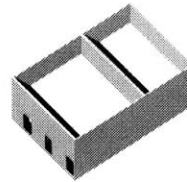
Step 2.1: Start floor



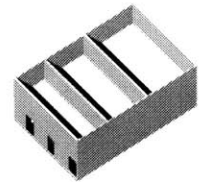
22. Introduce slab



23. Enclose floor



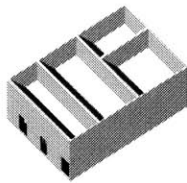
24. Replicate division into inside zones



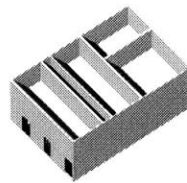
25. Replicate division into living/sleeping zones

Step 2.3: Define circulation scheme

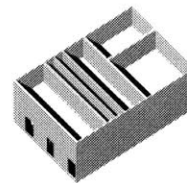
Step 2.4: Define rooms



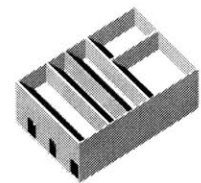
26. Replicate division into service/yard zones



27. Extend staircase

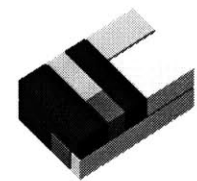
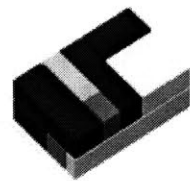
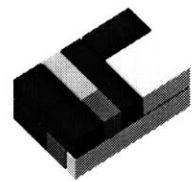
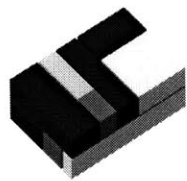
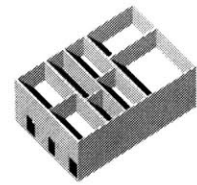
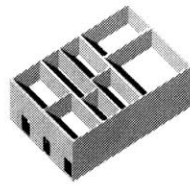
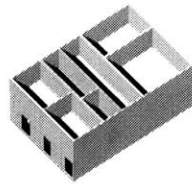
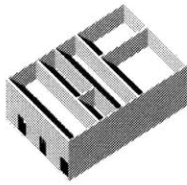


28. Define circulation scheme



29. Connect circulation

Figure 6.9 (continued) – Derivation of the new Malagueira design designed by the author of the grammar out of its rules, and which Siza considered stylistically correct.



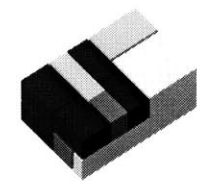
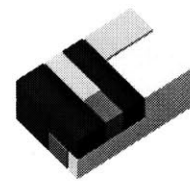
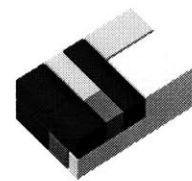
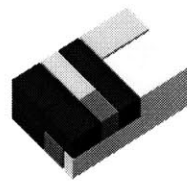
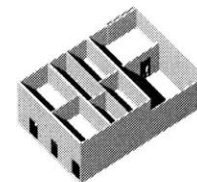
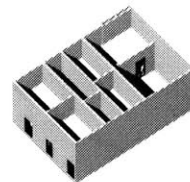
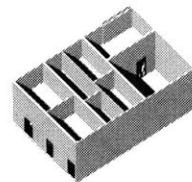
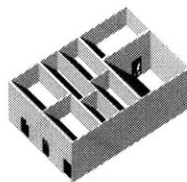
30. Locate bathroom

31. Locate bedroom

32. Locate bedroom

33. Assign terrace

Step 2.5: Adding details



34. Pierce slab

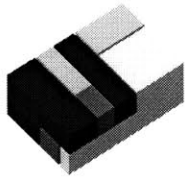
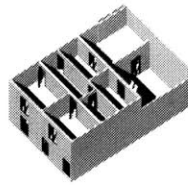
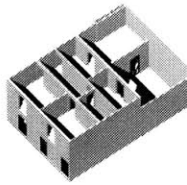
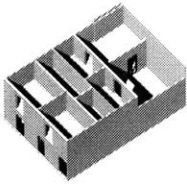
35. Extend chimney

36. Adjust wall height

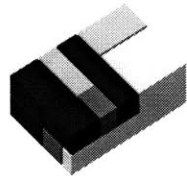
37. Adjust wall height

Figure 6.9 (continued) – Derivation of the new Malagueira design designed by the author of the grammar out of its rules, and which Siza considered stylistically correct.

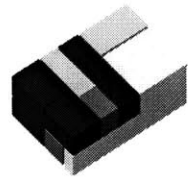
Step 2.6s: Create Openings



38. Pierce external openings



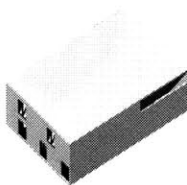
39. Pierce internal openings



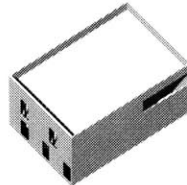
40. Introduce openings

Stage 3: Define terrace

Step 3.1: Start floor

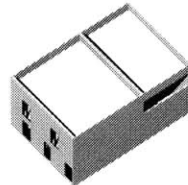


41. Introduce slab



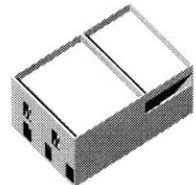
42. Enclose floor

Step 3.2: Replicate 1<sup>st</sup> floor divisions



43. Replicate division into in-/outside zones

Step 3.3: Define rooms

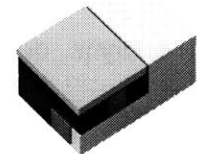
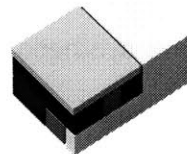
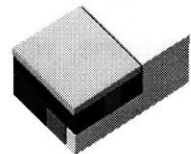
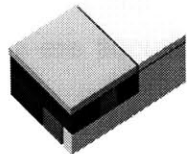
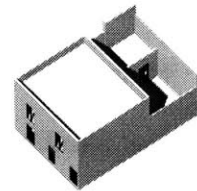
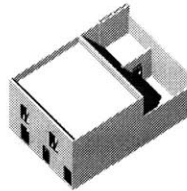
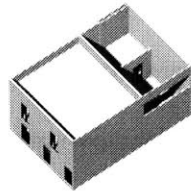
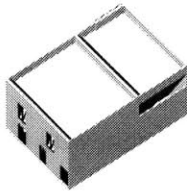


44. Assign terrace

Figure 6.9 (continued) – Derivation of the new Malagueira design designed by the author of the grammar out of its rules, and which Siza considered stylistically correct.



Step 3.4: Adding details



45. Extend chimney

46. Pierce slab

47. Adjust wall height

48. Adjust wall height

Figure 6.9 (continued) – Derivation of the new Malagueira design designed by the author of the grammar out of its rules, and which Siza considered stylistically correct.

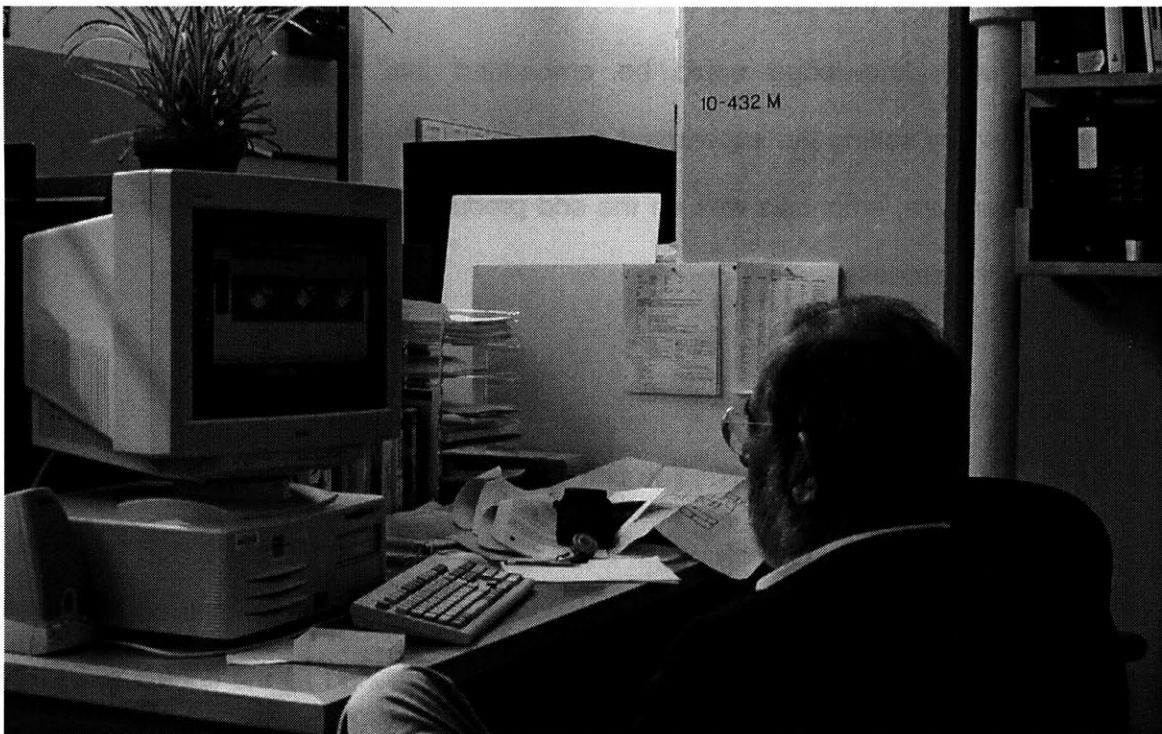


Figure 6.10 – Siza evaluating the new design by the author of the grammar, which he considered to be stylistically correct.

## **6.4 Experiment 3: goal-oriented design I (goal test)**

### **6.4.1 Goal**

This experiment simulated the situation in which a client asks a designer to design a Malagueira house, and the designer uses the grammar to generate a house in the style that satisfies the requirements. The experiment had two goals. The first goal was to test the ability of the grammar to generate criteria-matching designs, which corresponds to the performance of a goal test as described in Chapter 2. However, it also has implicit analytic, descriptive, and synthetic tests to determine, whether the designs are in the language, whether the grammar encodes the rules of syntax of Malagueira houses, and whether it conveys the rules to designers non-familiar with the design language. The second goal was to find how designers use the grammar to arrive at a design solution. Namely, it aimed at finding the criteria used to choose a particular rule at each step of the derivation, how that assured that the generated house satisfied the requirements, and how such knowledge could be embedded into the grammar. This was accomplished by setting the experiment as a protocol study in the way defined by Akin (1989). Therefore, emphasis was on the end product, as in the previous experiments, but also on the process.

### **6.4.2 Subjects**

There were four sets of subjects. The first set was a married couple with two kids who performed the role of a client. The second set was composed of an architect and a territorial engineer who assisted in the development of the grammar and the teaching tool mentioned in Section 6.2. These two subjects functioned as the control group and are referred to as subjects C1 and C2, respectively. The third set was formed by graduate students in architecture who had no previous knowledge of grammars,

including the Malagueira one. These students were divided into groups of two, each constituting a design team. These teams are referred to as subjects S1 through S5. The fourth set included Siza, who assessed the results to determine if the houses were in the Malagueira style.

### 6.4.3 Setting

Four settings were used in the experiment. (Figure 6.11) The first included two chairs, a table, a computer, and a video recording camera. This setting was used to interview the clients and to show the catalog of Malagueira houses to them. The second setting included a computer and an LCD projector, and it was used to describe the Malagueira grammar to the designers. The third setting included two chairs, a table, a video camera and a video recorder, paper, pencil, a list of the rules, and a table summarizing the dimensional requirements of Malagueira houses. It was used to videotape the designers generating houses. The fourth setting used a chair and a computer, and served to show the new designs to Siza.



a) Design subjects



b) Siza

6.11 – Different subjects and the corresponding experimental settings.

#### **6.4.4 Task**

The experiment included four tasks. The clients were assigned two tasks, the first was to describe the house that they needed, and the second was to comment on how each of the houses satisfied their needs. The designers were in charge of generating a house within the grammar that satisfied the client's requirements, as much as possible. Siza was asked to verify whether the houses were syntactically correct.

#### **6.4.5 Procedure**

The clients were asked to describe their desired house in three steps. Firstly, they were given complete freedom to describe their dream house. They had to talk about their family and their needs, starting with the things that they consider priority. Secondly, they were asked to consider cost issues, and set priorities. Thirdly, they were shown the Malagueira development, and the catalog of Malagueira houses, using the Web-based tool mentioned in Section 6.2, and asked to reframe their previous description to respect constraints imposed by the Malagueira framework. At the end of this step, they had to fill in a form with questions regarding their needs. This interview with the clients was videotaped.

The design part of the experiment last two weeks and included four work sessions. In the first week, the designers became familiar with the grammar through two lectures and two papers, one on grammars in general, and the other on the Malagueira grammar. They also were given the Web-based tool as a learning aid. In the second week, they had to design the house. First, they were shown the tape with the interview and then given the form that the clients had filled in. They were given two hours to sketch the grammar using a pencil and a stack of millimetric paper with rectangles representing the lot. They were asked to apply a rule on each rectangle and then to move on to the next

rectangle to continue with rule application. They were not allowed to use eraser to make it easier to keep track of all the changes to the design. Each time they made a mistake or wanted to backtrack, they had to restart the design on the next rectangle, from the point to which they wanted to backtrack. They also were asked to explain aloud what they were doing. While they designed, a camera recorded the graphic and verbal protocols of the process. Later, they were asked to design 2D plans and to build a 3D digital model of the house, and to look at the tape record of their design process and reconstruct it in the form of a tree-diagram. They also were asked to comment the diagram, explaining what they were trying to accomplish, why a particular rule had been selected, or why they had backtracked.

In the last procedural step, 3D digital and physical models of the houses were shown to Siza.

#### **6.4.6 Results**

A summary of the interview with the clients describing the requirements of their house is presented in Figure 6.12. A video still showing the gathering of protocol material is presented in Figure 6.13. A sample of the graphic protocol obtained at the end of one work session is included in Figure 6.14. The plans and the 3D digital models of the new designs are shown in Figure 6.15 and 6.16, respectively.

*Ideal scenario (after being given complete freedom to describe their dream house):*

A **master bedroom** for the couple with a private **bathroom** including a jacuzzi, and a shower. | Five **single bedrooms** for the children, of which 3 have private **bathrooms** and 2 share another **bathroom** with two washbasins. There should be closets in all the bedrooms. | A big **living room** with a fireplace, communicating with the **dining room** and the **studio** through sliding doors. | A big **kitchen** with a dining table or connected to the dining room. Its should include a small **pantry**, and it should be connected to a small **courtyard** to dry clothes. This courtyard should have no other access than through the kitchen. The kitchen should have a big window or sliding doors. | An **entrance hall** connecting the kitchen, the living room, and the dining room. | A big **garden** with a swimming pool. | A **restroom** to be used in connection with the studio. | They have 2 children now but they want to have 5. They want all the bedrooms built at the beginning.

*Real scenario 1 (after being asked to consider cost issues):*

A **double bedroom** and a **single bedroom** with a private **bathroom** each. | Four single **bedrooms** sharing 2 bathrooms with a shower, a bathtub, and two washbasins each. | A **laundry** next to the **kitchen** and connected to the **courtyard**.

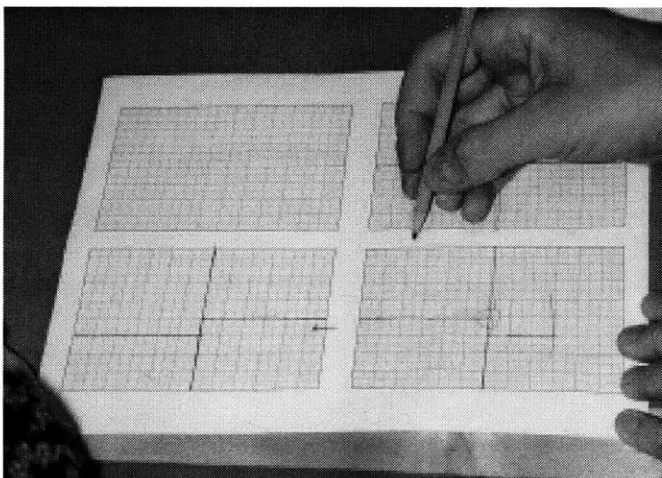
*Real scenario 2 (after being asked to relax a bit about cost issues):*

Three **bedrooms** with a private **bathroom** each. | A big **living room**. | A **dinning room**. | A **kitchen**. | A small **studio** connected to the living room through sliding doors. | An **entrance hall** connecting the kitchen, the living room, the dinning room, and a small **restroom**. | The sleeping area should be separated from the living and service zones by a corridor, or it should be on a separate floor. The studio should be on the ground floor. | They are four people now, but possibly five in the future.

*Malagueira scenario (after browsing through the Malagueira catalog and filling the form):*

A backyard house with street at the front, right, and back. | Two floors. | Four bedrooms (on the upper floor.) | No balconies (but will accept them if created). | A large living zone; a medium sleeping zone; a medium service zone ; a small yard zone. | A laundry and a pantry connected to the kitchen through a door. | A dining room connected to the kitchen through window. | A master bedroom connected to a bathroom.

Figure 6.12 – Summary of the interview with the clients, during which three scenarios were considered.



6.13 – Still of video protocol.





Type C1.1



Type C1.2



6.15 – Plans, sections and elevations of designs generated in Experiment 3.





C2



S1

6.15 (continued) – Plans, sections and elevations of designs generated in Experiment 3.



S2



S3

6.15 (continued) – Plans, sections and elevations of designs generated in Experiment 3.



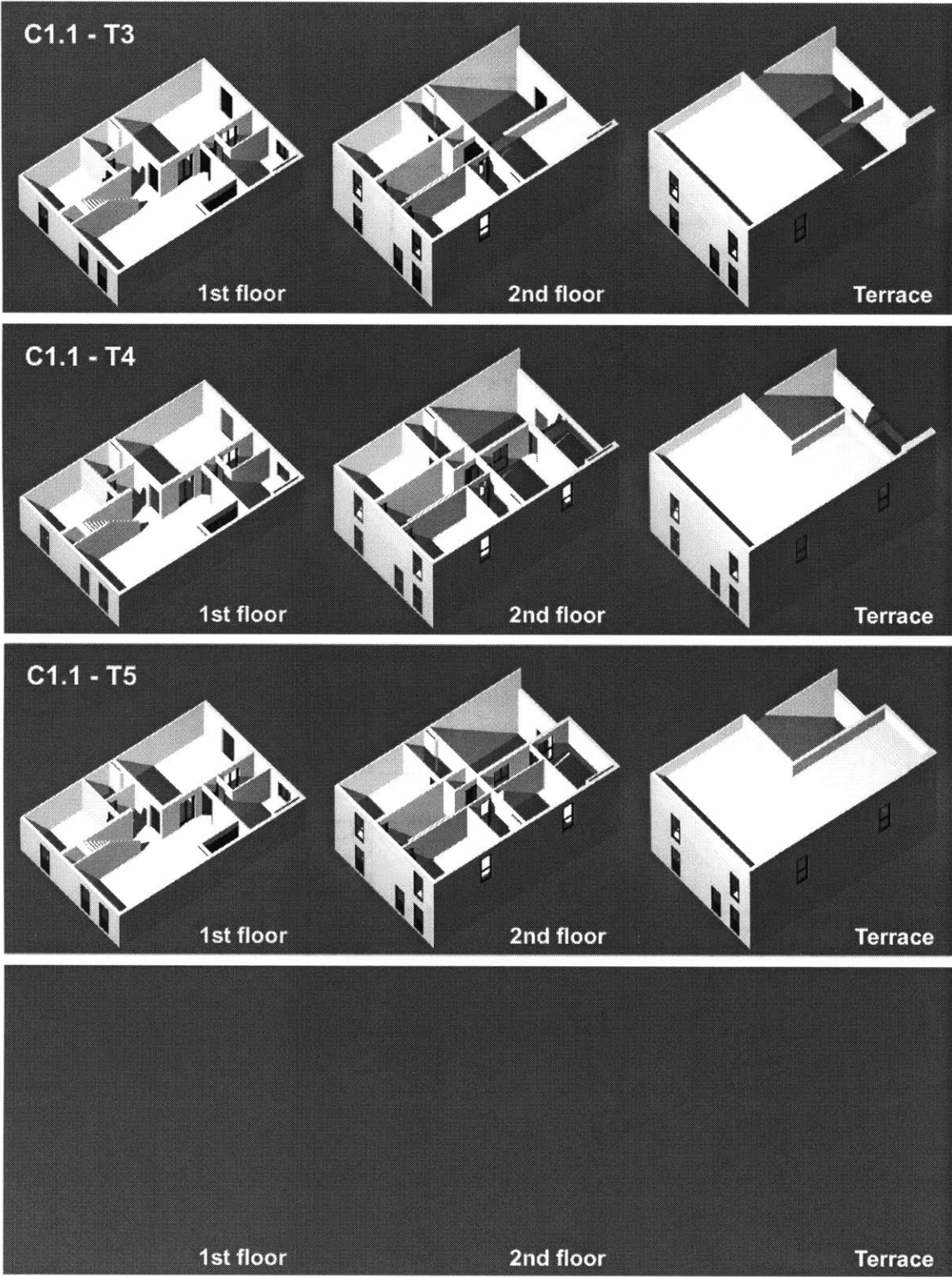
S4



S5

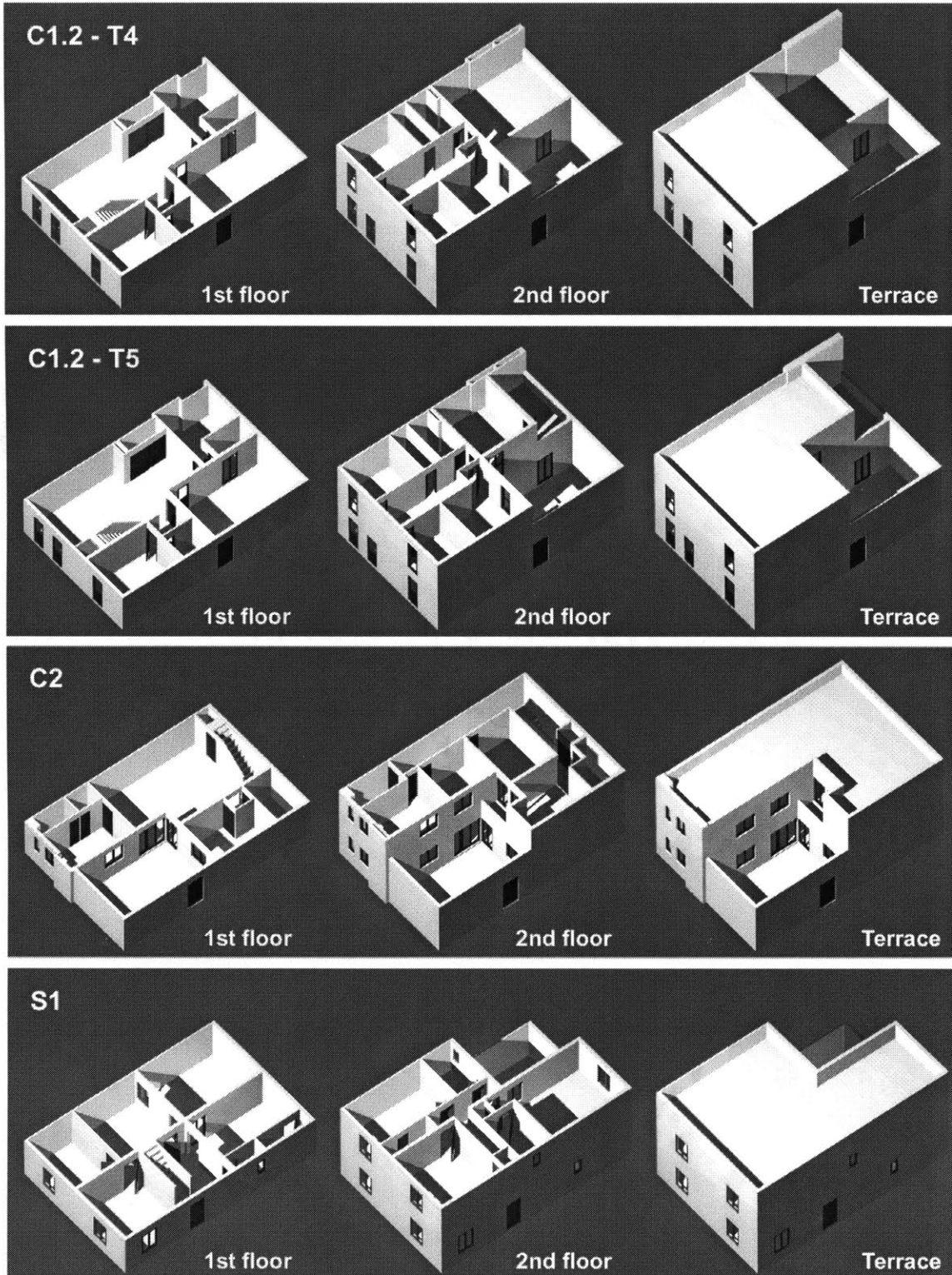
6.15 (continued) – Plans, sections and elevations of designs generated in Experiment 3.

**New designs** (same client, different authors) Malagueira - Alvaro Siza Vieira



6.16 – Digital model of designs generated in Experiment 3.

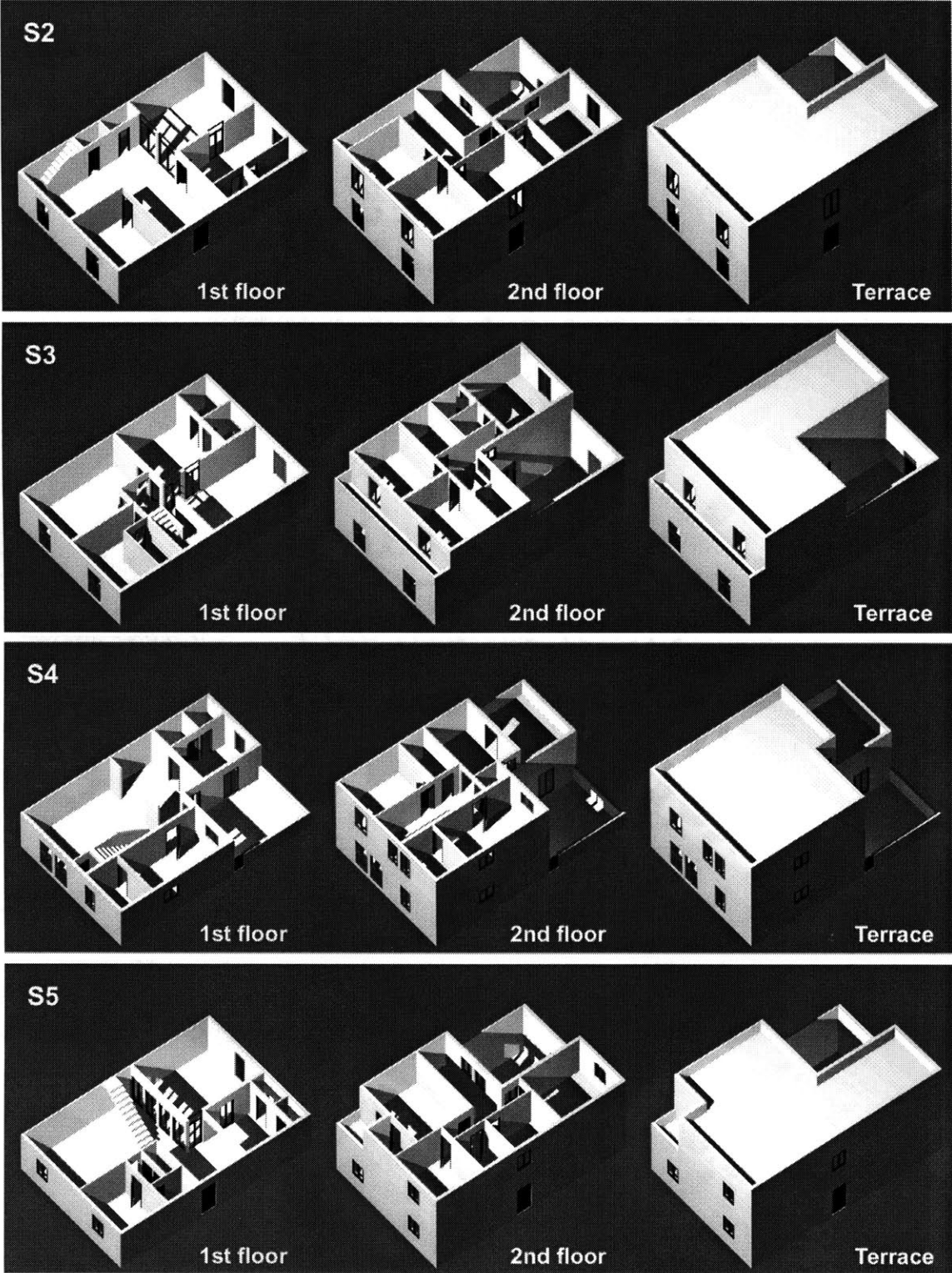
**New designs** (same client, different authors) Malagueira - Alvaro Siza Vieira



6.16 (continued) – Digital model of designs generated in Experiment 3.



**New designs** (same client, different authors) Malagueira - Alvaro Siza Vieira



6.16 (continued) – Digital model of designs generated in Experiment 3.

#### 6.4.7 Discussion

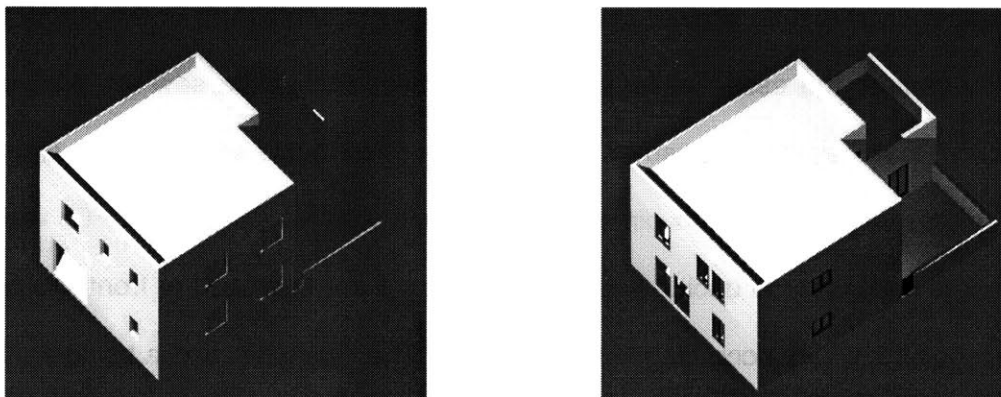
The first step in the analysis of results is to determine whether the designs are in the language. This can be done by asking Siza what he thinks, and by checking whether the design subjects respected the grammar rules. Then, one of three situations is possible. In the first situation, the designs respect both Siza's and the grammar rules, in which case the designs are clearly in the language, and the grammar succeeds both in capturing Siza's rules and in conveying them to designers. In the second situation, the designs respect the grammar rules, but they fail to respect Siza's, meaning that the grammar fails in capturing Siza's rules, let alone in conveying them to designers. In the third situation, the designs do not respect either of the rules. This might signify that the grammar captures Siza's rules, but that it fails to convey them to designers, or simply that it fails to capture the rules. It is, then, important to ask the design subjects about their difficulties in understanding the rules. In all the situations, it is necessary to determine the extent to which the rules are respected or disrespected.

Analytic test: are the designs in the language?

The experimental results show that the designs are not completely in the language, but that they capture the essential features. When Siza and Lopes saw the designs they pointed out that they did not respect Siza's design rules in two aspects. The first aspect was a disrespect of the patio dimensioning regulations (Table 4.3). To verify this aspect, it is necessary to decide whether the designs are backyard or frontyard houses, as the regulations vary according to the situation. The clients wanted a backyard house, but the urban context, with streets on three sides, makes the designs ambiguous. Considering them as frontyard, only design C2 respects such regulations on both floors and only designs C1 and E3 respect them on the ground floor. Considering the designs

as backyard, only design E5 respects the regulations on the ground floor. In the remaining designs, the patio is 0.5 m narrower than allowed by the regulations. However, the grammar deviated from the regulations by allowing frontyard patios to be less deeper on the second floor. This deviation existed because the grammar was developed before we were told of the regulations, and consequently, we based the rules on the observation of post-construction changes made by users in which the patios became smaller, and on the observation that overhangs were never allowed in any case. In these circumstances, all the houses are within the grammar. The rules could easily be corrected to follow the building regulations, but we decided to leave them as they are.

The second aspect mentioned by Siza concerned the openings. According to Siza, the openings in some designs did not respect his rules in terms of number, location, and dimension. (Figure 6.17) A look at the rules of the grammar confirms Siza's opinion. A possible explanation for this outcome might be the short time available for learning the grammar, which might have been not enough to allow subjects to become familiar with rules for design features perceived as secondary. Another explanation is that the rules are difficult to understand by designers. We will come back to this issue further below.



6.17 – The design by subject S5 before and after the openings were corrected. The drawings and models in Figures 6.15 and 6.16 show the corrected designs.



The analysis of the designs and the derivation processes revealed that some of the designs did not respect the grammar rules in other aspects, as well. Figure 6.18 partially diagrams the derivation of houses in this experiment. Recall that in the design of the grammar, it was decided to limit design possibilities to those that corresponded to the strictest interpretation of Siza's rules. For instance, only 32 out of 192 possible topological patterns were considered. In this strictest interpretation, designs E2 and E5 do not respect the rules because the living zone is not diagonally opposite to the patio zone. However, the design subjects considered this topological solution acceptable because in the clients' view the patio did not need to be accessed from all the other zones, and the internal distribution space was not the living room but an entrance hall, unlike in Siza's designs. In addition, Siza accepted such solutions. Then, in designs E2 and E5, the staircase is located in the middle of the living zone, a situation that was avoided in the strictest interpretation of Siza's rules followed in the design of the grammar. The disrespect of this rule by the design subjects was conscious and fostered by the need to satisfy the desire of the client to have an entrance hall. Siza's opinion was that their decision was acceptable. Finally, design E1 also disrespected the grammar by locating the dining room not adjacent to the kitchen. The reason for this decision was also to satisfy the client's requirement to have an entrance hall connecting the kitchen, the dining room, and the living room. However, it was not possible to satisfy this topological requirement, within the dimensional requirements set by the client, and the spatial configurations permitted by the grammar. In conclusion, the design subjects disrespected some of the rules to satisfy the clients' requirements, but the solutions were considered acceptable by Siza. Therefore, it is possible to change the grammar to match the universe of design solutions that corresponds to the broadest interpretation of Siza's design rules.

DEFINE 1st FLOOR

LOCATE FUNCTIONAL ZONES  
Locate outside / inside zones

Locate patio / service zones

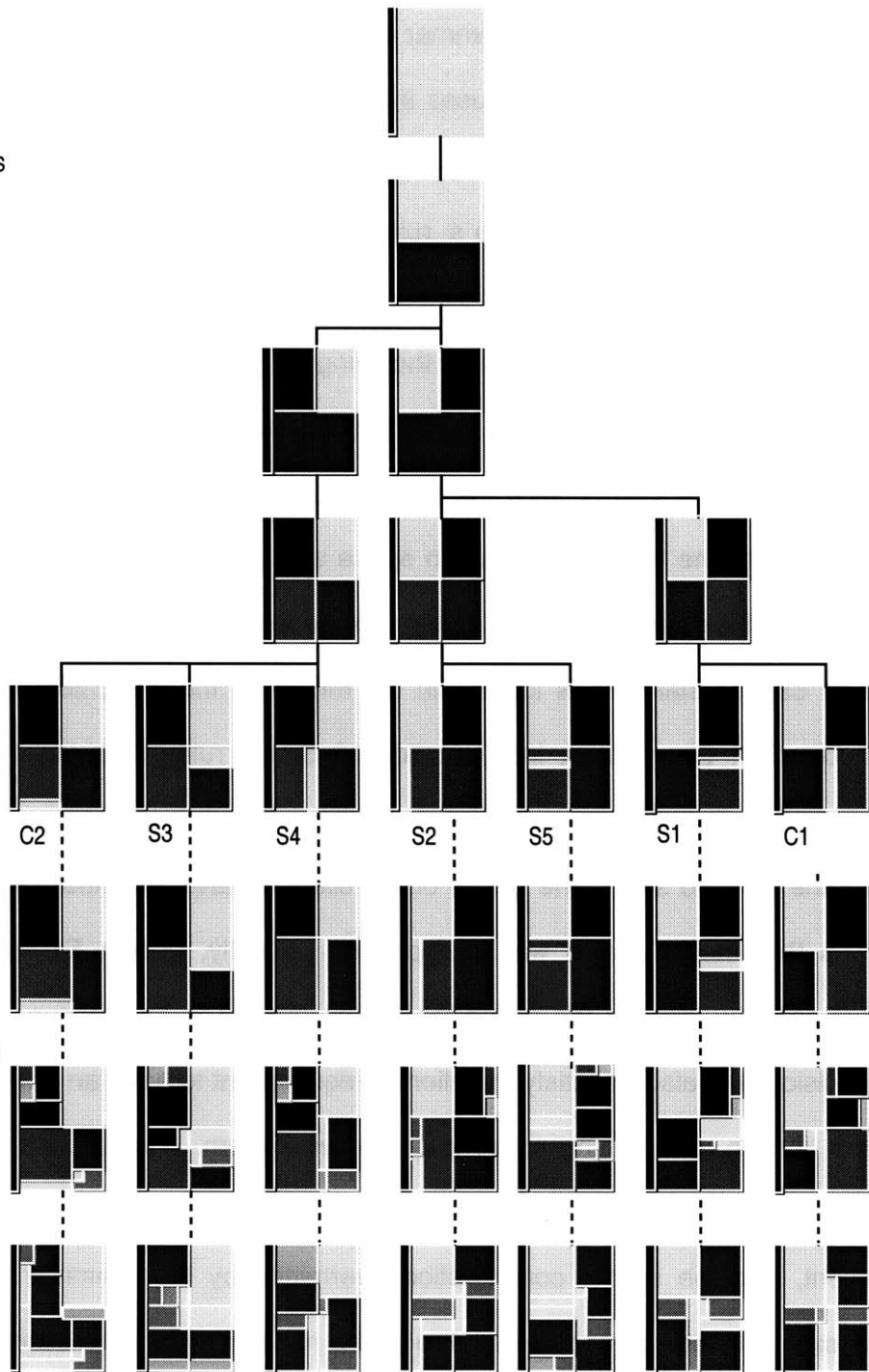
Locate remaining zones  
*Basic pattern defined*

LOCATE STAIRCASE  
*Type defined*

DIMENSIONING

DIVIDE FUNCTIONAL ZONES  
*Basic layout defined  
(functional organization)*

DEFINE 2nd FLOOR



6.18 – Partial tree diagram showing the derivation of the houses designed by the design subjects in Experiment 3. (Compare with the one in Figure 5.9)

*Descriptive test: does the grammar explain the common underlying features of designs?*

In interviews undertaken at the end of the experiment, the design subjects mentioned that there were too many rules and that the math and label parts of the rules were not intuitively apprehended in the short time available. These comments suggest that the grammar succeeded in encoding the rules, but it failed to convey them to designers in a visually understandable way, even with the help of the teaching tool. This result suggests that there is an important difference between developing a grammar to teach human designers and developing a grammar to implement in the computer.

Nevertheless, the design subjects were impressed by how fast they were able to generate a design solution, which is supported by the facts, as it only took them between 1h 45 m and 2h m to sketch a complete solution. Their explanation was that the grammar helped them through the decision-making process by providing a well-defined framework within which to work. Therefore, one has to conclude that the grammar succeeds in structuring the decision-making process so that designers understand how Malagueira houses are generated.

*Synthetic test: does the grammar tell how to generate new designs in the style?*

Above we saw that the designs were, to a certain extent, in the language. Therefore, we have to conclude that the grammar succeeds in specifying how to generate new designs that are instances of the style.

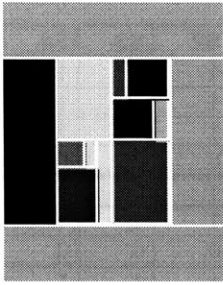
*Goal test: do the designs meet the given requirements?*

The importance of the goal test was twofold: (a) if the designer succeeded in generating a customized Malagueira house, to find out the criteria for choosing a particular rule at each step of the derivation, and how that assured that the house matched the program; and (2) if the designer failed, what changes were necessary to give the grammar such an ability.

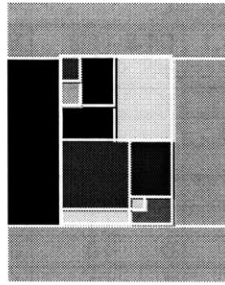
The first issue that immediately comes out of the experimental results is that the problem defined by the clients' requirements in the ideal home scenario, *is overconstrained*. It is overconstrained because it is impossible to satisfy such requirements within the framework provided by the Malagueira grammar. For instance, the area requirements that correspond to such an ideal home exceeds the area available in the 8 by 12 m Malagueira plot. Therefore, none of the designs meets the goal in the ideal scenario. For a similar reason, none of the designs meets the goal set in both real scenarios. On the other hand, the designs satisfy all the requirements that the clients specified in the Malagueira scenario, with the exception of including a fourth bedroom (designs S1 and S3,) and having the laundry adjacent to the patio (designs C2, S1, S2, and S4.)

The second issue is the *variety of design solutions*, which shows the potential of the grammar and Siza's design scheme. It also shows that there might exist different design solutions that satisfy a given problem in general terms. Nevertheless, among the different designs, some satisfy the requirements better than others.

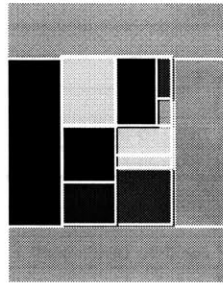
First floor



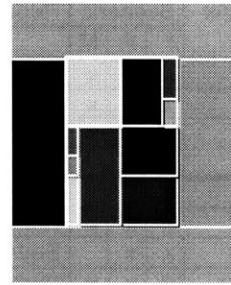
C1 t4



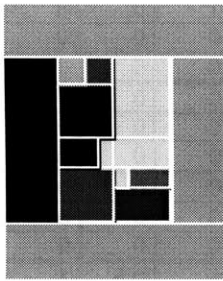
C2 t4



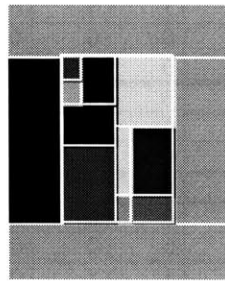
S1 t3



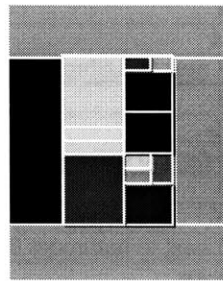
S2 t4



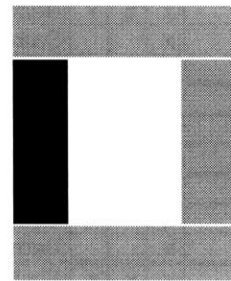
S3 t3



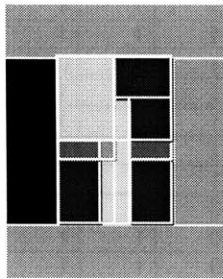
S4 t3



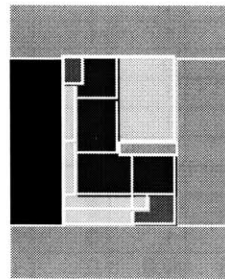
S5 t4



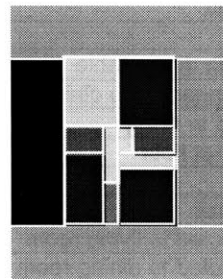
Second floor



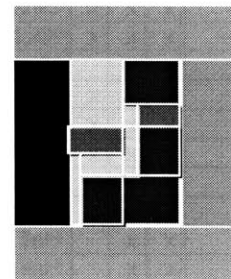
C1 t4



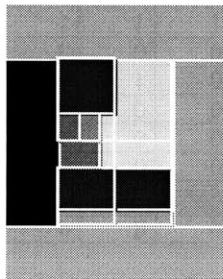
C2 t4



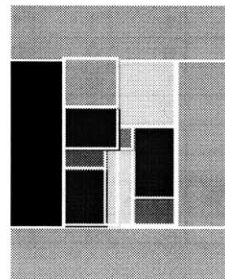
S1 t3



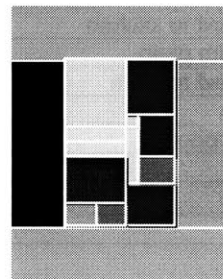
S2 t4



S3 t3



S4 t3



S5 t4

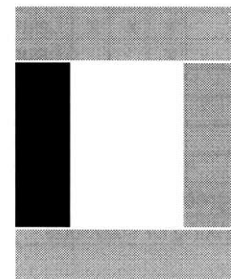


Figure 6.19 – The designs generated in Experiment 3 placed in their urban context. Black rectangles represent houses, whereas gray rectangles represent streets.

**Table 6.1 – Satisfaction of the clients' requirements by the designs**

Rank	Scenario			Requirements	Designs							
	I	R	M		C1.1	C1.2	C2	S1	S2	S3	S4	S5
1	x	x	x	Couple bedroom 1	1	1	1	1	1	1	1	1
2	x	x	x	Bathroom 1	1	1	1	1	1	1	1	1
3	x	x	x	Bedroom 1 next to bathroom 1	0.5	1	1	1	1	1	1	1
30	x			Jacuzzi and shower	0	0	0	0	0	0	0	0
4	x	x	x	Double bedroom 2	1	1	1	1	1	1	1	1
5	x	x	x	Bathroom 2	1	1	1	1	1	1	1	1
21	x	x		Bedroom 2 next to bathroom 2	0	0	0	0	0	0	0	0
6	x	x	x	Bedroom 3	1	1	1	1	1	1	1	1
22	x	x		Bathroom 3	0	0	0	0	0	0	0	0
23	x	x		Bedroom 3 next to bathroom 3	0	0	0	0	0	0	0	0
7	x	x	x	Bedroom 4	1	1	1	0	1	0	0	1
31	x			Bathroom 4	0	0	0	0	0	0	0	0
32	x			Bedroom 4 next to bathroom 4	0	0	0	0	0	0	0	0
33	x			Bedroom 5	0	0	0	0	0	0	0	0
34	x			Bathroom 5	0	0	0	0	0	0	0	0
35	x			Bedroom 5 next to bathroom 5	0	0	0	0	0	0	0	0
36	x			Bedroom 6	0	0	0	0	0	0	0	0
37	x			Bathroom 6	0	0	0	0	0	0	0	0
38	x			Bedroom 6 next to bathroom 6	0	0	0	0	0	0	0	0
8	x	x	x	Big living zone	1	1	1	0.5	1	0.5	1	1
9			x	Medium sleeping zone	1	1	1	0.5	1	0.5	1	1
10			x	Medium service zone	1	1	1	0.5	0.5	0.5	1	1
11			x	Small patio	0.5	0.5	0.5	1	1	0.5	1	1
12	x	x	x	Living room	1	1	1	1	1	1	1	1
13	x	x	x	Dining room	1	1	1	1	1	1	1	1
14	x	x	x	Kitchen	1	1	1	1	1	1	1	1
39	x			Kitchen with dining table	0	0	0	1	1	1	0	1
15	x	x	x	Kitchen connected to dining	1	1	1	0.5	1	1	1	1
16	x	x	x	Studio next to living	1	1	1	1	1	0	1	1
17	x	x	x	Studio on 1st floor	1	1	1	1	1	0	1	1
24	x	x		Restroom	1	1	1	1	1	1	1	1
25	x	x		Entrance hall	1	1	0	1	0	1	1	1
26	x	x		Hall connected to living room	1	1	0	1	0	1	1	1
27	x	x		Hall connected to dining room	0.5	0.5	0	1	0	1	1	1
28	x	x		Hall connected to kitchen	0	0	0	1	0	1	0	0
29	x	x		Hall connected to restroom	1	1	0	1	0	1	0	0
18	x	x	x	Sleeping separate from li. & se.	1	1	1	1	1	0	1	1
19	x	x	x	Laundry next to kitchen	1	1	1	1	1	1	1	1
40	x			Laundry with patio	1	0	0	0	0	0	0	0
20		x	x	Laundry next to patio	1	1	0	0	0	1	0	1
41	x			Big garden	0	0	0	0	0	0	0	0
42	x			Swimming pool	0	0	0	0	0	0	0	0
43	x			Fireplace	0	0	0	0	0	0	0	0
Degree of satisfaction					77%	77%	68%	64%	68%	70%	71%	78%
Ranking of solutions					3	2	6	7	5	8	4	1

Note : The design requirements were ranked as follows. The requirements that existed in the Malagueira scenario were ranked first. The requirements that existed in the real and Malagueira scenarios were ranked second. The requirements that existed in all the scenarios were ranked third. The requirements that existed in the real scenario were ranked fourth. Finally, the requirements that only existed in the ideal scenario were ranked fifth. Within each category, the requirements were ranked in chronological order of specification by the clients, except if they specifically mentioned otherwise. The number 1 means that the design satisfies the requirement,

zero means that it does not, and 0.5 means that it design does not satisfy the requirement, but almost. The degree of satisfaction of a design is the ratio between the score of the design and the maximum possible score expressed in terms of percentage.

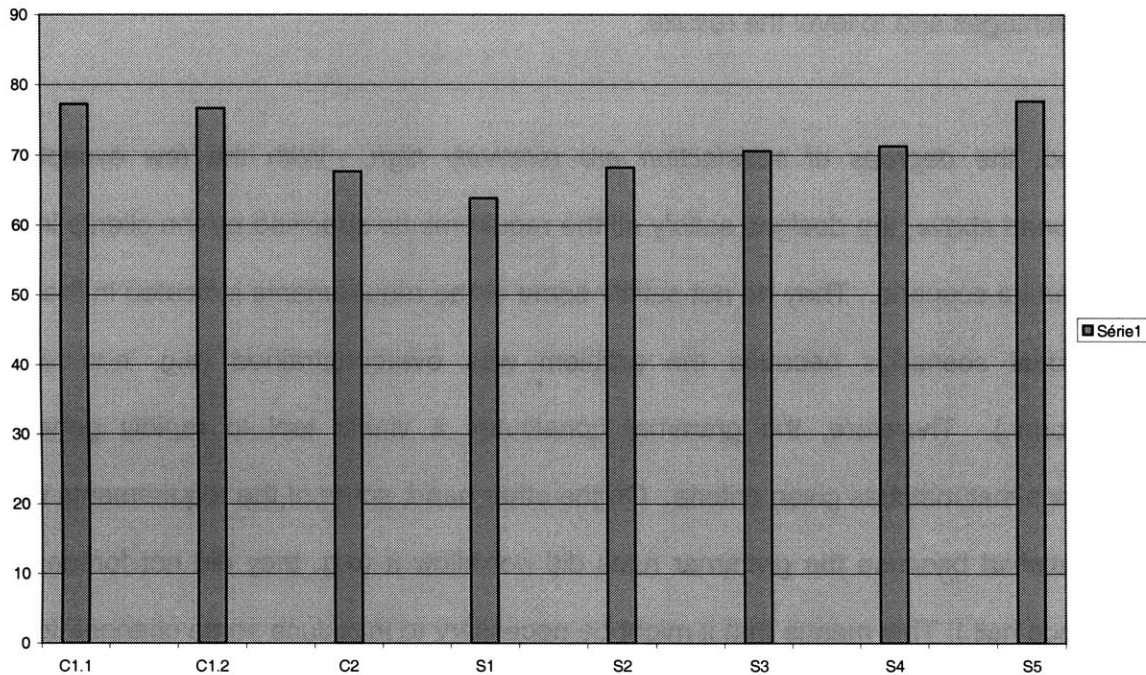


Figure 6.20 – Satisfaction of the clients' requirements by the different designs

By ranking the design requirements specified by the clients in all the scenarios according to their order of importance for the client, it is possible to rank the designs in terms of the degree of satisfaction of such requirements. (The functional organization of the different designs is shown in Figure 6.19, the satisfaction of the clients' requirements is shown in Table 6.1 and in Figure 6.20.) Such ranking permits some other interesting observations.

First, the *degrees of satisfaction of the different designs are all fairly close*. Namely, the architect in the control group did not perform better than the remaining subjects who also were architects. The designs of this control subject ranked 2<sup>nd</sup>, which indicates that her previous knowledge of the Malagueira framework might have helped her, but not

significantly. Moreover, the design of the subject who was not an architect was not the worst. Therefore, these results suggest that the grammar contributed to eliminate such disadvantages and to level the results.

Second, the *degrees of satisfaction are relatively high*. With the few exceptions mentioned above, the designs satisfy all the requirements specified by the clients in the Malagueira scenario. They do not satisfy some of the requirements indicated in the real and ideal scenarios because the problem was overconstrained (e.g. number of bedrooms.) Therefore, the grammar constitutes a viable tool to rapidly generate solutions that matches given criteria. On the other hand, some of the requirements were not satisfied because the grammar rules did not allow it (e.g. they did not foresee an entrance hall.) This means that it might be necessary to introduce some changes to the grammar to increase the possibility of client satisfaction. Namely, these changes should aim at allowing the generation of houses for programs that were not initially foreseen, but that can be satisfied within the framework, anyway. This can be done by introducing new rules into the grammar, but this requires one to anticipate such programs, which might be difficult. A better way is to give the rules a degree of generality so that the generation of spatial configurations is prompted by the housing program. For instance, instead of having a specific rule that dissects a rectangle into kitchen and laundry, one could have a general rule that dissects a rectangle into any two adjacent spaces.

The third observation is that some design subjects did better than others, despite the degree of satisfaction are all high and close. Design S5 (78%) is the best, followed by subject C1's two designs (77%,) S3 and S4 (71%,) C2 and S2 (68%,) and S1 (64%). This result also corresponds to the perception of the clients who, when asked which house satisfied their requirements better, indicated designs S5 and C1.1. This means



that our ranking system captured, at least roughly, the priorities set by the clients. Design S5 is better than C1.1 and C1.2 because it includes an entrance hall and a small patio, which are not permitted by the grammar. Design S3 and S4 are worse because they do not include a fourth bedroom, or consider its future allocation, despite the fact that they do better regarding the entrance hall (especially S3.) Designs C2 and S2 rank worse because they do not include any form of entrance hall. Finally, design S1 does not respect the relative sizes of the functional zones, which contributes to make it the worst of the group. For instance, design S1 has big bedrooms and a relatively small living room, but the clients indicated big and medium as the area requirements for the living and sleeping zones.

It is then interesting to find out why some subjects did better than others.

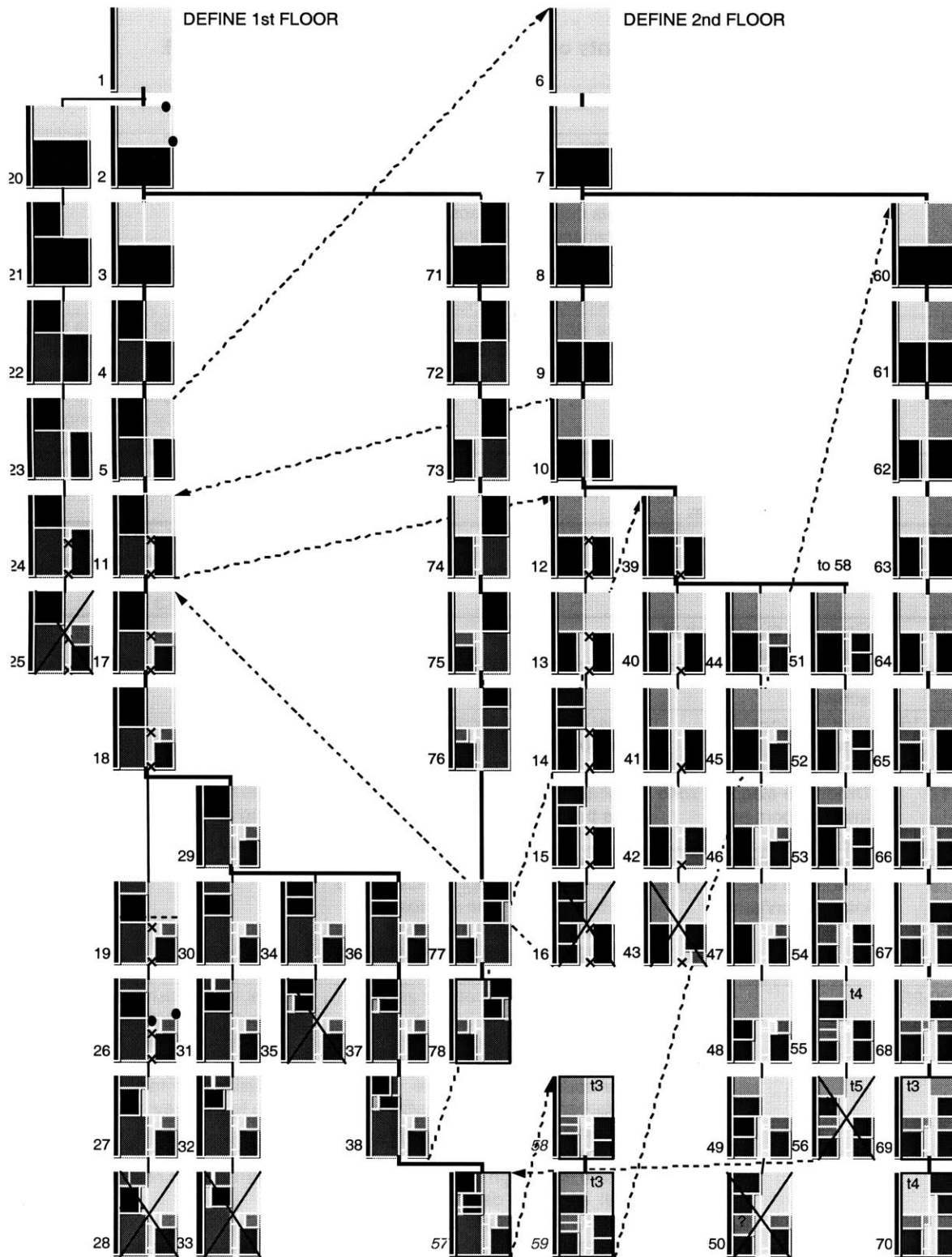
The first explanation is that *they set the right priorities* before starting to design. Because the problem was overconstrained, the design subjects had to decide which requirements were more important to satisfy. Therefore, they interpreted the program in different ways by augmenting the importance of some requirements, and diminishing the importance of others. Results show that the interpretation of those design subjects that achieved better results is closer to the clients' intentions. Consider, for instance, design subject C1. (We chose her design to illustrate the discussion below because it is the one that achieves the best balance between respecting the rules and satisfying the clients' requirements.)

Design subject C1 thought that it was unlikely that they would have five children as they mentioned in the ideal scenario, but she admitted that they could have one more child, as they had mentioned in the real scenario. Therefore, she decided that she would try to

allocate, one bedroom for the parents, one bedroom for each of the existing children, and, eventually, another for the future child. Consequently, before she started to design, she mentioned that there were three possible design briefs: (1) a three bedroom apartment and a studio/bedroom; (2) a three bedroom house that could expand up to four bedrooms, and a studio; (3) a four bedroom house that could expand up to five bedrooms. She also said that she intended to allocate three bathrooms: one in the lower floor for guests, and two in the upper floor (one for the parents, and the other for the children). It turns out that this order of priorities corresponded to the clients' own priorities.

The second explanation is that the design subjects who achieved a better result were more *successful in informing their derivation process with their a priori interpretations*. Consider the derivation of design subject C1 reconstituted in Figure 6.21 after the video and graphic protocols. The comments on the derivation process by the designer are shown in Table 6.2, as well as other analytical comments made a posteriori.

The history of her derivation process can be described as follows. First, she developed the design up until obtaining a stair pattern (moves 1-5). However, she did not decide the exact location of the sleeping and service zones. Instead, she moved on to the upper floor to see how that stair pattern worked on the second floor (moves 6-10). Before continuing studying the functional organization of the upper floor, she quickly returned to the lower floor to dimension the staircase (move 11). Then she returned to the upper floor to sketch its functional organization (moves 12-16).



6.21 – Tree diagram of the derivation of the houses designed by subject C1 after the video and graphic protocols. The numbers indicate the sequence of design moves (rule applications.) Please compare each move with the corresponding comments on Table 6.2. The small crossed lines indicate the location of the staircase landings. The big crossed lines indicate that the derivation reached a dead end. The bold rectangles indicate her design solutions.

**Table 6.2 - Comments of the derivation diagram in Figure 6.19.**

Move	Rule	Design subject's comments <sup>(1)</sup>	Analytical comments <sup>(2)</sup>
1	Introduce initial shape		
2	Divide the lot into inside and outside zones	There is no verandah on the upper floor, therefore the dissection is in the middle. The dots indicate the possible locations for the entrance to the patio	
3	Divide the outside zone to allocate the patio.	The location of the service and sleeping zones is not definite yet.	
4	Divide the inside zone to allocate the living.	The location of the service and sleeping zones is not definite yet.	The living is at the opposite corner of the patio as in all the Siza's houses.
5	Locate the staircase.		The staircase is placed in the living zone, but by borrowing area to the sleeping zone
6-10	Replicate the division of the lower floor into zones and staircase.		
11	Divide the staircase into landings and main steps.		She made a mistake by not considering that the landings could steps too. $2 \times 0.9 + 3.5 = 5.3$
12	Replicate the division of the staircase.		
13	Define circulation scheme		
14-16	Divide "zones" into rooms.	First attempt to locate the bedrooms and the bathrooms.	The exact locations of the bedrooms and bathrooms are not indicated.
17	Divide the sleeping zone into bedroom and bathroom	Locating the bedroom/studio and the bathroom	Decides the location of the sleeping and service zones.
18	Create circulation	I do not like the hall.	
19	Divide the sleeping zone into kitchen/laundry	<i>There is not enough space for the entrance hall. The service zone is too big, and the living and dining rooms are joint but they do not have enough area.</i>	
20-25		<i>I will redivide the lot into zones to make the patio shorter, as desired by the client. The sleeping zone gets too big, and the house is too massive relatively to the lot. I realize I made mistake regarding the staircase.</i>	Attempt to diminish the service area and increase the living area, but this division does not respect the Malagueira grammar rules. She realizes her mistake and returns to the previous solution.
26		There is enough space for the hall, after all. I can now define the entrance to the plot and the house.	She corrects her mistake.
27	Create a circulation.	Creating a circulation to access the laundry. The kitchen incorporates an informal dining area.	She did not perceive that the circulation could be external. This would allow to connect the laundry to the patio, as desired by the client, and bring direct light to the kitchen.

28	Divide laundry into wash and dry areas.	I will divide the laundry into wash and dry areas, being the latter close, or even connected to the patio as desired by the client. <i>The service zone is too big, and there is not enough space for a joining dining/living areas</i>	She satisfies the clients requirements, regarding the connections between the kitchen, the laundry, and the patio, but is not happy with the formal dining area.
29-31	Locate the kitchen.	Reduce the service area to increase the living/dining area. This makes it possible to have enough space in the living zone for a formal dining area for all the residents. The informal dining will only have enough space for two people.	She perceives that there is not enough space for both formal and informal dining areas for all the residents and <u>she</u> makes a trade - off judgement.
32	Create a circulation	She mentions that she could widen the circulation to make an informal dining area close to the patio.	She abandons this solution because the kitchen has no direct sunlight.
33		The other circulation worked better because it was closed to the entrance, but the kitchen gets natural light now. <i>The pantry is missing.</i>	The relative positions of the kitchen and the circulation are reversed to allow the kitchen to get direct sunlight, but she is unhappy with both solutions. In addition, they have no pantry and she abandons them.
34	Locate the kitchen	The kitchen can get light in this position.	
35	Locate the pantry	This is the transitional space. <i>The kitchen is too small.</i>	She is thinking of the transitional space as an informal dining area, but the kitchen is too small.
36	Locate the kitchen.		
37	Locate the pantry.		
38	Locate the laundry	The laundry is closed to the patio.	She satisfies the client's requirements regarding the spaces included in the service area.
39-41			She redesigns the upper floor due to correct the mistake the staircase mistake.
42	Locate bathroom		She seems to have preferred to locate the bedroom facing the patio.
43	Create circulation.		She seems to realize that the bathroom on the upper floor is not above the bathroom on the lower floor, and <i>she abandons this solution.</i>
44	Locate bathroom		She redesigns the upper floor, making sure that the bathrooms are on the top of each other.
45	Create circulation		This bathroom is the private bathroom.
46	Create circulation		
47	Create closet		
48	Locate bedrooms	This other space can only be a bathroom because it does not have direct sunlight.	
49	Locate bedroom and terrace/bedroom	This solution is abandoned because one of the bedrooms has no sunlight.	

50	Locate bathroom	<i>I have too many bedrooms.</i>	She realizes that she has too many bedrooms and she abandons the solution.
51-55			She redesigns the upper floor, turning the room with no access to light into two bathrooms, and the other bathroom into a bedroom.
56			She tries to allocate a fifth bedroom, but she realizes that the fourth bedroom would become interior, and she backtracks.
57		I will enlarge the transitional space and turn it into a single dining area.	She redesigns the lower floor with no changes, except that at the end, she realizes that she can enlarge the transitional space, thereby merging the informal and formal dining areas into a single space. (She places the windows and doors.)
58-59			She redesigns the solution for the upper floor, redimensioning the rooms, and showing how it can evolve from three up to four bedrooms. (She places the windows and doors.)
		At this stage I have considered to mirror the solution to take better advantage of the urban context.	She did this after the protocol experience finished.
60-70			Developing the mirrored solution, correcting aspects perceived as problematic.
71-78			Developing the mirrored solution, correcting aspects perceived as problematic.

(1) These comments are adapted from those made by the design subject in the video protocol, and the comments that she made in the reconstitution of her derivation process. These comments try to explain the intention of the design subject during her designs moves, when her comments are not clear enough.

After confirming that such a stair pattern worked for the upper floor, she decided to place the service zone on the side of the living zone because she saw an opportunity to repeat a pattern of functional organization on both floors: the master bedroom connected to the bathroom on the upper floor, and the studio/bedroom connected to the restroom on the lower floor (moves 17-18). (when she reached move 16, she did not went on to allocate the master bedroom and the bathroom but she saw the opportunity to do so and went back to the upper floor to allocate the studio and the restroom).

Then she moved on to the service zone and allocated the laundry (move 19). While studying the circulation pattern in the service zone, she worried about the design of the staircase and the entrance hall. Because she made a mistake in the dimension of the staircase and thought that she would have no space to allocate a satisfactory entrance hall, she decided to experiment with another basic pattern (moves 20-25).

After realizing her mistake, she returned to the initial basic pattern to resume the study of the service zone. She created a corridor in the kitchen, and she divided the laundry into wash and dry areas (moves 25-26). Then she perceived that living zone in which she included the living and the formal dining areas was too small, and increased the living zone at the expense of the service zone (moves 29-32). She explained that initially she was trying to allocate a big kitchen because she wanted to include an informal dining area for all the residents, but this jeopardized the possibility of allocating a formal dining area for them all in the living room. Therefore, she decided that it was better to have a formal dining area for all the residents, and an informal one for only two, or so. Then she realized that the kitchen had no direct natural light and experimented to flip the position of the kitchen and the circulation (move 33). She was not happy with the new solution because one had to traverse the living zone to reach the kitchen and seemed to there be no room for the pantry. She also realized that the laundry probably was too big relatively to the kitchen. She looked at Siza's solutions and backtracked to start dividing the service zone from scratch. Then she allocated the kitchen, the transitional space, and the pantry, in this order (move 34-35,) before realizing that the kitchen was too small relatively to the transitional space (perceived as the informal dining area), and enlarging it (move 36-37). Then she allocated the laundry, placing it close to the patio (move 38).

At this stage, she seemed fairly satisfied with the functional organization of the lower floor and returned to the study of the upper floor. She started the division of the upper floor from the stair pattern to correct the stair design mistake. Then she allocated the master bedroom and the private bathroom (moves 39-43). She did not like the solution because the bathroom was not on the top of the one on the lower floor and she backtracked. Then, she redefined the functional organization of the upper floor, reproducing her initial sketch, but placing the bathrooms on the top of each other (moves 44-50). She realized that one of the spaces could only be a bathroom because it had no natural light. However, such a bathroom would be as big as the bedrooms. She seemed to have noticed that it had enough space to allocate two bathrooms, but she was bugged by the fact that she could not expand the house into a four bedroom (58), in case the clients had the third child. Therefore, she abandoned this solution, backtracked and, redefined the functional organization of the upper floor, placing the two bathrooms on the slot that had no natural light, but turning the first bathroom into a bedroom (moves 51-56). She seemed content with this solution that had the right number of bedrooms (one for the parents, one for each of the existing children, and one for the future children).

At this point, she redesigned the lower floor to check whether it respected the grammar the rules and satisfied the clients' requirements, and to place the openings (as there were no changes to the layout, these moves are not shown). At the end of these moves, she realized that she could enlarge the transitional space at the expense of the living zone, thereby fusing the informal and formal dining areas into a single dining room, and making the living room bigger (move 58). She hesitated whether she would enlarge the pantry, because she was afraid that it would become too big, but she did it in the end. Finally, she redesigned the upper floor for similar reasons, but she did in such a way as to show how it could, in fact, expand from three (57) up to four bedrooms (55). (The full



sequence of moves is not shown because it is very similar to 51-55, except that the division of the terrace into a bedroom and a smaller terrace is done at the end.) At the end of this processes she placed the windows and doors.

After ending the experimental session, and before making the 3D model, she noticed that if she mirrored the house it would be possible to pierce openings on the side wall, thereby solving the problem of the fourth bedroom loosing access to light when building the fifth bedroom. There is no video protocol of this design moves, but there is graphic protocol diagrammed in moves 59-69 for the upper floor, and moves 70-77 for the lower floor. In this new solution the lower floor is simply mirrored, but there are some changes on the upper floor. Namely, the bedrooms and the bathrooms are bigger, and one of the bathrooms is on the top of the one in the lower floor, as she initially desired.

The first feature of C1's derivation process that becomes immediately apparent is that *she moved back and forth between the designs of the lower and upper floors*. In this sense, she did not respect the grammar, which foresaw first the derivation of the lower floor, and then, the derivation of the upper floor. Moreover, those design subjects who tried to follow this constraint, for instance C2, had difficulties in achieving a satisfactory solution, and finally gave up and worked alternately on the two floors. This result suggests that the grammar should be changed to account for such feature. For instance, not only should the grammar have rules that place labels on the second floor to signal changes on the first one, but the reverse should also be true.

The second feature is that *the derivation of the design searching for a solution is not a continuous and linear progress towards the goal*. It contains some dead ends (e.g. 25,) backtracking (e.g. 28-29,) and jumps in different directions (e.g. 60.) The question that

comes immediately to our mind is whether it would be possible to avoid such moves saving time in the derivation process. This means whether it would be possible to gather enough knowledge to make the derivation process as linear as possible. Consider, for instance, the urban context. Noticing that most of the design subjects decided to mirror the design at the end of the design process, one could be tempted to create a rule that says “if the lot has no houses on the right side, then place the patio on the left side.” This would permit to pierce openings on the right and to decrease the circulation area. However, the results show that this does not necessarily lead to solutions with the best degree of satisfaction. For instance, designs S3 and S4 ranked better than some mirrored designs. This suggests that there might be some niches in the universe of design solutions that contain good solutions that will never be reached if one introduces such rules. The tradeoff involved in such a decision is one between accelerating the generation of designs, or increasing the probability of achieving a better solution.

What is, then, the alternative? The design subjects who achieved better results seemed to have used the following strategies.

First, they tried *to satisfy the requirements that were considered more important*. For instance, C1’s first concern was to allocate the bedrooms and the bathrooms, and then, to define the spaces that formed the service zone, which ranked at the top of the priority list set by the client. (Table 6.1)

Second, they tried *to allocate the spaces with bigger demands* both in terms of area and topological requirements. For instance, the placement of the kitchen, the biggest space in the service area with a multitude of connection requirements to the other spaces in the

zone and to the other zones was instrumental to the definition of the service area in C1's process.

Third, they used *transformation rules as a way of doing "local optimization"* of the design solution. These rules worked as short-cuts in the derivation process. For instance, C1 moved from position 28 to position 32, in a single move. This move corresponds to the rule shown in Figure 6.22, which was not foreseen in the shape grammar. The grammar included other transformation rules, such as Rule 61—turn kitchen into bedroom and transitional space into kitchen (permuting functions), and Rule 132—expand a bedroom to achieve natural light. The conditions for the applications of these rules were very restricted—the number of bedrooms should be two in the case of Rule 61, and the bedroom had no windows in the case of Rule 132. It is possible to loosen such conditions so they can be used both for expanding and for contracting rooms in the process of the dimensioning of rooms.

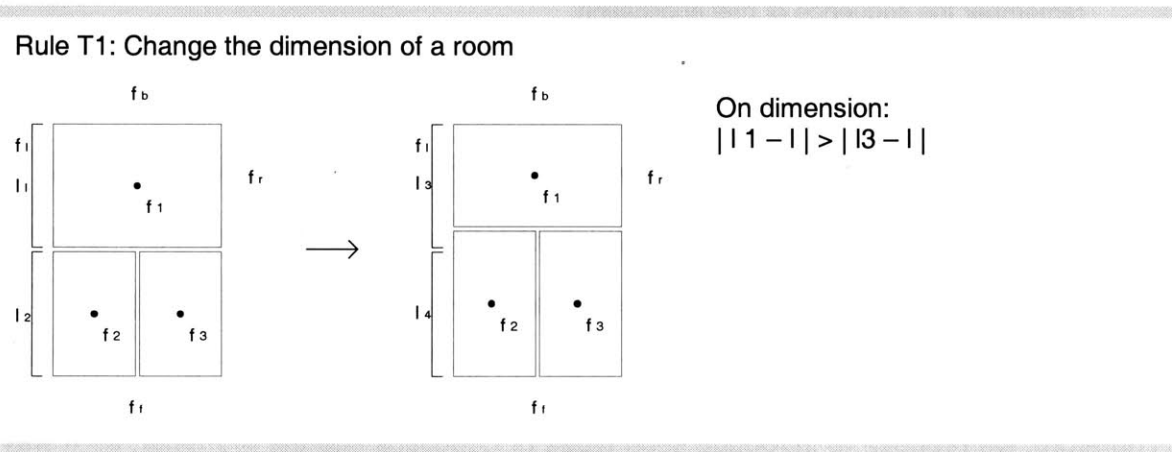


Figure 6.22 – Transformation rule used by design subject C1 in her design process.

Fourth, they did some kind of *informed backtracking*. When they moved up in the derivation tree to take a new path, this process was influenced by the knowledge acquired while traversing the initial path. Such an influence was perceived in the selection of new branches in the tree, as well in the sequence of rule application. Consider, for instance, the move from design state 33 to 34. Assessing the design solution in 33, C1 considered the kitchen too small compared to the laundry, and noticed that there was no space left for the pantry. Therefore, she backtracked, until none of these spaces was allocated and then, she re-allocated the kitchen increasing its area, followed by the pantry, and then, the laundry. This procedure can be captured into an algorithm like the following:

If space x is too small, and space y is too big, then  
backtrack to the point where none of these spaces was allocated;  
allocate the space with bigger area demands, correcting the problem;  
allocate the other spaces.

Other procedures, such as the one from design state 35 to 37, lead to simpler algorithms:

If space x is too small, then  
backtrack to the point where it was allocated;  
remember the sequence of rule application;  
re-allocate the space, correcting the problem;  
and re-apply the rules in the remaining sequence.

This type of procedures is considerably different from doing random transformations to the design solutions, hoping to correct the problem, which is the procedure used by stochastic optimization processes like genetic algorithms and simulated annealing.

Therefore, we propose to introduce such types of heuristics into the grammar, as a way of saving time, and achieving better design solutions. We will call such a grammar a discursive grammar as described in Chapter 7.

## **6.5 Experiment 4: goal-oriented design II (goal test) / collaborative design**

### **6.5.1 Goal**

This experiment simulated the situation in which a group of design teams is in charge of designing houses for various clients in the same development. Thus, some of its goals were similar to those of the previous one: to test the Malagueira grammar, especially its ability to generate criteria-matching designs. It had, nevertheless, additional goals. The first additional goal was to check whether the use of the grammar contributed to improve collaboration among different designers. It targeted both collaboration among members of the same team, and collaboration among different teams. The second goal was to verify whether the grammar provided the means to solve the following variety/unity paradox faced by each design team. On the one hand, it wanted to customize the houses as much as possible to solve the problem of their particular client. On the other, it needed to integrate the house into the whole. The third goal was to study client/designer interaction mediated by a protocol, which is instrumental to the housing provision framework envisaged in this research. In this experiment, such a protocol was provided by a questionnaire composed of key-questions regarding the house, filled by the client, and used by the designer as a basis for designing.

This experiment took the form of an intensive four-week workshop conducted between the Massachusetts Institute of Technology (MIT) in Cambridge, U.S.A., and Miyagi University (MU) in Sendai, Japan. The designers had to design a housing block composed of units, for a given set of clients, both by following the grammar rules and by changing these rules. Rapid prototyping techniques were used in the process of design. The project required students at both ends to work collaboratively through Web-based and videoconferencing technologies.

### **6.5.2 Subjects**

There were four sets of subjects in this experiment. The first set was formed by four Japanese and two North American families who volunteered to perform the role of clients. (Figure 6.22) These families were represented by one or two of its family members, who were university professors at MU (4), MIT (1), and the Chinese University of Hong Kong (1). These families were selected from a larger pool using as criteria the need to obtain varied social profiles (members, age, gender, activity, interests, lifestyle, etc.) to provide a variety of design problems.

The second set of subjects consisted of four design teams. These teams were formed by two MIT graduate students and three Miyagi undergraduate students according to the following criteria: (1) one of the MIT students had reasonable knowledge of shape grammars; (2) the other MIT student had basic knowledge of shape grammars; (3) two of the Miyagi students were architecture students with no knowledge of shape grammars; and (4) the third Miyagi student was a non-design student whose role was to work as a language assistant. None of the design subjects were knowledgeable of the architectural and cultural contexts in which the Malagueira project was developed.

The third set of subjects included the author of the grammar and Siza, who were in charge of reviewing the results to determine their stylistic fitness.

### **6.5.3 Setting**

Four experimental settings were used in this experiment. The first setting served to interview the client remotely. It included a computer with Internet connection on each end, and the following software: Picturatel (videoconference,) Netmeeting (Web-based videoconference, chat system, drawing board, and desktop and application-share,) ICQ

(alternative chat system,) Internet Explorer (Web browser,) and Camtasia (desktop recording.)

The second setting was used in remote work sessions among team members. (Figures 6.23 and 6.24) It was similar to the first setting, but it also included a document camera, a video recorder, paper with the Malagueira lot drawn over a milimetric grid, pencil, a list of rules, and a table summarizing the dimensional requirements of Malagueira houses. All these items were used for synchronous work. In addition, this setting included various e-mail applications, Web-based pin-up pages, and file transfer sites for non-synchronous work.

The third setting was used in remote sessions (lectures and presentations) attended by all the participants and it used the same devices of the previous setting. (Figure 6.25) In addition, it included wall projection equipment on each end so that all the local participants could easily see desktop and document camera images, as well as room images captured by several video cameras.

The last setting was used to show the designs to Siza. It was very similar to the one described in Section 6.4, except that it used a portable computer.

#### **6.5.4 Tasks**

The overall task of the experiment was to redesign one of the Malagueira city blocks, from which some houses had been deleted. (Figure 6.26) Then, there were specific tasks assigned to the clients, the design teams, and the reviewers. The clients had to choose a plot, and to describe the house that they needed. Then, they had to comment on how the design solutions satisfied them.

In an attempt to overcome the limitations of the previous experiment, the design teams were assigned four tasks of increasing difficulty. The first task was in the realm of abstract grammars and consisted of three parts, each constituting a short version of the remaining three tasks. The second task was to come up with the derivation of existing Malagueira designs, based on the given grammar rules. The goal of these two tasks was to give those design subjects who were not familiar with grammars the opportunity to learn the basics and become familiar with the Malagueira grammar. The third task was to design a house that satisfied the clients' requirements by strictly following the rules of the grammar. The subjects were, thus, put in the position of Siza's collaborators. The explicit request for strictly following the rules was motivated by the results of the previous experiment, and the goal was to clarify whether designers did not respect the rules because they did not know them enough, or because their design problem demanded so. The fourth task assigned to the design teams was to generate a house that satisfied the clients' requirements, but they were allowed to change the grammar rules by deleting, changing, or adding new rules, as long as they respected the building regulations defined by Siza. They were, thus, placed in the position of the designers who were not affiliated with Siza's office and had to design houses for the Malagueira development. In this fourth task, the design teams had to design a house for a new client, or to re-design the house for the former clients, depending on whether their designs had been considered satisfactory.

The author of the grammar had to verify whether the houses respected the grammar rules during the design process. Siza's task was to make the final comment regarding stylistic compliance.



### **6.5.5 Procedure**

The clients were asked to describe their desired house by filling in a form. This form was similar to the one used in the previous experiment and is presented in Appendix 1. Then, they had to attend a short interview (2-5 minutes) with the designer of the grammar, who asked them to mention the important aspects about their house that were not covered in the form, or to clarify the answers to some of the included questions. This interview proceeded through videoconference, and it was recorded.

This experiment took four weeks, each devoted to one of the tasks mentioned above. The first week intended to brief the design teams on shape grammars and it included an introductory lecture and a task in which design teams manipulated simple abstract grammars. The second week included a lecture on the Malagueira grammar, and the task was to propose a derivation for a given Malagueira house. In the following two weeks, the design teams were asked to design houses for given clients, the tasks that formed the core of the experiment. Before starting these tasks, they were given on-line access to their clients' forms and interviews, and provided with the list of rules, the area requirement table, and a plan with the location of the plots. Then, they were asked to start designing the house, using paper, pencil, and the document camera or a CAD application with the application-share feature turned on. This session was videotaped for posterior analysis. They were allowed to continue developing the houses after this session, and to show them to the client and to the author of the grammar for comments on requirement satisfaction and stylistic compliance, respectively. For communicating with the reviewers, the design teams could post drawings on the pinup page, and then use e-mail or a chat system. At the end of the week, they had to present their housing solutions to the clients through videoconference, using 2D and 3D drawings, as well as physical models produced by rapid prototyping. Also, they were required to show the

derivation of their designs, indicating the rules applied at each step, including any eventual new rules. The workshop terminated with a final presentation in which all the produced houses were gathered to form the housing block. At the end of the workshop, the design subjects were asked to fill-in a questionnaire regarding their understanding of the grammar and Siza's architecture at Malagueira.

After the workshop finished, the individual houses and the city block were shown to Siza.

### **6.5.6 Results**

The forms filled in by the clients are shown in Appendix 1. Figure 6.27 shows the assignment of clients to the available plots, and the plans of the new houses inserted into the block. The 3D digital model of the block before and after the new houses were inserted is presented in Figure 6.28. Figure 6.29 includes examples of plans used by one of the design teams to show its solution to the client, and the author of the grammar. Figure 6.30 presents the plans, sections, and elevations of the designs generated in the experiment 4 for task 3, whereas Figure 31 presents the same elements but for task 4. Figure 6.32 and 33 presents the corresponding 3d digital models. Finally, Figure 6.34 shows physical models produced at MU and MIT using different rapid prototyping techniques.

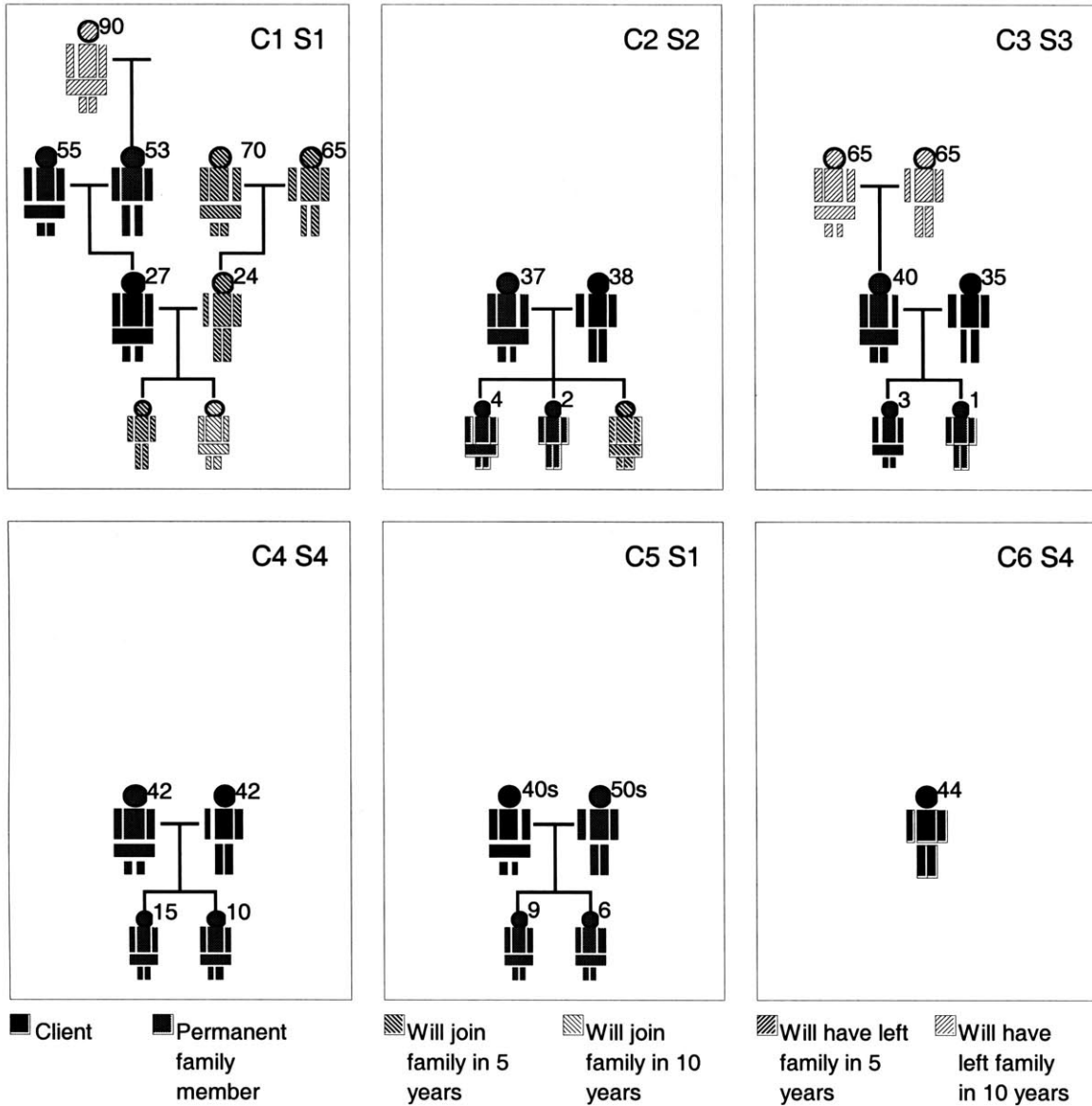


Figure 6.22 - The families of the clients who participated in the experiment. The clients are identified by the letter C, followed by a number, whereas the designers of their houses are identified by the letter S, also followed by a number. In each frame, each level represents a generation; in top-down fashion: great grandparents, grandparents, parents (the client's generation,) and children. The numbers next to family members indicate their ages.

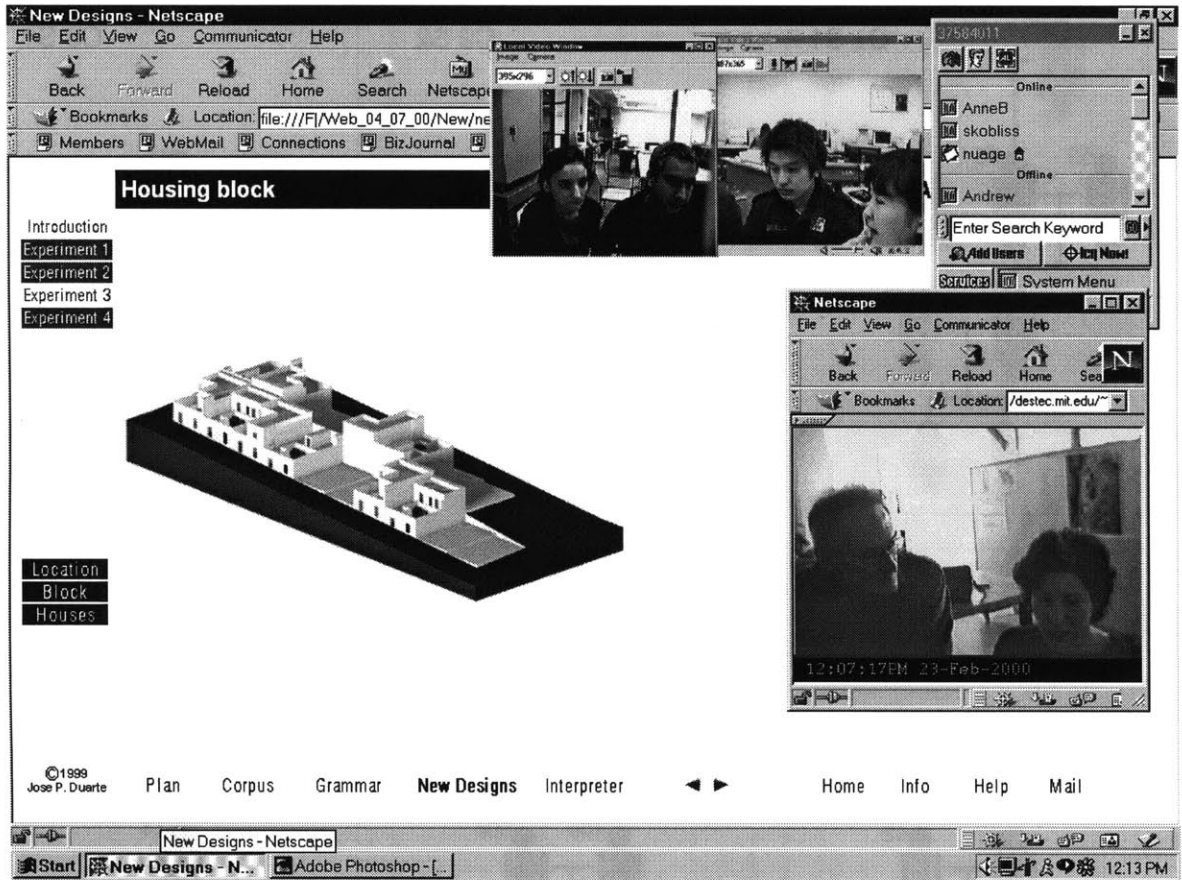


Figure 6.23 – Work session with members of the same design team at MIT and MU communicating through videoconference and a chat system (top,) while listening to the interview with the client (bottom right,) and looking at the Web page with site information (background.)

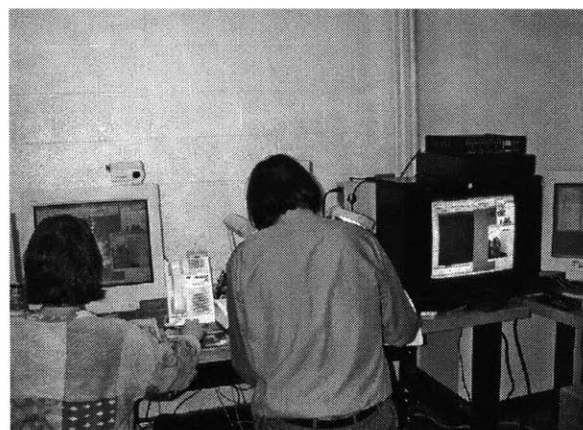
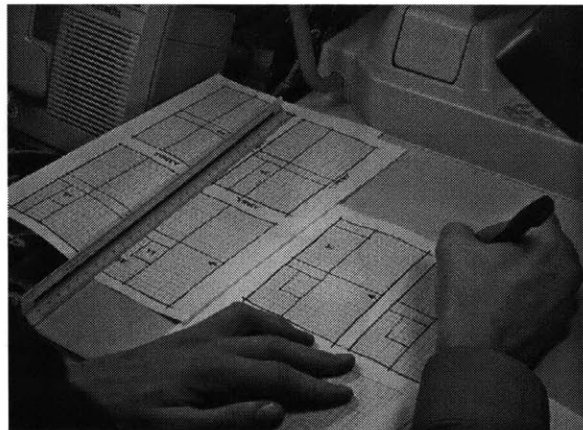


Figure 6.24 – Snapshots of work sessions with videoconference at MIT showing the design subjects working on the derivation of their houses together with their Miyagi teammates through the document camera. The video recording set up is shown on the bottom right image.

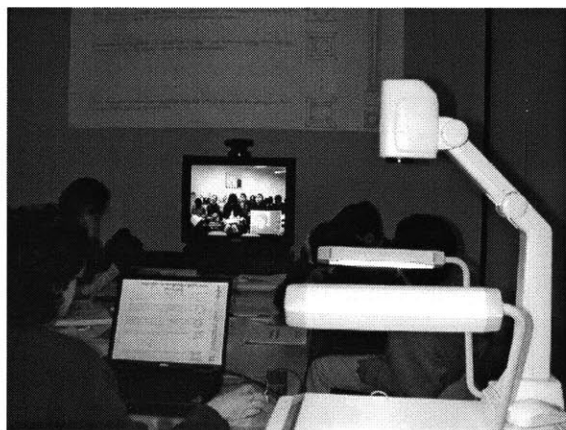


Figure 6.25 – Snapshots of sessions attended by all the participants: lecture (left,) and presentation (right.)

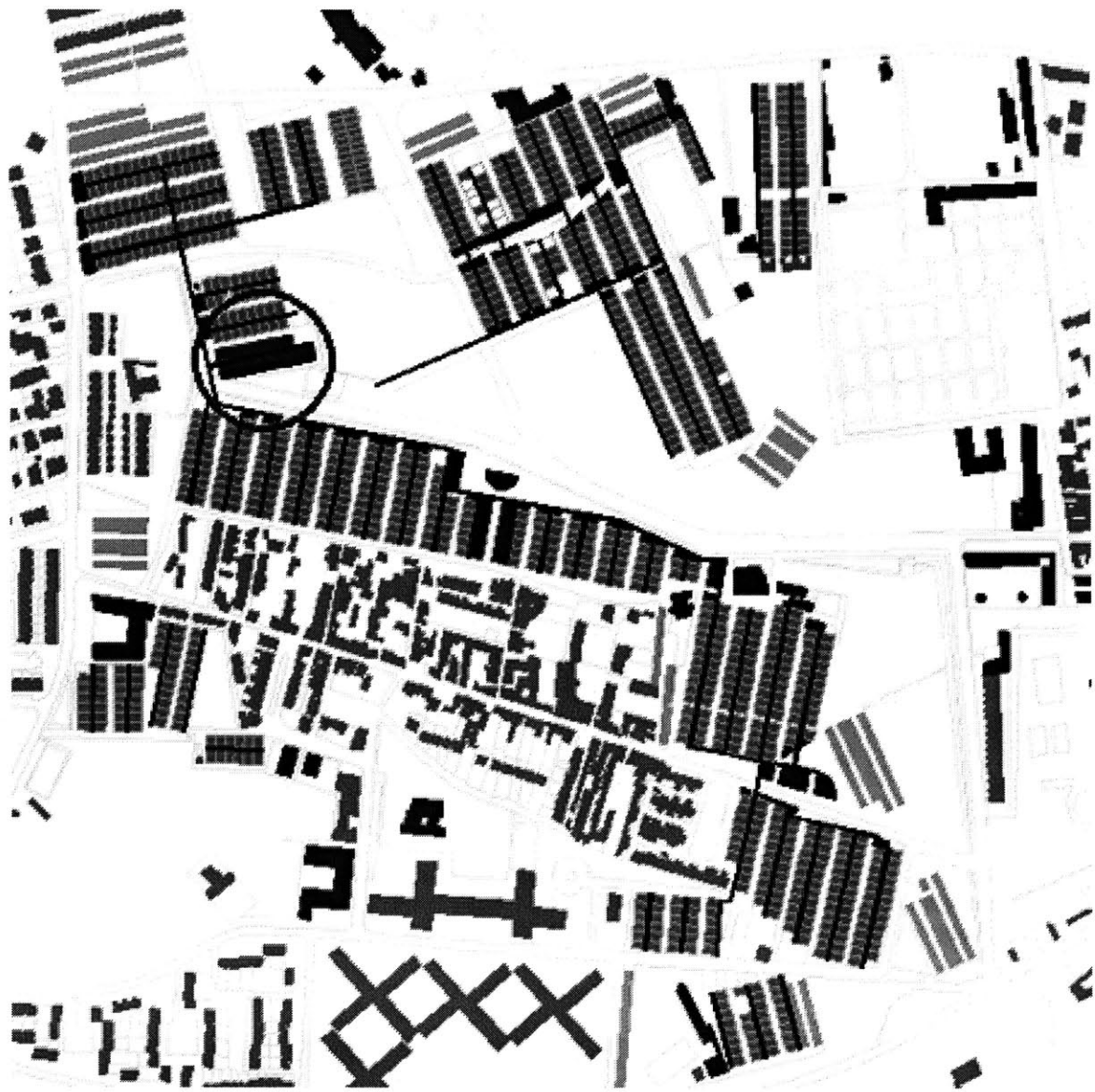
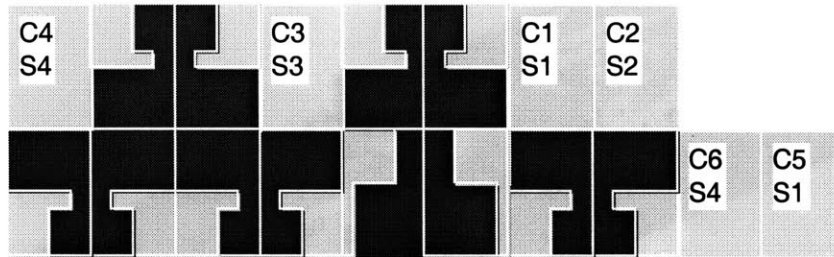
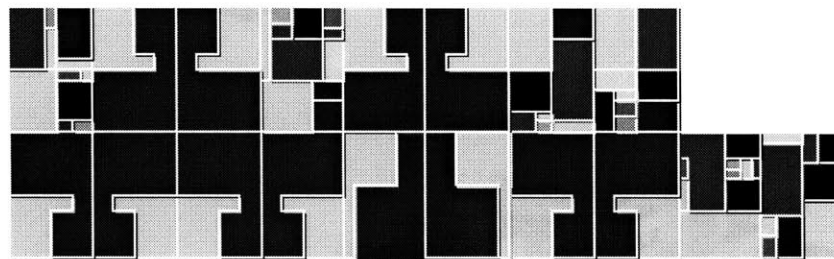


Figure 6.26 - The location of the Malagueira city block redesigned in this experiment.

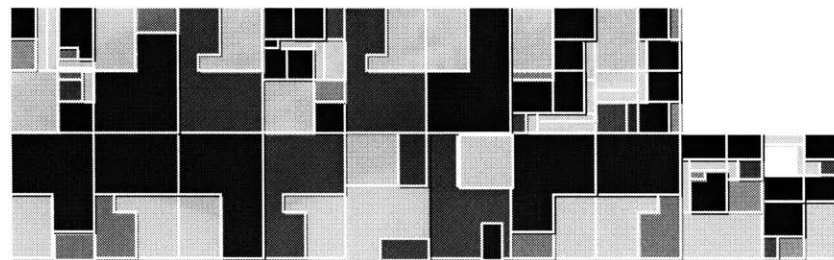
Before



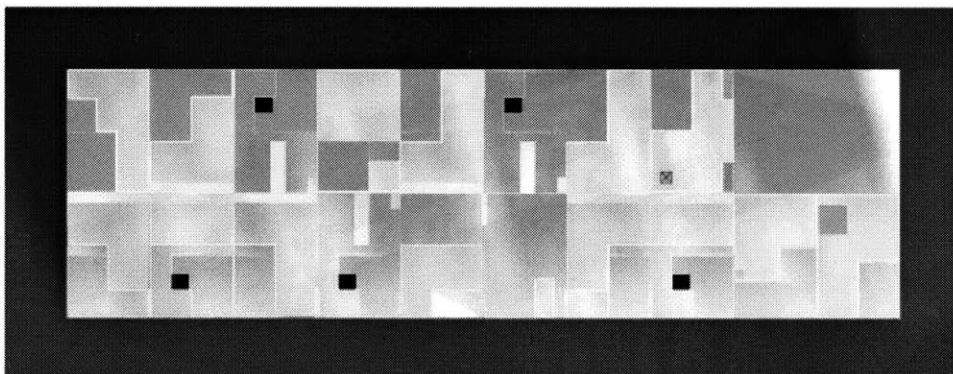
After



1st floor



2nd floor

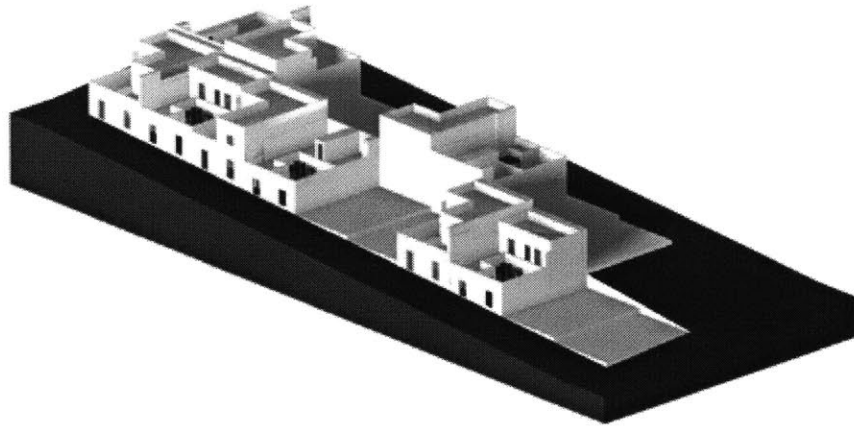


Terrace

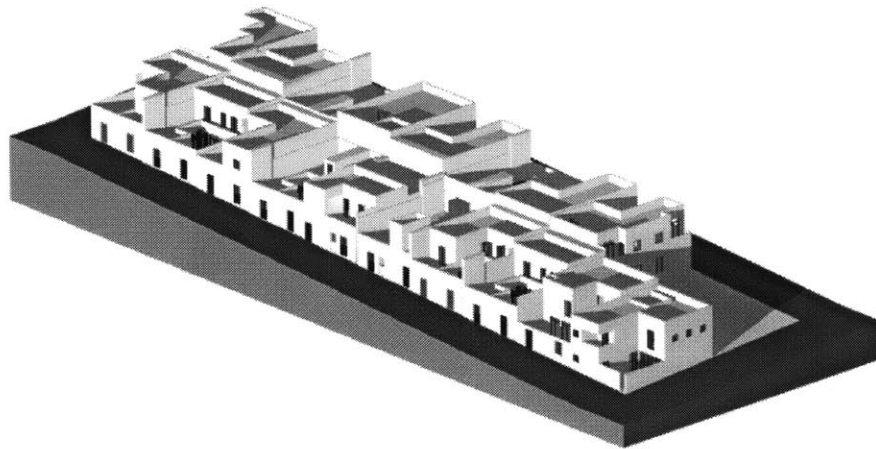
Figure 6.27 - The plan of the city block before and after the new houses were introduced. The top plan also shows the assignment of clients to plots.



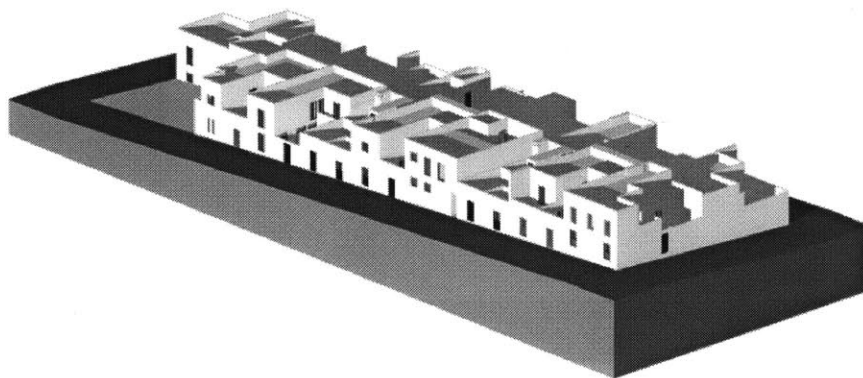
Before



After



South



North

Figure 6.28 - The 3D digital model of the city block before and after the new houses were inserted.



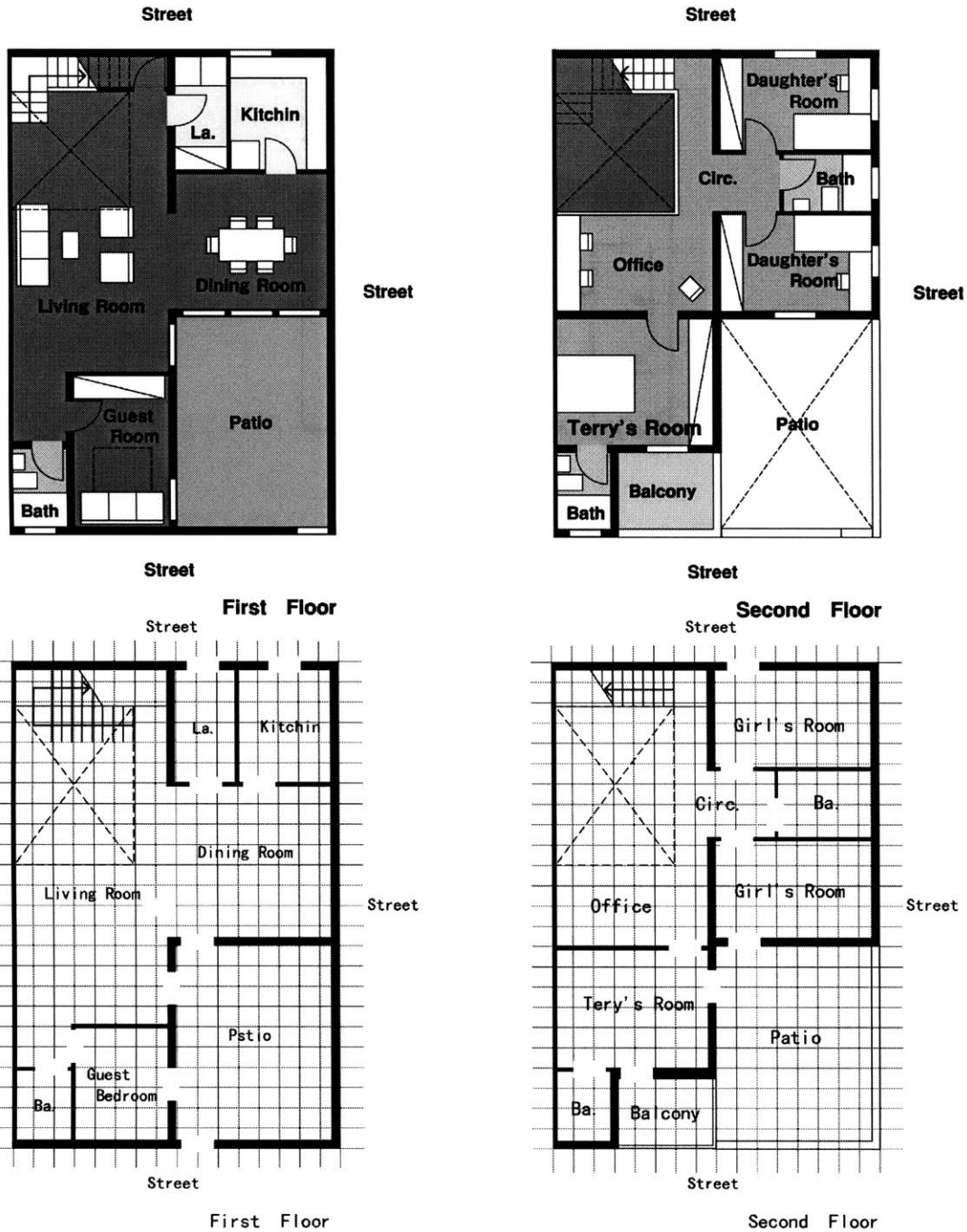
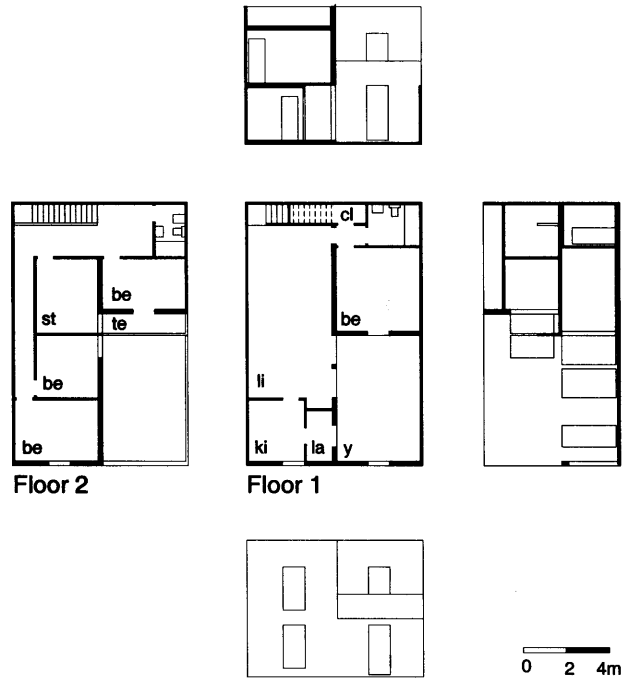


Figure 6.29 – Floor plans used by one of the design teams to communicate the solution to the client (top,) and author of the grammar (bottom.)



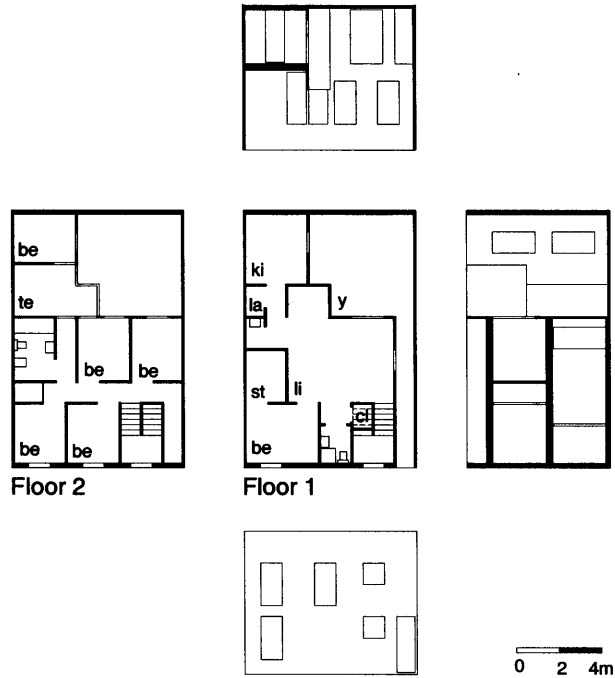
C1 S1 t4



C2 S2 I t4

Figure 6.30 – Plans, sections and elevations of the designs generated in Experiment 4, task 3 (respecting the rules). Design C2 S2 I was considered non-satisfactory (see text.)

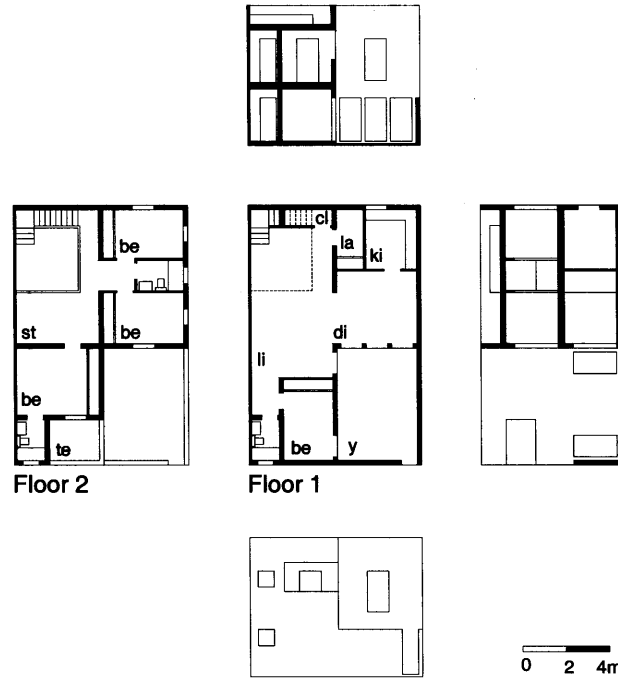
C3 S3 t6



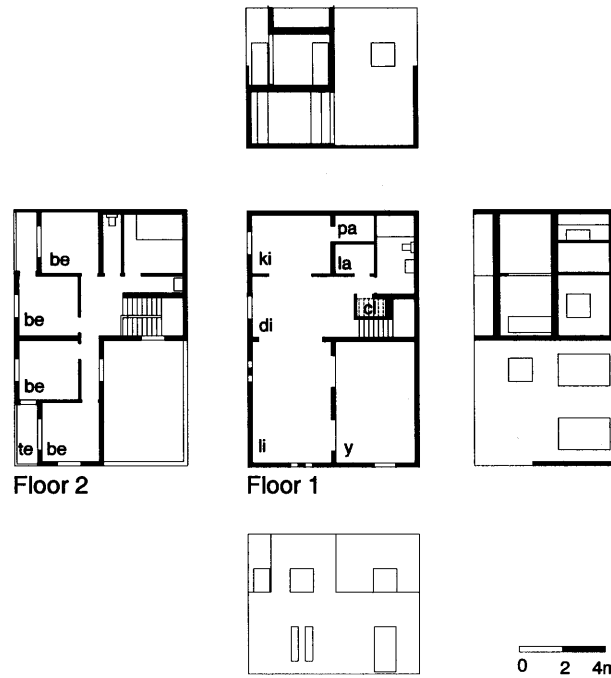
C4 S4 t4



Figure 6.30 (continued) – Plans, sections and elevations of the designs generated in Experiment 4, task 3 (respecting the rules). Design C3 S3 I was considered non-satisfactory (see text.)

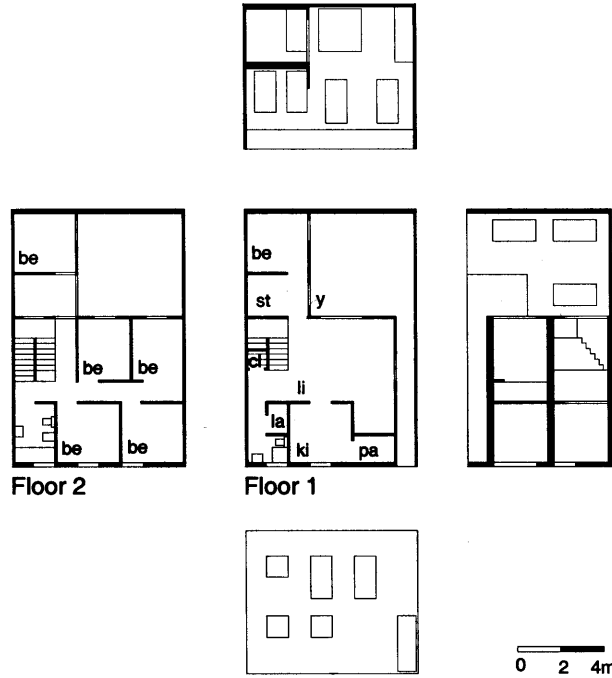


C5 S1 t3

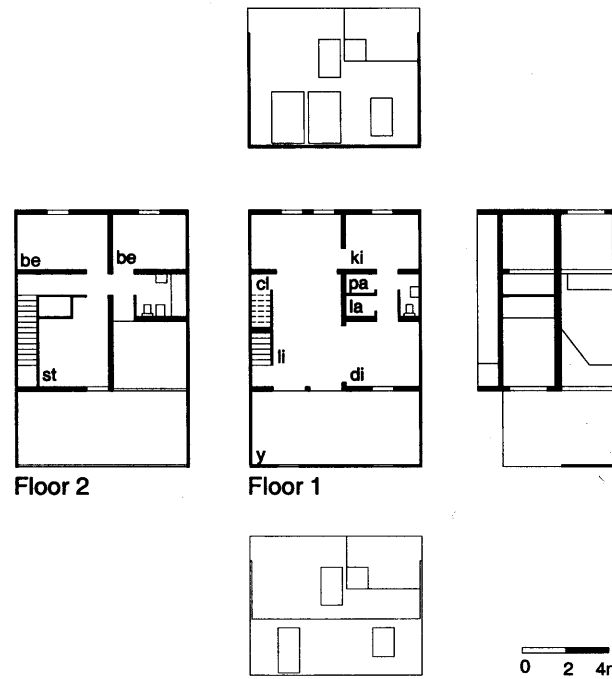


C2 S2 II t3

Figure 6.31 – Plans, sections and elevations of the designs generated in Experiment 4, task 4 (changing the rules).



C3 S3 II t6



C6 S4 t2

Figure 6.31 (continued) – Plans, sections and elevations of the designs generated in Experiment 4, task 4 (changing the rules).

## New designs (different clients, different authors) Malagueira - Alvaro Siza Vieira

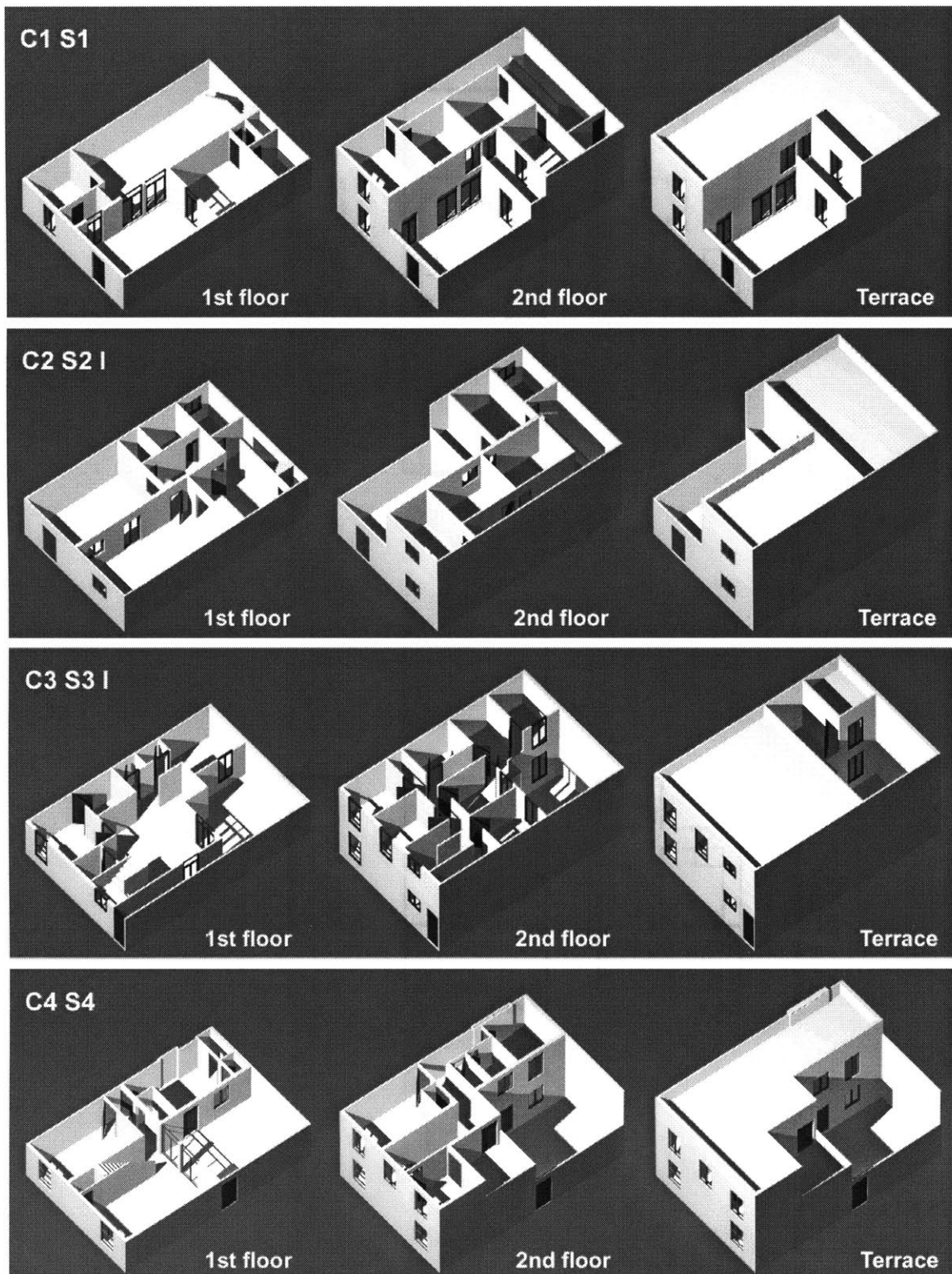


Figure 6.32 – Digital models of the designs generated in Experiment 4, task 3 (respecting the rules.) Designs C2 S2 I and C3 S3 I were considered non-satisfactory (see text.)

# New designs (different clients, different authors) Malagueira - Alvaro Siza Vieira

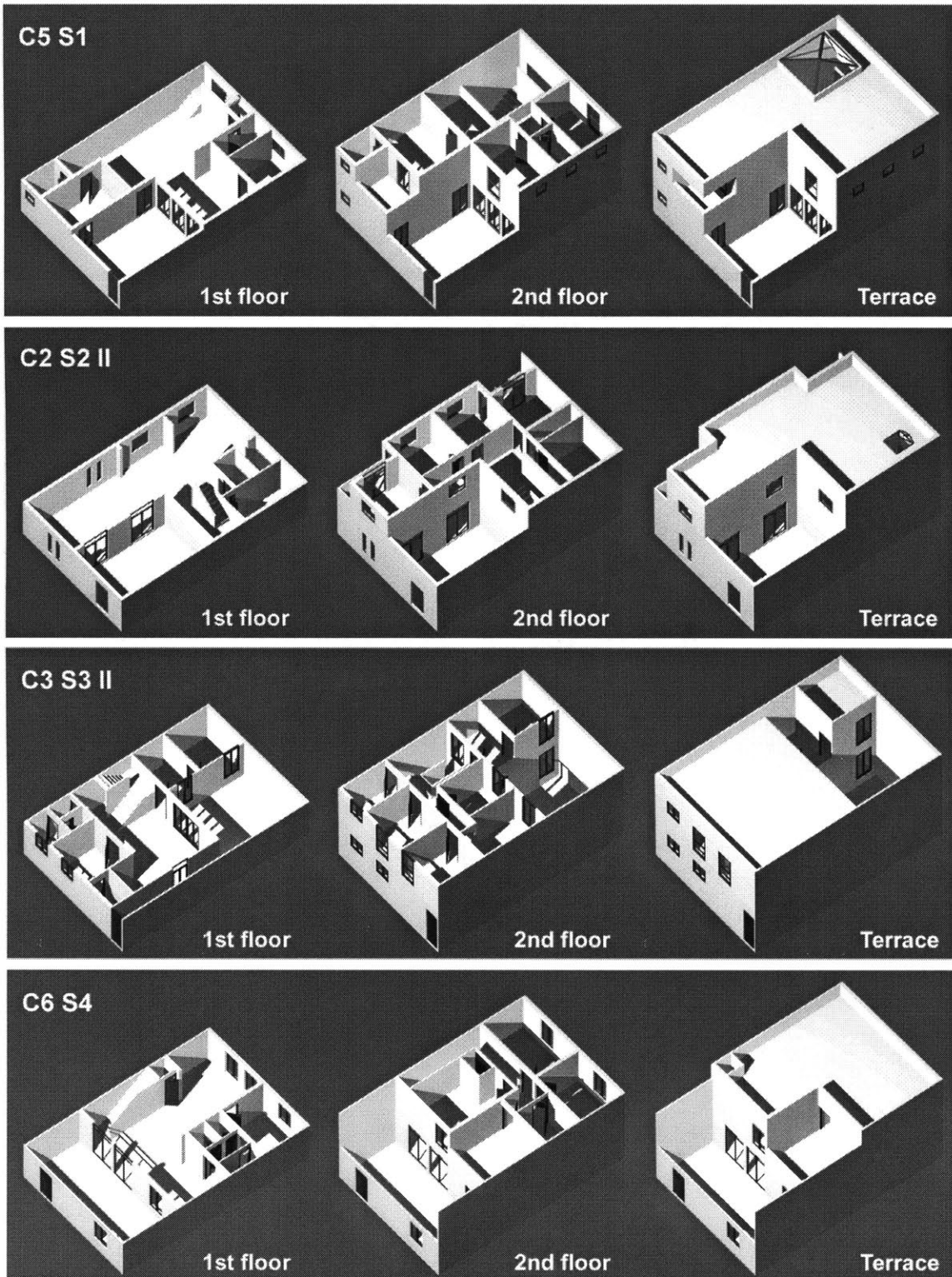
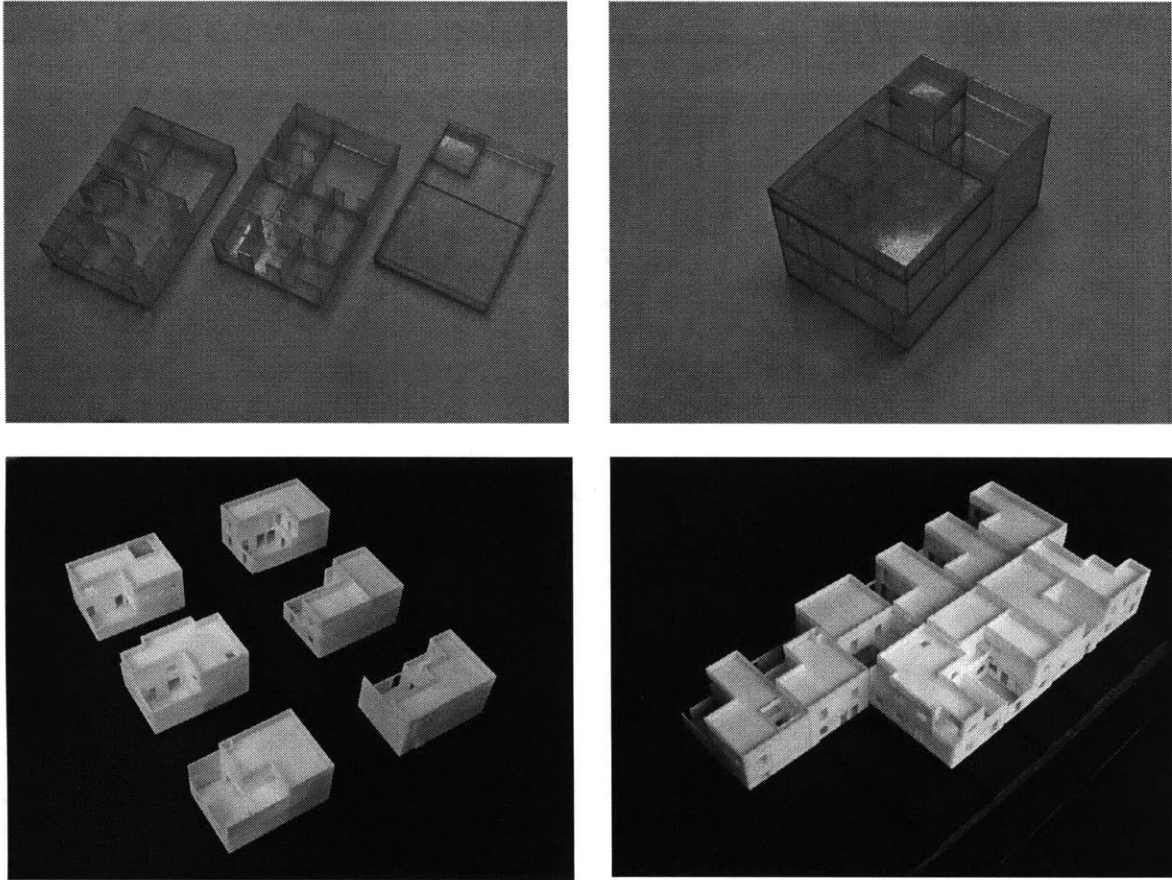


Figure 6.33 – Digital models of the designs generated in the Experiment 4, task 4 (changing the rules.)





S

Figure 6.34 – Physical model of a house generated in the experiment produced by stereolithography at MU (top) and physical models of all the houses and the city block produced by Fused Deposition Model at MIT (bottom.)



### 6.5.7 Discussion

*Analytic test: are the designs in the language?*

Similarly to the previous experiment, we relied on two procedures to find the answer to this question. The first procedure was to ask Siza, and the second was to analyze the derivations of the new designs to check whether rules had been properly applied. Siza's comments on designs produced in the third task, when designers were asked to follow the grammar, were similar to those on the results of the previous experiment. Some of the designs disrespected the rules for dimensioning the patio, and the rules for placing and dimensioning openings. The disrespect for the patio dimensioning rules was, nevertheless, in accordance with our decision to allow the patio to be smaller on the second floor, by eliminating the verandah, as explained in Section 6.4.7. The only exception was design C3 S3 I, whose designers made the patio smaller on the first floor, as well. The disrespect for the opening rules existed in the number, location, and size of openings. In addition, Siza mentioned that the proportions of some rooms "did not seem right." When asked to clarify this aspect, Siza said that he liked to give the rooms certain "ideal" proportions such as 1:1, 1:2, 3:4, etc. When we called his attention that in the design of types A and B, the use of such proportions was evident, but that in the remaining it was not so, Siza explained that the need to create varied housing programs had constituted a constraint. When we suggested that the same could have happened in the new designs, Siza said that the use of non-ideal proportions was acceptable, but he preferred designs that used them, anyway, suggesting that the grammar should incorporate such a preference.

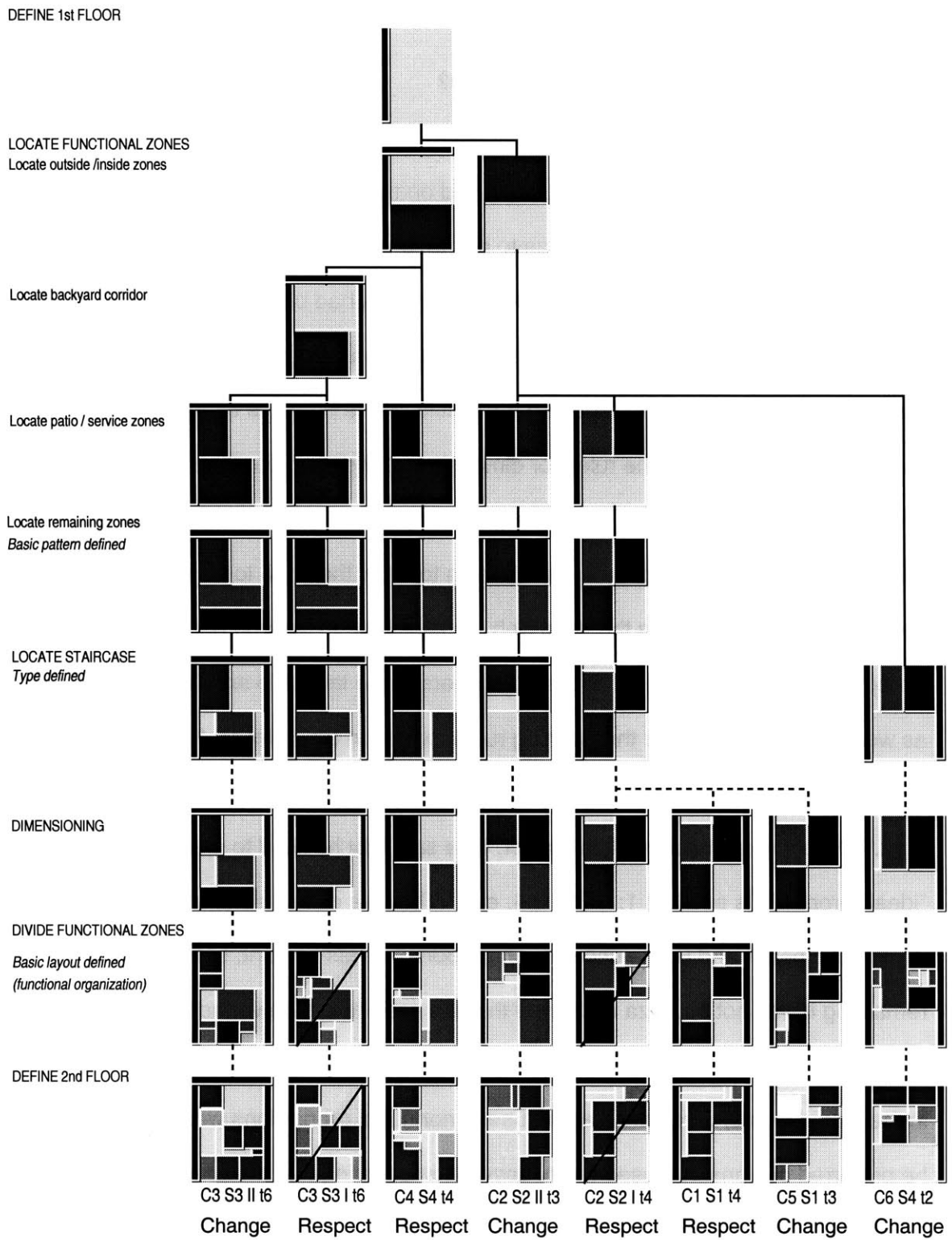


Figure 6.35 - Partial tree diagram showing the derivation of the designs produced in Experiment 4, tasks 3 and 4. (Compare with the ones in Figures 5.9, 6.6, and 6.18.)

The analyses of the derivations, summarized in Figure 6.35, confirmed the disrespect for the patio dimensioning rules in design C3 S3 I, and the disrespect for the opening rules in the majority of the designs, although to a much lesser extent than in the previous experiment.

Analysis also revealed other problems with design C3 S3 I, such as a studio with no windows. It also showed that need for allocating a large piano required by the client, and the choice of stair patterns available within the grammar were behind the flawed design. Because the client chose a backyard patio, and because the grammar in the strictest interpretation of Siza's rules did not allow the placement of a U-shaped staircase in the living zone, the design team decided to use the same stair pattern of Siza's type B. However, the location of the staircase next to the façade in this pattern, required the design team to carve the studio out of the living zone, next to the wall between the house and the neighboring one. This caused the studio to have no windows, and the piano to be allocated by carving space out of the patio. The client did not accept the solution, and the designers were asked to re-design the house in the fourth task. They solved the problems in the new design (C3 S3 II) by permuting the location of the service and sleeping zones, and by creating a rule for allocating the staircase in the living zone, next to the wall between the two houses.

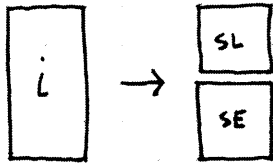
Analysis also showed that in design C2 S2 I the living room had now windows and was too small, whereas the dining room was too big. The topological requirements specified by the client, and the choice of basic patterns available within the grammar, in the strictest interpretation of Siza's rules, hampered the possibility of achieving a satisfactory solution. Because the dining room needed to be connected to the kitchen, the designers enlarged the dining room at the expense of the living room. However, as the patio was

located on the side with no houses and the living zone was diagonally opposite to the patio in the available patterns, such an enlargement caused the living room to have no windows. The client considered the design non-satisfactory, and the designers were asked to re-design the house in the fourth task. They solved the problems in the derivation of the new design (C2 S2 II) (Figure 6.36) by shifting the location of the patio, and by creating a rule for locating the living zone next to the patio, and a rule for locating the service zone next to the sleeping zone. (R(new 1) and R(new2), Figure 6.37)

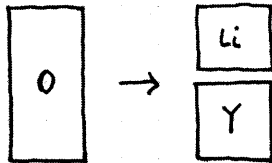


Figure 6.36 - The derivation of design C2 S2 II presented by its design team in the final presentation. Rule numbers do not match those in Figure 5.19 because the rule numbering system was changed after the experiment.

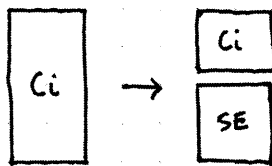
R (new 1): Dissecting the inside zone into sleeping and service zones.



R (new 2): Dissecting the outside zone in living and yard zones.



R (new 3): Dissecting a circulation room to expand the service zone.



R (new 4): Adjusting the thickness of the bathroom wall to install a large bathtub.

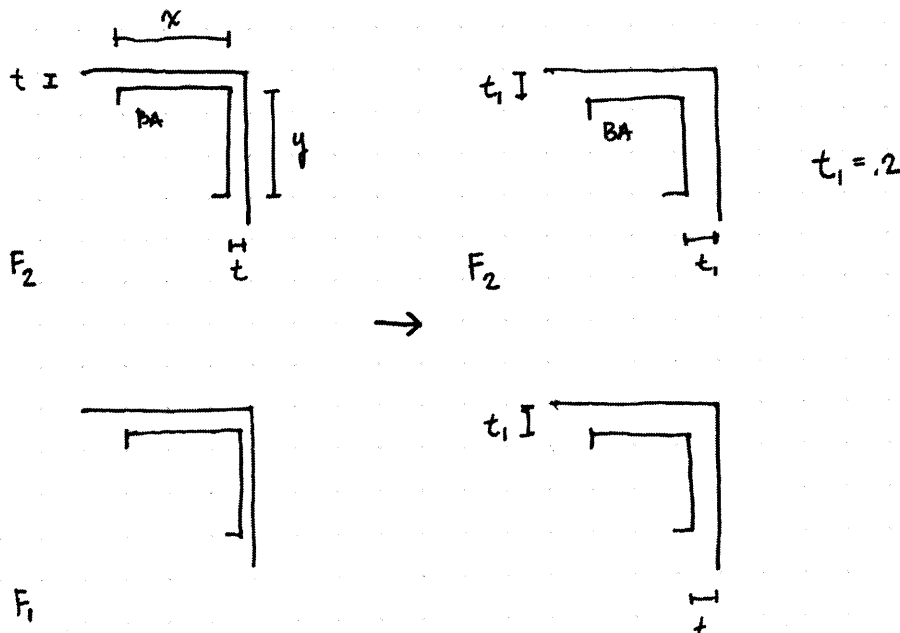
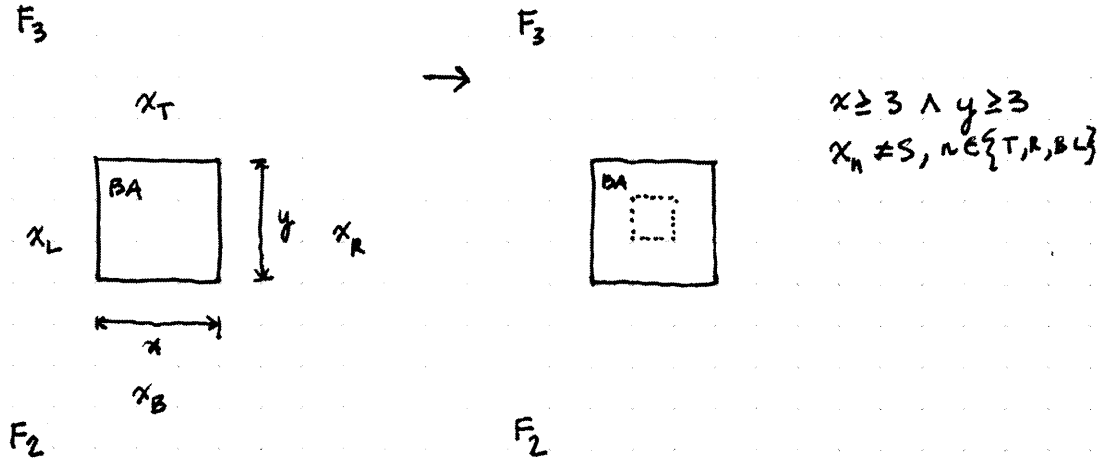


Figure 6.37 - Some of the rules introduced in the derivation of design C2 S2 II, in the fourth task.

R (new 5): Pierce a skylight above the bathroom.



R (new 6): Pierce two parallel thin windows on the living room wall.

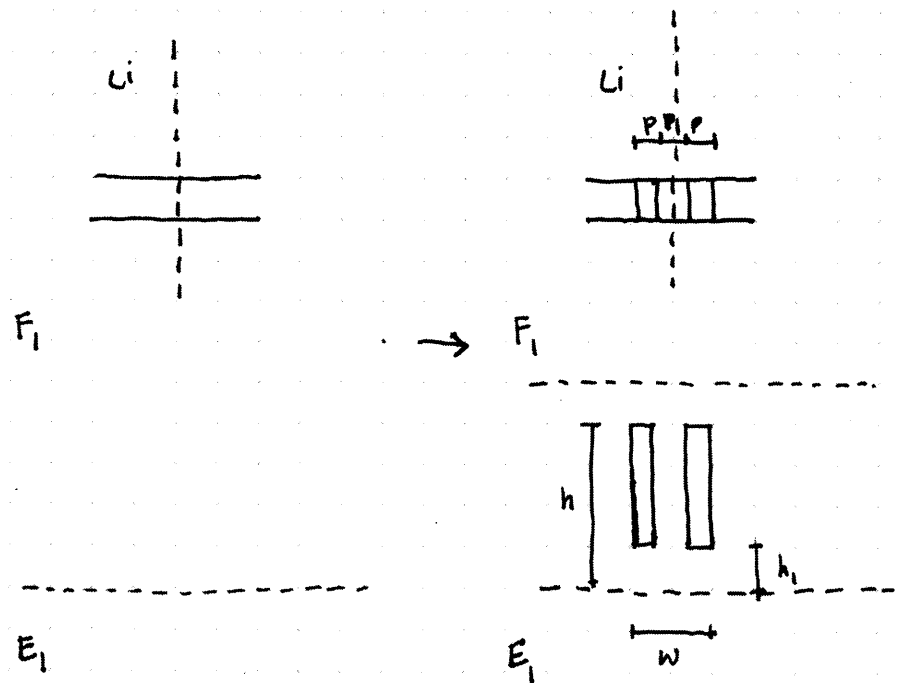


Figure 6.37 - Some of the rules introduced in the derivation of design C2 S2 II, in the fourth task.

Therefore, the analyses of designs C2 S2 and C3 S3 derivations revealed the conflict between the need to satisfy the client's requirements, and the need to respect the pattern generating rules in the third task, leading designers to change them in the fourth task. Siza accepted the new designs, which suggests that the universe of design solutions could be enlarged to encompass the patterns that can be inferred from a broader interpretation of Siza's rules. (Figure 5.12)

In addition to the rules for creating new patterns just mentioned, designers introduced other rules in the fourth task. Some of these rules were prompted by idiosyncratic formal preferences on the behalf of the designer, such as the rules for piercing two vertical, thin windows next to each other; (Design C2 S2 II, R(new6)) and a rule for creating an overhang over a terrace by not piercing the slab. (Design C5 S1) However, the great majority was motivated by the need to satisfy functional requirements specified by the client. Namely, they introduced: a rule for creating skylights; (Design C2 S2, R(new5), Design C6 S4, Design and C5 S1) a rule for checking the lower floor structure after introducing a wall on the upper floor; (Design C5 S1) a rule for piercing the slab of an internal room to create a double height room; (Design C5 S1) an assignment rule for creating a terrace between the core of the house and another interior room; (Design C3 S3) a rule for adjusting the wall of a bathroom to introduce a jacuzzi; (Design C2 S2 II, R(new4)) and a rule for creating a laundry next to a bathroom. (C2 S2 II)

The introduction of such rules confirmed the conflict between the need to satisfy the clients' requirements and the need to respect the grammar. The conflict was largely motivated by the cultural difference between the origin of the client (Japanese) and the context for which the grammar was developed (Portuguese). For instance, in the

Japanese tradition the bathroom is located next to the bathroom, whereas in the Portuguese tradition it is next to the kitchen. Interestingly enough, the introduction of such rules did not imply radical changes to the grammar, but it was achieved by changing the conditions for the application of existing rules. Moreover, Siza seemed to accept such changes, because they did not cause visible stylistic discrepancy. This result confirms results of the previous experiment that suggested the need to change the grammar to increase the possibility of client satisfaction, and the possibility of making such changes within the same stylistic framework. As discussed in Section 6.4.7, these changes should allow the generation of housing programs not foreseen in the initial grammar. Considering that to determine beforehand all the possible combinations of functional requirements is very difficult, the solution is to re-write the rules as a general algorithm to generate spatial configurations based on given functional requirements.

<ol style="list-style-type: none"><li>1. Difficulties in using the grammar for designing:<ol style="list-style-type: none"><li>a) Was it difficult?</li><li>b) How difficult it was?</li><li>c) What was difficult?</li><li>d) Which information did you think was missing?</li></ol></li><li>2. Limitations caused by the grammar:<ol style="list-style-type: none"><li>a) Were there limitations caused by the grammar?</li><li>b) What were they?</li><li>c) Were such things prompted by the client's requirements or by your personal style?</li></ol></li><li>3. What did you learn about Siza's architecture at Malagueira?</li></ol>
--

Figure 6.38 - The questionnaire presented to design subjects at the end of the workshop.

*Descriptive test: does the grammar explain the common underlying features of designs?*



Once more, we relied on two strategies to answer the above question. The first strategy was to ask design subjects what they thought, and the second was to consider their performance. The list of questions presented to design subjects at the end of the workshop is shown in Figure 6.38.

The design subjects indicated two difficulties in working with the grammar. The first was to understand the rules, and the second was to apply the rules. They indicated the high number of rules, the symbolic notation used to specify the conditions for rule application, the lack of a "procedural clarity," and the short time available as the major reasons behind their understanding difficulty. However, most of them considered that they eventually had overcome this difficulty. They were then faced with the rule application difficulty and indicated the high number of rules and conditions, and the lack of an automated engine to find the rules that could be applied at each step in the derivation, as the biggest hurdles.

The design subjects confirmed the existence of two limitations linked to the use of the grammar. The first limitation constrained the expression of the designers' idiosyncratic formal preferences, and the second limited possibility of satisfying some of the client's functional requirements. Nevertheless, they acknowledged that the grammar helped them to structure their decisions during the design process, and that it taught them about Siza's work at Malagueira, including how to generate a house in the style. Not surprisingly, the subjects who were not architects or architecture students (the language assistants) were even more enthusiastic about the use of the grammar by stating that it had taught them a lot about architecture. As one of the architects acknowledged, "the rules do not require a trained designer to generate an acceptable outcome." (Member of Team S2)

The opinions expressed by the design subjects are in accordance with the analysis of their designs discussed above. Furthermore, Siza stated that "these houses are much better than most of the houses designed by other designers [non-affiliated with his office] at Malagueira," who only followed the building regulations. Therefore, the grammar succeeded in explaining the essential underlying features of Malagueira houses, thereby overcoming the descriptive test.

*Synthetic test: does the grammar tell how to generate new designs in the style?*

Considering that designs generated by the design subjects were, to a considerable extent, in the Malagueira style, it is reasonable to accept that the grammar also overcame the synthetic test.

*Goal test: do the designs meet the given requirements?*

As discussed above, results of the third task suggested that some designers did not succeed in satisfying the clients' requirements because they were asked to respect the grammar rules. Moreover, results of the fourth task showed that such design subjects succeeded in satisfying their clients when they were allowed to change the grammar. On the other hand, Siza seemed to accept such changes when they did not cause stylistic discrepancy. Therefore, experimental results showed the need for changing the grammar to allow the generation of functional features not foreseen in Siza's initial designs, and confirmed the possibility of making such changes while maintaining stylistic coherence. The first proposed change is to enlarge the universe of solutions by incorporating all the design patterns that can be inferred from Siza's designs, including those that Siza did not use. The second change is to re-write the

rules in a general format to diminish the number of rules, to highlight the algorithmic nature of Siza's approach to the Malagueira design problem, and to permit the satisfaction of spatial configurations based on user requirements. The variety of the designs generated in the in Experiment 4 confirms the potential of the grammar to satisfy varied requirements if such changes are incorporated. In conclusion, the designs meet the requirements, but it advisable to change the grammar to increase the possibility of satisfying diverse requirements.

*Did the grammar contribute to improve collaboration among designers?*

We rely again on the design subjects' opinion and on the analysis of their design processes to answer this question. The designers' opinion can be summarized as follows. First, they stressed the need to have a solid understanding of the grammar before collaboration could take place and mentioned that they were too concerned with learning the grammar in the workshop to take effective advantage of its eventual collaborative potential. Second, they acknowledged such a potential. As one designer put it: "Since the basic rules are already established by the grammar, it provides a good platform to begin collaborative design. Much of the ground is already covered, value judgements are already made, and one can focus on finer points." (Member of Team S1)

The analysis of the design processes confirms the potential of using grammars for collaborative design. The design teams took advantage of such potential in different degrees, depending on the working strategies that they adopted. Some teams used a strong division of labor with some members generating the plans, others making the 3D-model, and other preparing the presentations. On a first glance, these teams took less advantage of the collaborative potential of grammars. However, results suggest that

their division of labor was successful exactly because the use of the grammar limited conflict. When they had to evaluate their design before switching shifts, their discussion was focused because the grammar made decisions less arbitrary, diminished the importance of authorship, and focused the discussion around the satisfaction of user needs. Thus, they could easily come to an agreement. Other teams followed a weaker division of labor with all its members involved in each task. The role of the grammar in limiting conflict was even more useful in these cases. Discussion was centered on what existing rules permitted, on which rules should be used, on how they should be applied to satisfy the clients' requirements, or on what rules needed to be created. In conclusion, independent from the working strategy adopted by a design team, the grammar provided the common thread that guided its members through the design process.

*Did the grammar contribute to solve the variety/unity paradox?*

Collaboration among the different design teams was low. In fact, interaction among members of different teams was restricted to the exchange of information regarding the location and size of the yard, and the number of floors in their houses. However, by looking at the 3D model of the city block, it is reasonable to state that it possesses stylistic unity. There are no striking differences among the different houses in terms of color, proportions, the size and location of openings, or in any other visible stylistic aspect. On the other hand, it does present some formal variety. Moreover, the houses that form the block are tailored to their users, and they were designed by designers with varied backgrounds. Therefore, it was possible to achieve a balance between the satisfaction of different user requirements and a formally coherent urban environment. We argue that balance was possible because the grammar provided a formal protocol, a

common architectural language that permitted the expression of individual requirements without jeopardizing the whole.

*Did the questionnaire provide the means to mediate the client/designer interaction?*

The analysis of experimental results show that the use of the questionnaire was useful but insufficient, as explained below.

Results show that *the design problem often was over-constrained*. It was over-constrained because the client specified too many requirements to satisfy within the Malagueira framework. For instance, the area to allocate exceeded the available area in design problem C1 S1. The problem also was over-constrained because the client specified contradictory requirements. For instance, the client wanted a sunny backyard house, in a lot surrounded by houses on three sides in C1 S1. The approach used by design subjects to solve the over-constrained problems was to talk to the client, proposing alternative solutions:

“Hi, this is your client [C1 S1]. First of all, thank you for your design in spite of my tough request. I like it very much [with] only one exception. Could you connect the bathroom directly to the Grandma's bedroom? It will be easier for Grandma and somebody who helps her to access ... it. About [the] yard, I agree with your idea; in this case, front-yard looks much better than backyard because of sunlight, wind, and other environmental aspects. I got you.”

Experimental results also show that *the problem often was ill-defined*. It was ill-defined because clients did not have a very clear idea of what the needs were until a solution was seen:

“Hello! I just took a look at “my” house. It is looking good, though of course now that I see it I have second thoughts about my requirements.”

The designers approach such cases was to go through a design-show cycle with the client, until the solution eventually became stable.

Results also showed that, even when *the problem was not over-constrained or ill-defined, designers made qualitative judgments* about the requirements set by the client. For instance, in design C6 S4, designers deliberately chose to connect the dining room to the patio, instead of to the kitchen, although the client had specified otherwise and both were possible.

Therefore, results suggest that the interface between the client and the designer should support a *dynamic interaction between the client and the designer*. Namely, it should announce that the problem becomes over-constrained when the design brief is being specified, for instance, by telling the client that the available area has been exceeded. It should also provide the means for the client to assess a solution, to change the requirements, and to generate a new solution. Ideally, it should also enquire the client, when the design is being generated, although this seems more difficult to achieve.

In Experiment 4 we tried to measure the satisfaction of clients with their houses using the concept of degree of satisfaction proposed in Experiment 3. Recall that degree of satisfaction is the ratio between the number of satisfied requirements over the total of requirements. To count the number of requirements it is necessary to decompose the requirements specified by the user into atomic requirements. For instance, the requirement “a large living room communicating with the yard through a door” yields the atomic requirements “living room,” “large living,” “living room communicating through a door with yard,” and “yard.” To find out which requirements are satisfied one just has to answer the following questions. “Is there a living-room?” “Is it large?” “Does it communicate through door with the yard?” “Is there a yard?” However, unlike in

Experiment 3 in Experiment 4, we did not consider the ranking of requirements because the clients did not rank them.

Table 6.3 – Requirements and satisfaction of designs in Experiment 4

	C1 S1	C2 S2 I	C3 S3 I	C4 S4	C5 S1	C2 S2 II	C3 S3 II	C6 S4
Requirements	131	48	50	44	39	48	51	24
Satisfaction	55.7%	93.7%	80.4%	81.8%	84.6%	97.9%	86.3%	95.8%

The use of the degree of satisfaction to evaluate the designs in Experiment 4 showed some of the difficulties in using such an evaluation system, but suggested how they can be overcome. Table 6.3 presents the number of requirements, and the degrees of satisfaction in Experiment 4. The table shows that satisfaction tends to drop as the number of requirements increases. In other words, the higher the number of requirements, the smaller the possibility of achieving a higher degree of satisfaction. Therefore, it is not appropriate to use the degree of satisfaction to compare solutions to different design problems. For instance, the degree of satisfaction of client C1 was much smaller than that of client C2, but C1 showed more satisfaction than C2, when they evaluated their houses. In addition, the degree of satisfaction might be high but the solution might be unacceptable. For instance, design C2 S2 degree of satisfaction is 93.7%, but it includes a room with no windows.

Nevertheless, results show that it is possible to compare the degrees of satisfaction of different solutions for the same design problem. For instance, the degrees of satisfaction of designs C2 S2 II and C3 S3 I are higher than those of designs C2 S2 I and C3 S3 II, respectively, which corresponded to client perception. Therefore, it is possible to use the degree of satisfaction to guide the generation of a solution to a given problem.

## 6.6 Summary

A set of experiments were devised and undertaken with the goal of generating new designs and testing the grammar. Results of Experiment 1 (analytic test) showed that the grammar could successfully account for the generation of a design by Siza, not included in the original corpus, with the exception of features related to a change in plot shape. Moreover, results showed that it was possible to explain such features with the introduction of rules to deal with the new plot. Results of Experiment 2 (synthetic test) showed that the grammar could be used in the random generation of new designs in the language. A set of designs produced in this way was shown to Siza who did not distinguish them from his own designs. Results of Experiment 3 and 4 (goal test) showed that the grammar could be used by designers not familiar with Siza's work to generate designs in the language that matched given requirements. Nevertheless, results suggested some changes to increase the possibility of generating customized designs. These changes include unrestricting the rules to enlarge the universe of design solutions, and re-writing them as an algorithm to generate spatial configurations based on given functional requirements. Results of Experiment 4 also showed that the use of the grammar by different designers could guarantee a balance between the satisfaction of individual requirements and a formally coherent whole in the design of urban environments. Also, results showed that design problems are over-constrained and ill-defined and the need to support dynamic interaction with the client to overcome these difficulties. In Chapter 7, we will see how changes suggested by experimental results are incorporated into the grammar.



# New Designs

Malagueira - Alvaro Siza Vieira

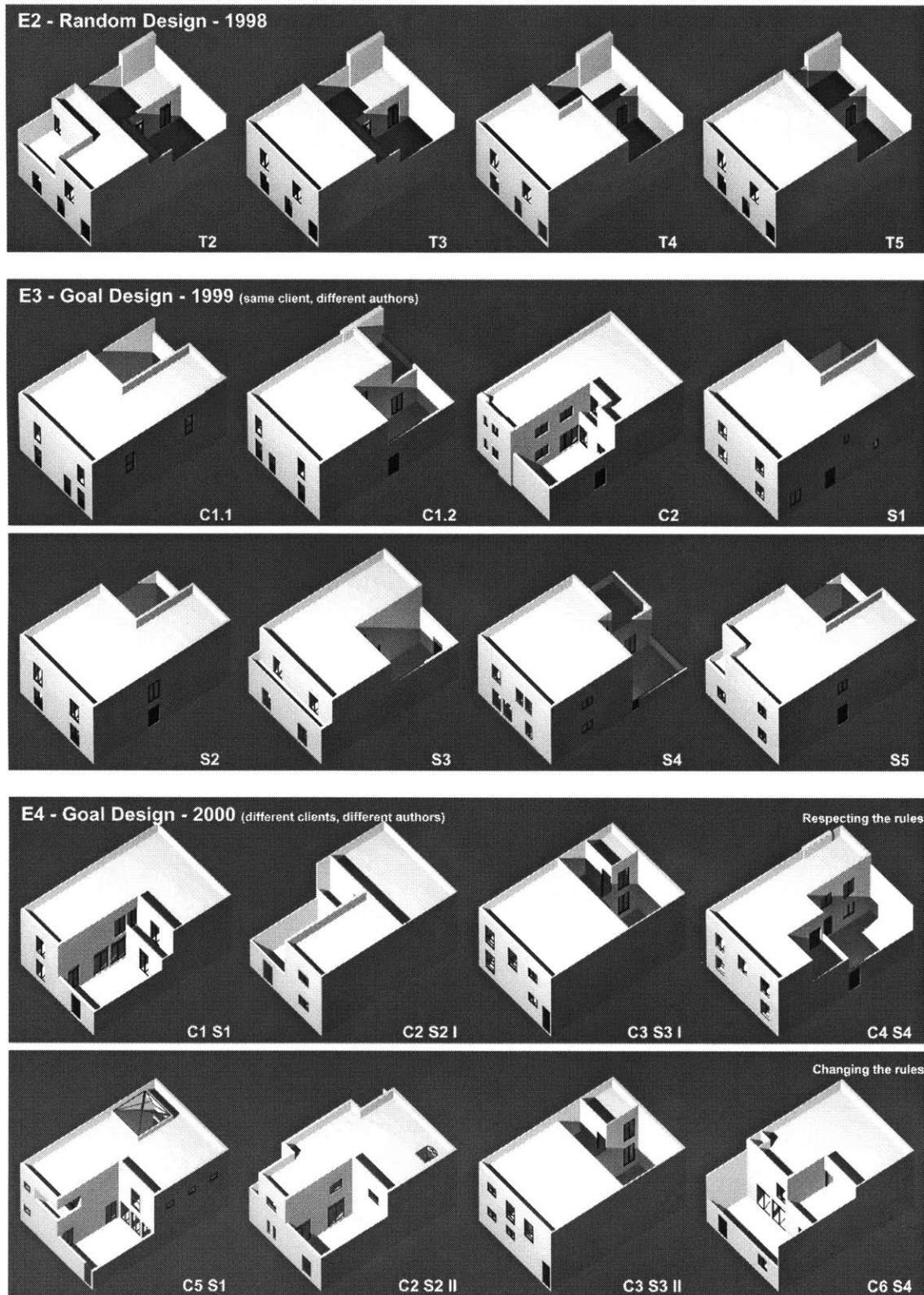


Figure 6.39 – The set of Malagueira designs by design subjects in the experiments.

## **References**

Akin, Omar 1987 "Exploration of the Design Process" in *The Design Process*, pp. 23-28.

## 7. Discursive grammar

### 7.1 Introduction

This chapter presents a mathematical model for the problem of finding a solution that matches given requirements within the set defined by a shape grammar, called discursive grammar. Chapter 5 presented a shape grammar for Siza's Malagueira houses, based on the compositional principles of dissecting and concatenating rectangles. This grammar was developed considering an upper and a lower bound for the universe of design solutions. The upper bound corresponded to the exhaustive set of solutions that could be derived from those two compositional principles, whereas the lower bound corresponded to the subset that could be derived from a strict interpretation of Siza's compositional rules. The grammar was restricted to generate only the solutions in the lower bound. Chapter 6 presented a set of experiments undertaken with the goal of testing such a grammar and with the goal of generating new designs. In these experiments, subjects were asked to use the grammar to generate designs that matched given requirements. In some experiments, they had to respect the grammar rules, and in others they were allowed to change them. The designs were then analyzed to determine whether they satisfied the requirements, and shown to Siza to determine whether he considered them to be in the grammar. Compared results of both experiments, showed that in some cases it was not possible to generate solutions for the requirements, while respecting the lower bound rules, and that it was possible to unrestrict the rules and enlarge the universe of solutions, while maintaining stylistic coherence. Results also suggested re-writing the rules so that spatial configurations

were generated based on given functional requirements. However, enlarging the universe of design solutions makes the problem of finding a solution more difficult. Mitchell (1989) illustrated the need for shape grammars by comparing a designer's attempt to design without one to Gulliver's Lilliputians attempt to write books by randomly combining words. A grammar guarantees that English sentences will be generated, but one problem remains, how can one assure that the grammatically correct sentences will say what we are trying to convey? To expect this is as hopeless as expecting the random concatenation of words to generate English sentences. So, there are two parts to the problem. The first is concerned with the generation of *legal designs*--designs in the language, the other with the generation of *suitable designs*--designs that match requirements given at the outset. In other words, the goal is to generate both formally and semantically correct designs. Only a grammar with such a power can be used as an effective tool for customizing mass housing. The problem of building grammars that generate suitable (semantically correct) designs is, to a certain extent, foreign to previous architectural shape grammars, as they only provide the means to generate solutions that match very general criteria. Engineering grammars, however, have been developed with the goal of generating optimized solutions for given design contexts. This was the case, for instance, of Reddy's and Cagan's<sup>5</sup> truss design grammar. This grammar used a directed stochastic search algorithm, shape annealing, to guide the generation of designs by the grammar towards a global optimum. We propose a different approach called a *discursive grammar*.

## 7.2 Definitions

A discursive grammar is a grammar that is capable of generating both syntactically and semantically correct designs. In other terms, it deals with both form and meaning so that it finds a design within the language that matches given criteria.

### 7.2.1 Technical definition

From the technical viewpoint, a discursive grammar consists of a shape grammar, a description grammar, and a set of heuristics. The concept of shape grammar was invented by Stiny and Gips. (1972) A shape grammar specifies how designs can be composed with shapes starting with an initial shape and then proceeding recursively by applying shape rules. The concept of description grammar was developed by Stiny<sup>1</sup> to account for features of designs not covered by shape grammars. (1981) A description grammar describes the design in terms of other features considered relevant according to some criteria of interest. Stiny suggests that the description grammar can be considered a grammar of another language and that it is possible to translate back and forth between the two languages. Our proposal is to use such a translation mechanism to obtain the design from a goal description. The set of heuristics is used to guide search through the space of solutions until one that closely matches the goal is encountered. This is accomplished by selecting at each step the rules that bring the description of the evolving design closer to the goal description.

---

<sup>1</sup> According to Stiny (1990), designs are descriptions of artifacts and they comprise both shapes and symbols. In this sense, shapes also are descriptions. In the literature, however, the term *shape* has been used to designate *shape descriptions*, whereas as the term *description* has been used to refer to *symbolic descriptions*. We will follow this convention whenever there is no sort of confusion as to which type of description we are referring to, and we will use the extended expressions when such confusion can arise.

### 7.2.2 Operative definition

From the operative viewpoint, the discursive grammar consists of a programming grammar (or programmer) and a designing grammar (or designer.) The programming grammar processes the user and site data (design data) to generate the housing program (design brief). The designing grammar uses the housing program to generate a housing solution (design). The relation between these grammars and the description and shape grammars mentioned above is diagrammed in Figure 7.1. The programming grammar has a description part and an empty shape part, whereas the designing grammar has both a description and a shape part.

Technical viewpoint	Operative viewpoint	
	Programming grammar	Designing grammar
Description grammar	√	√
Shape grammar	∅	√

Figure 7.1- Technical and operative definitions of discursive grammars.

### 7.2.3 Mathematical definition

From the mathematical viewpoint, a discursive grammar can be described by a nine-tuple (D, U, G, H, S, L, W, T, F, I). D is a set of description rules. U is the initial description, to which the first rule applies to start a computation. Other rules than apply to define G, the goal description, that is, the description of the intended design. S is a set of shape rules of the form  $A \rightarrow B$  that specifies that whenever a shape A is found in the design, it can be substituted by a shape B. L is a set of labels that are used to control computations. W is a set of weights associated with shapes in a specified algebra. Weights can be used to control computations, when meaning is assigned to weights as in color grammars (Knight 1989), a special case of grammars with weights. Weights also can be used to account for visual features of designs to improve its

readability like in the use of lines with different thickness.  $T$  is the set of similarity transformations (rotation, translation, scaling, and so on) under which rules apply.  $F$  is a set of functions that assigns values to parameters in rules, for example: the width and length of a rectangle.  $I$  is the initial shape to which the first rule applies to start a computation. Other rules then apply recursively to define a design within the language defined by the grammar. Finally,  $H$  is a set of heuristics that are used to decide which rule to fire at each stage of the design process in such a way as to guarantee that a design with a description that closely matches the goal is generated. Heuristics are to description rules as labels are to shape rules; they constrain the ways in which rules are triggered and fired.

### 7.3 The PAHPA-Malagueira discursive grammar

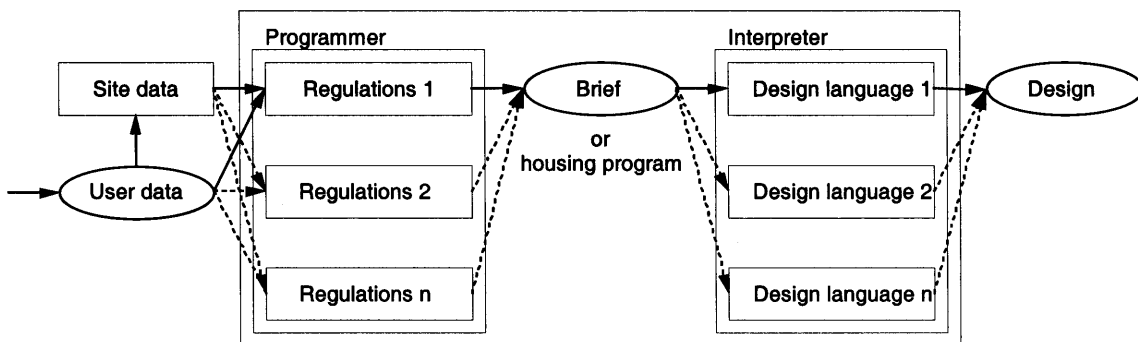


Figure 7.2 – Different programming grammars can be used in combination with different designing grammars.

In theory, different programming grammars can be combined with different designing grammars to form various discursive grammars. (Figure 7.2) First, one could use several programming grammars to generate housing programs appropriate for various contexts. For instance, one could have programming grammars that encode the rules set by the Swedish, or the Portuguese regulations. Second, one could use several designing grammars to generate a solution for the same housing program. For instance,

one could have grammars that generate designs in the style of Frank Lloyd Wright's prairie houses, or Siza's Malagueira houses. Then, one could generate a housing program using any of the programming grammars, and then generate a solution for that program using any of the designing grammars.

In practice, the independence of programming grammars from designing grammars is limited. The first limitation exists because there needs to be contextual compatibility between the two grammars, otherwise no solution can be found in the designing grammar that satisfies the housing program specified by the programming grammar. For instance, the housing program specifies area requirements that cannot be met by the designing grammar. A way to overcome this limitation is to ensure that the contexts of both grammars match. For instance, one could use a programming grammar that generates housing programs that are appropriate for a given social, cultural, economic, and geographic context, say Portugal. Then, one could have several designing grammars that generate solutions that also are appropriate for this context in different styles, such as Álvaro Siza's, Rafael Moneo's, Frank Lloyd Wright's, and so on. The second limitation exists because the programming grammar needs to be informed of constraints posed by the designing grammar. For instance, the area limits set by a designing grammar needs to be taken into account in the generation of the housing program to limit the possibility of generating over-constrained programs. This limitation is solved by conceiving the programming and the designing grammars in such a way that when the two grammars are put together, information on constraints posed by the designing grammar are transferred to the programming grammar.

In the specific discursive grammar proposed in this work, the programming grammar adapts both the rules of the Portuguese housing program guidelines (Programa



Habitacional-PH, Pedro 1999), and the rules of the Portuguese housing evaluation system (Indicadores de Qualidade Arquitectónica, Pedro 2000). It is, therefore, called the adapted Portuguese Housing Program and Evaluation grammar, or just the PAHPA grammar.<sup>2</sup> The designing grammar encodes Siza's rules for the design of Malagueira houses. Therefore, the proposed grammar is called the PAHPA-Malagueira grammar. (Figure 7.3) The PHAPA programming grammar is presented in Section 7.4, and the Malagueira designing grammar is presented in Section 7.5.

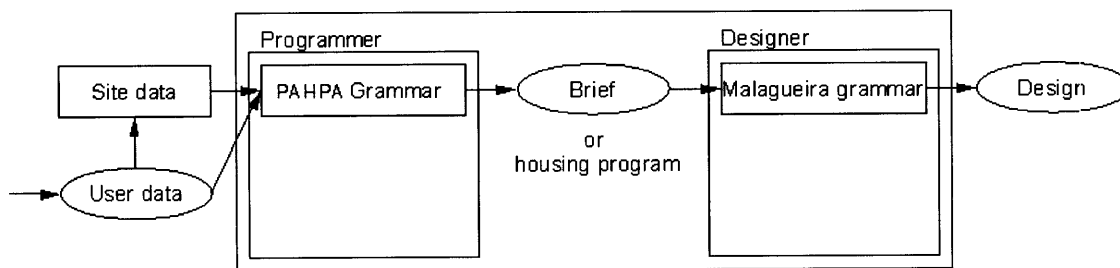


Figure 7.3 - The proposed PAHPA-Malagueira grammar includes the grammar of the Portuguese housing program and evaluation guidelines (programming grammar) and the grammar of Siza's Malagueira houses (designing grammar).

#### 7.4 The PAHPA programming grammar

The programming grammar follows the PHAP guidelines rules to generate the housing program from site and user data and to evaluate the outcome. The PAHP were selected for the following reasons. They are the sequential development of the documents that regulated the design of housing when Siza designed the Malagueira houses. They are recommended by the major Portuguese institutions that regulate housing issues. They are written in a way that allows the flexible specification of housing programs. Finally, they were selected after the grammars were shown to have the sort of contextual compatibility referred to in the section above. Nevertheless, the programming grammar does not strictly follow the PAHP. It includes only a subset of the housing features that

<sup>2</sup> PAHPA - *Programa e Avaliação Habitacional Portuguesa adaptada*

they foresee, and introduces new ones. It also reorganizes features into a different hierarchy. Moreover, it includes rules that capture the intelligence of a human programmer browsing through the PH to explore design brief possibilities, thereby regulating the interdependency among features.

Stiny indicates two issues as being crucial in the development of description grammars (1981). The first issue is fixing the contents of the description, that is, which features to include. The second issue is developing the description rules. Choosing the categories and developing the rules are important because they determine which questions to ask the client, and how to derive the housing program, and ultimately, how to derive and evaluate the design.

Two approaches were taken to overcome such difficulties. The first approach was to consider the features proposed in the PH presented in Table 3.1. The second approach was to consider the features used by the subjects in the experiments presented in Chapter 6. The final selected features are presented in Table 7.1 in a simplified manner, and in Table 7.2 in more detail, including the data structure. They include only a subset of the PAHPA features, but it is possible to extend the grammar to account for non-included features, such as comfort, security, and personalization. They also include new features, such as context, morphology, typology, and aesthetics. The values that each feature can take are listed in Table 7.3.

#### **7.4.1 Description (features): constraints, quality, and cost**

As shown in Figure 7.1, the features are organized into three main groups, according to the role that they perform in the derivation of the housing program and in the derivation of the design solution.

**Table 7.1 – Main features in the housing program description.**

Main groups	Groups of features	Features	Elemental features	
Constraints		Context	Lot	$\alpha_0$
			Urban context	$\alpha_1$
			Solar orientation	$\alpha_2$
		Typology	Customization	$\alpha_3$
			Users	$\alpha_4$
			Bedrooms	$\alpha_5$
			Quality	$\alpha_6$
		Morphology	Housetype	$\alpha_7$
			Floors	$\alpha_8$
		Balconies	$\alpha_9$	
Quality	Function	Spatiality	Capacity (dwelling)	$\alpha_{10-13}$
			Capacity (spaces)	$\alpha_{10-13}$
			Articulation (spaces)	$\alpha_{10-13}$
			Spaciousness (dwel. & spaces)	$\alpha_{15-16}$
			Topology	$\alpha_{17}$
			Aesthetics	Proportion
Cost		Construction	$\alpha_{24}$	

The first group is formed by features such as context (lot, urban context, and solar orientation), typology (customization, number of users, number of bedrooms, and quality level), and morphology (housetype, number of floors, and balconies.) This group is called constraints because the values of elemental features are specified by the user<sup>3</sup> and cannot be changed by the programmer, thereby constraining the values of subsequent features. The only exceptions are quality and balconies, whose values can be updated after the user changes quality features. The second group includes function and aesthetics. Function includes spatiality (dwelling capacity, and space capacity, articulation, and spaciousness) and topology. The only aesthetic quality considered is proportion, which Siza regarded as important. The features in this group describe the performance of the programmed house and are referred to as qualities. The user can assign weights to these qualities to express their relative importance and to determine

<sup>3</sup> In this Chapter, the term “user” will be used to refer to the user of the program, and the term “dweller” will be used to refer to the user of the house.

the overall quality. The third group includes only the construction cost. Constraints, qualities, and cost frame the problem of designing a house as follows. Within the specified contextual, typological, and morphological constraints, design a house with the specified qualities at a specified cost.

As shown in Table 7.2, the detailed description includes two parts and shows the data structure used to encode the housing program. The first part is the *variable description*, that is, the set of features that constitute the housing program (design brief), whose values are defined by user prompt information and the programming rules. These features are identified with the Greek letter  $\alpha$ , followed by a number. They correspond to the features mentioned above plus the features building elements (windows, doors, walls, and pavements) and history, which are not specified by the user, but constitute features of the future design. History is the record of the design rules application sequence. The variable description actually is the program description. The second part of the detailed description is the *fixed description*, whose features have pre-defined values that cannot be changed by the user. These categories are identified with the letter  $\beta$ , followed by a number. Among such features are spatial dimensions, sectional dimensions, and cost. Spatial dimensions refer to the minimum width, height, and area of the space that is required to perform a given function (e.g. sleeping) whose values are those indicated in the PAHPA. Sectional dimensions refer to the pavement thickness, and the floor height, whose values are determined by the upper and lower bounds of the values found in the Malagueira houses, presented in Table 4.4. The cost includes the costs per square meter of enclosed, covered, and open spaces, which are regularly published in governmental tables. A brief explanation of each description precedes the explanation of its rules, given further below.

**Table 7.2 – Housing program description**

Variable description (program description)				
Features	Groups of variables	$\alpha$	Variables	
Contexts	Lot	$\alpha_0$	< w, l, h, a >	
	Urban	$\alpha_1$	< front, right, back, left >	
	Solar orientation	$\alpha_2$	< front, right, back, left >	
Typology	Customization	$\alpha_3$	< degree >	
	Users	$\alpha_4$	< number, [(name, gender, age, share),...] >	
	Bedrooms	$\alpha_5$	< number, [(couple, number), (double, number), (single, number)] >	
	House quality	$\alpha_6$	< initial quality, current quality >	
Morphology	Yard location	$\alpha_7$	< yard >	
	Floors	$\alpha_8$	< floors >	
	Balconies	$\alpha_9$	< balconies >	
Spatiality	Capacity	Minimum obligatory spaces	$\alpha_{10}$	< [use, number, ((articulation, number)...)...] >
		Initial obligatory spaces	$\alpha_{11}$	< [use, articulation, weight],... >
		Current optional spaces	$\alpha_{12}$	< [use, articulation, weight],... >
		Current spaces	$\alpha_{13}$	< [name, id, ( users, functions, (capacity, weight), (articulation, weight), (spaciousness, weight), (insertion point, rotations, width, length, height, area)],... >
		Zones	$\alpha_{14}$	< [use,rooms, area],... >
	Spaciousness	Areas	$\alpha_{15}$	< available, (max interior gross, min exterior gross, 1 <sup>st</sup> Floor gross), (max interior gross, min exterior gross, 2 <sup>nd</sup> Floor gross), (max interior gross, min exterior gross, house gross), useful/gross >
			$\alpha_{16}$	< used, (inhabitable, interior useful, exterior useful, 1 <sup>st</sup> Floor useful), (inhabitable, interior useful, exterior useful, 2 <sup>nd</sup> Floor useful), (inhabitable, interior useful, exterior useful, house useful), inhabitable/useful >
Topology	Adjacency graph	$\alpha_{17}$	< [ (room1, room2, relation, weight) ] ... >	
Building elements	Windows	$\alpha_{18}$	< [window, (room 1, room 2), (insertion point, depth, width, height, area)],... >	
	Doors	$\alpha_{19}$	< [door, (room 1, room 2), (insertion point, depth, width, height, area)],... >	
	Walls	$\alpha_{20}$	< [wall, (room 1, room 2), (insertion point, thickness, width, height, area)],... >	
	Pavements	$\alpha_{21}$	< [pavement, floor, (insertion point, width, length, thickness, area)],... >	
Aesthetics	Proportion	$\alpha_{22}$	< [proportion1, weight],... >	
Quality		$\alpha_{23}$	< [function, weight], [spatiality, weight], [capacity, weight], [articulation, weight], [spaciousness, weight], [topology, weight], [aesthetics, weight] >	
Cost	Cost	$\alpha_{24}$	C	
History	Rule	$\alpha_{25}$	< r1, r2,..., rm >	

Fixed description				
Spaces dimensions	width	$\beta_1$	tables	width (space, quality)
	height	$\beta_2$	tables	height (space, quality)
	area	$\beta_3$	tables	area (space, quality)
Sectional dimensions	Pavement thickness	$\beta_4$		thickness (pavement)
	Floor height	$\beta_5$		height (floor)
Cost		$\beta_6$	table	unit_cost (element, material)

**Table 7.3 – Housing program features values**

Feature	Feature	$\alpha$	Feature	Values	
Morphology	Lot	$\alpha_0$	Width	8 m <sup>2</sup>	
			Length	12 m <sup>2</sup>	
			Area	96 m <sup>2</sup>	
	Urban	$\alpha_1$		Houses on three sides (default), house on one side, house at the back, house on the side and back	
Solar orientation	$\alpha_2$		N, NE, E, SE, S, SW, W, NW		
Typology	Customization	$\alpha_3$		Custom, type (default)	
	Dwellers	$\alpha_4$	Number	1, 2, 3, 4, 5, 6, 7, 8, 9	
			Name	User prompted Blank (default)	
			Gender	Male Female Blank (default)	
			Age	0-1, 2-5, 6-13, 14-17, 18-23 23-65, > 65, Blank (default)	
	Share	Room Bed Blank (default)			
	Bedrooms	$\alpha_5$	Number	1, 2, 3, 4, 5	
Quality*	$\alpha_6$	Initial	Minimum (default), medium, maximum (high)		
		Current	Minimum (default), medium, maximum (high)		
Morphology		$\alpha_7$	Yard	Front, back	
		$\alpha_8$	Floors	1, 2	
		$\alpha_9$	Balconies	True, False	
Spatiality	Capacity (dwelling)	$\alpha_{10}$	Minimum	List of spaces' IDs	
			$\alpha_{11}$		Initial obligatory
			$\alpha_{12}$		Optional
			$\alpha_{13}$		Current
			$\alpha_{14}$		Zones
	Spaciousness (dwelling)	$\alpha_{15}$	Available	See Tables 7.8-7.10	
			$\alpha_{16}$	Used	m <sup>2</sup>
	Name			Kitchen, laundry, pantry, living, closet, step-in-closet, stairs, patio, bedroom, bathroom, circulation, corridor, studio, balcony (terrace)	
	Space ID			Random number	
	Functions			See table 7.7	
	Capacity (spaces)			1, 2, 3, 4, 5, 6, 7, 8, 9	
Articulation (spaces)			Included, delimited, isolated		
Spaciousness (spaces)			See Tables 7.18-30		
Topology		$\alpha_{17}$	Relation	Away, close, adjacent, window, door, passage, merged, any (default)	
Aesthetics		$\alpha_{23}$	Proportion	1:1, 1: $\sqrt{2}$ , 1:2, 2:3, 3:4, 5:6	
Quality		$\alpha_{22}$	Weights	0, 5, 10, 15, ..., 100	
Cost		$\alpha_{24}$	Construction	USD \$ / m <sup>2</sup>	
History		$\alpha_{25}$	Sequence of rules	Sequence of rule numbers	

### **7.4.2 User/programmer interface**

The programming grammar captures the rules of the Housing program, but the study of the client/designer interaction, undertaken with the experiments described in Chapter 6 was instrumental for the definition of the user/programmer interface. The goal of the interface is to support the dynamic user interaction suggested by the experimental results as a way of limiting the possibility of over constraining and ill defining the design problem. The envisaged interface is represented in Figure 7.4.

### **7.4.3 Dependency among features**

The dependency among features and the flow of information originated by the application of the programming rules, is represented in Figure 7.5. In brief, the flow proceeds as follows. The context, the morphology, and the maximum affordable cost determine the available area. Then, as the user specifies the number of dwellers and the desired quality level, the programmer calculates the lists of obligatory and optional spaces (dwelling capacity), as well as each space's capacity, articulation, and spaciousness. Then, these spatial qualities are used to calculate the minimum width, height, and area of each space. These dimensional requirements, together with the list of spaces, are used to calculate the total used area, and then, to estimate the cost of the house. The user can change the lists of spaces, by adding and subtracting spaces, or change their spatial qualities. The programmer, then, recalculates the quality level, the available area, and the estimated cost. The number of floors and the list of spaces cause the programmer to define required and recommended topological relations among spaces. The user can then subtract recommended relations and add others. The programmer uses the topological relations in which floors assignments are involved to define the available area per floor. When the available area becomes close to zero, the user cannot make any further changes that require an increase in area. The user is,

### Context

Urban:

Orientation:

### Typology

Custom  Type

User	Gender	Age	
John	Male	23-65	Add
Mary	Female	23-65	Del

User	User	Share
David	Mary	Bed

Bedrooms	Users	Quality
2	3	minimum

### Morphology

Yard	Floors	Balconies
Front	2	yes

### Spatiality

#### Capacity (dwelling)

	Space	Weight	
Obligatory	Kitchen	100	
Optional	Playspace	35	Add
Current	Living	100	Del

#### Capacity (spaces)

	Value	Weight
Capacity	6	100
Articulation	Isolated	35
Spaciousness	Minimum	35

Width	Height	Area
4.00	2.40	16.00

#### Spaciousness (available and used areas)

non-useful	interior	exterior	gross	
38.80	112.50	- 4.00	108.50	free
inhabitable	interior	exterior	useful	
23.50	28.50	31.50	60.00	used

Ai / Au		Au / Ag	
actual	minimum	actual	minimum
0.82	0.77	0.77	0.77

### Topology

Rooms	Relation	Weight	
Kitchen			
Laundry	adjacent	90	Add
Kitchen/laundry merged		65	Del

### Aesthetics

Proportion	Weight
1:1	30

### Quality

Current:

#### Weights

Function	Spatiality	Capacity
45	60	40

#### Articulation

#### Spaciousness

#### Topology

#### Aesthetics

### Cost

Space Type	Cost / m2	Current cost
Covered	\$ 350.00	\$37,500.00

### Program

Buttons: Done, Save, Load

### Solution

Buttons: Derive, Save, Load

Figure 7.4 – The interface envisaged for the programming grammar.



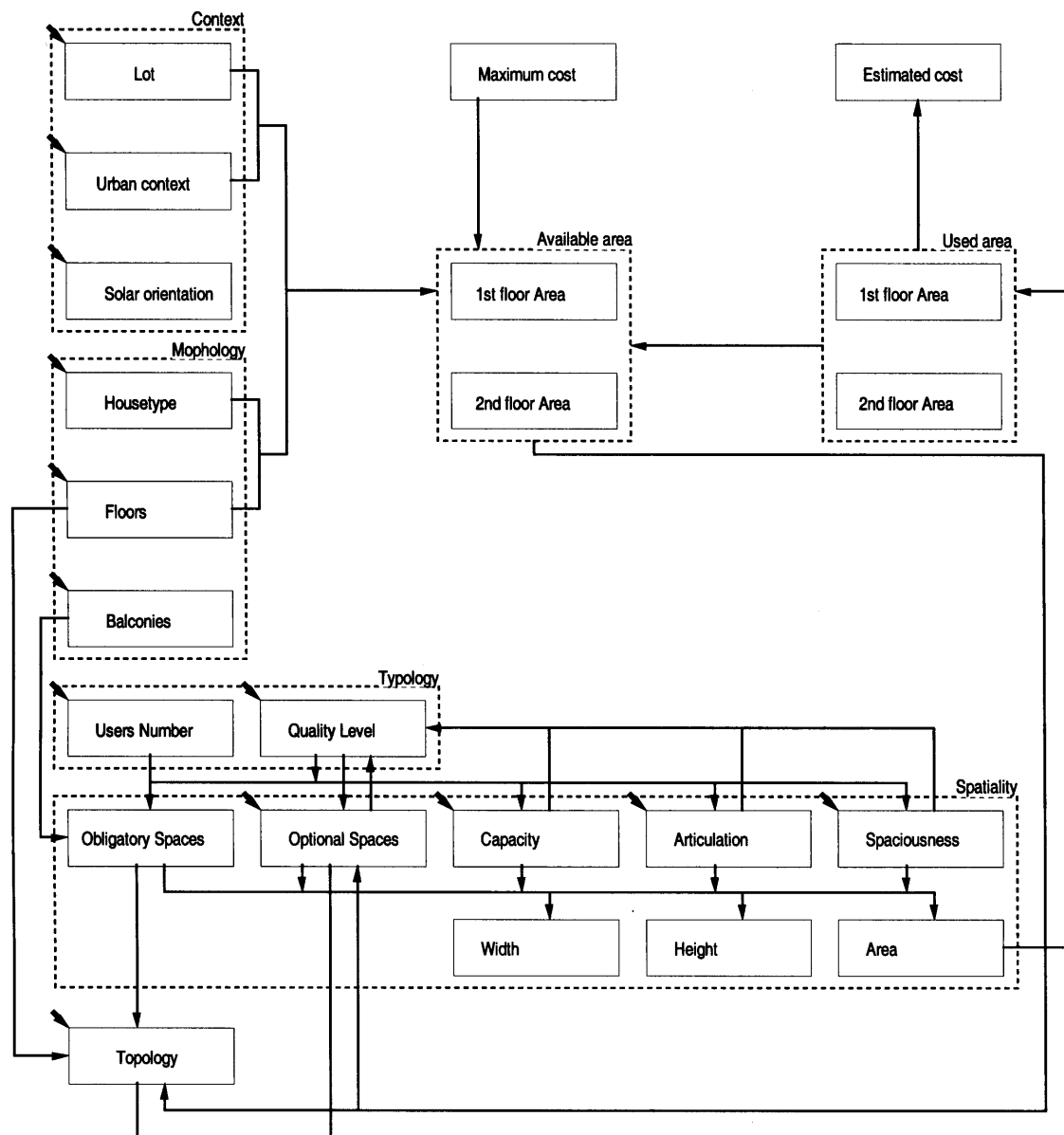


Figure 7.5 – The information flow from user and site data to the definition of the design brief by the programmer.

thus, led restrict his/her choices, thereby limiting the possibility of over-constraining the design problem. The programmer also assigns default weights to the requirements, which the user can change to set his/her own priorities. Once the housing program is defined it can be saved, retrieved, or sent to the designer for generating the solution.

The generation of default requirements by the programmer saves the user from the burden of defining all the requirements while overcoming the ill-definition problem. The user just has to define those requirements that he or she considers essential. Moreover, the user can change the initial requirements after seeing the solution generated by the designer, thereby refining the initial problem.

#### **7.4.4 Rules**

The rules of the programming grammar are shown in Table 7.35 at the end of this section. There are two types of rules. One is used for user-prompted data and the other to programmer-specified requirements. The first rule (g0) initializes a description consisting of 31 features. Explanations of the rules for the function, aesthetics, and cost features are provided below. In the explanation of the programming rules, the hierarchy of features was flattened to allow the explanation to follow the flow of rule interaction. Nevertheless, to help the reader to keep track of the position of the feature in the hierarchy, the sequence of the preceding hierarchical levels is shown in each section's heading.

##### **7.4.4.1 Constraints: Context**

The context describes basic site features, such as lot type, urban context and solar orientation. Topography was not included because in the Malagueira housing grammar the lot always becomes flat after urban planning rules are applied to sloped terrain. The programming grammar can be extended to include other site features, provided that the designing grammar includes rules to deal with such features. Rule g1 describes the features of the usual Malagueira land plot (width, length, and area). No rules were included to describe other plots, but it is possible to extend the grammar to include them. Rule g2 describes the default urban context. Rules g3 through g6 include in the

description user-prompted data regarding the urban context. These rules correspond to the four possible urban contexts that are in accordance with the Malagueira planning rules. They manipulate description  $\alpha_1$  (urban context), but also  $\alpha_{13}$  (list of current spaces) by subtracting the current descriptions and adding the new ones. They need to manipulate  $\alpha_{13}$  because the designing grammar will need the description of the contextual spaces (a set of four boxes named house or street placed around the lot) to apply its rules. Rule g7 describes the default solar orientation. Rules g8 through g15 include in the description user-prompted data about the solar orientation. They manipulate descriptions  $\alpha_2$  (solar orientation) and  $\alpha_{13}$  (list of current spaces) by updating these descriptions with the new data.

#### **7.4.4.2 Constraints: Typology**

The typology is a synthesized description of the house. It is determined by the degree of customization ( $\alpha_3$ ), the number of bedrooms ( $\alpha_4$ ), the number of dwellers ( $\alpha_5$ ), and the quality level ( $\alpha_6$ ). The programmer sets the default degree of customization to housetype (rule g16). The user can change it to describe a customized house (rule g17), or reset it back to housetype (rule g18).

If the user chooses to describe a customized house, the user needs to provide detailed dweller information ( $\alpha_4$ ) (rule g19). For each dweller, the programmer will add a bedroom to the list of current spaces ( $\alpha_{13}$ ) and to the list of sleeping zone spaces ( $\alpha_{14}$ ). Also, it will use the quality level ( $\alpha_6$ ) to retrieve the appropriate dimensional requirements ( $\beta_{1-5}$ ), and to update the used area ( $\alpha_{15}$ ), the available area ( $\alpha_{16}$ ), and the cost ( $\alpha_{24}$ ), accordingly. If no bedroom exists on the first floor, the new one will be assigned to this floor, otherwise, it will be assigned to the second floor ( $\alpha_{17}$ ).

Then, the user can specify that two given dwellers can share a bedroom (rule g20) or a bed (rule g21), and the programmer will merge their single bedrooms into one double or couple bedroom, respectively.

If the user decides to describe a housetype, the user needs to specify the number of bedrooms ( $\alpha_5$ ) (rules g22-g27) and the number of dwellers ( $\alpha_4$ ) (rules g28-g37). These features are mutually dependent as shown on Table 7.5. This means that, for instance, if the number of bedrooms is 1, then the number of dwellers can be only 1 or 2 (rule g23). Conversely, if the number of dwellers is 2, the number of bedrooms can be only 1 or 2 (Rule g30). Such a dependency exists because the *PAHP* guidelines assume that the household consists of a traditional nuclear family formed by the parents and their kids, that the parents do not share the bedroom with their children, and that no more than two children can share a bedroom.

**Table 7.5- Dependency between the number of bedrooms and the number of dwellers**

<b>Bedrooms</b>	<b>Dwellers</b>
blank	blank, 1 to 9
1	1 to 2
2	2 to 4
3	4 to 6
4	5 to 7
5	7 to 9

#### **7.4.4.3 Constraints: Morphology**

The morphology describes the basic features regarding the overall shape of the house, including the location of the yard ( $\alpha_7$ ), the number of floors ( $\alpha_8$ ), and whether it has balconies on the second floor ( $\alpha_9$ ). The programmer sets front as the default location of

the yard (rule g38), which the user can then change (rule g39). The number of floors depends on the number of bedrooms as follows. If the number of bedrooms is one ( $\alpha_5$ ), the number of floors ( $\alpha_5$ ) is one (rule g40). If it is two, it can be either one or two, depending on user's choice (rule g41). Finally, if the number of bedrooms is more than two, the numbers of floors is two (rule g42). As default, the programmer specifies that the house will have balconies (rule g43), which the user can change (rule g44).

#### 7.4.4.4 Quality

Quality describes how well the dwelling satisfies the functional and aesthetic requirements. In both the customized house and the housetype scenarios, the programmer sets the dwelling quality level to minimum, and assigns to qualities the default relative weights shown in Table 7.6 (rule g45). The user can, then, change the dwelling quality level by choosing one from minimum, medium, and maximum (rule g46), and reset the relative weights of qualities by selecting a value from the interval [5, 10, ..., 100] (rule g47). The programmer standardizes the weights to assure that they add up to 100, at each level in the qualities tree. For instance, if the user assigned the weight 30 to function, and the weight 30 to aesthetics, then the programmer resets both to 50.

**Table 7.6 - The default weights assigned to qualities**

Main groups	Groups of qualities	Quality	Elemental qualities				
Quality	Function	50	Spatiality	50	Capacity (dwelling)	25	$\alpha_{10-13}$
			Topology	50	Capacity (spaces)	25	$\alpha_{10-13}$
	Articulation (spaces)	25			$\alpha_{10-13}$		
	Spaciousness (dwel. & spaces)	25			$\alpha_{15-16}$		
	Proportion				$\alpha_{22}$		
	Aesthetics	50	Topology	50	$\alpha_{17}$		
			Proportion		$\alpha_{22}$		

The programmer uses the dwelling quality level to make assumptions about the dwelling, such as the spaces that it should include (dwelling capacity), the spatial requirements of its spaces (capacity, articulation, and spaciousness), as well as the required topological relations among them. If the user does not indicate specific quality levels for these features, the dwelling quality level becomes their default quality level. If the user indicates specific quality levels, the programmer re-calculates the dwelling quality level by calculating the weighted average of the qualities quality levels (rule g48).

The average dwelling quality level equals the weighted average of the function and the aesthetics average quality levels (Table 7.35, Rule g48, Equation 1). The function quality level is the weighted average of the spatiality and topology quality levels (Equation 2). The average spatiality quality level is the weighted average of the dwelling capacity, and the spaces capacity, articulation, and spaciousness average quality levels (Equation 3).

The average dwelling capacity is the weighted average of its spaces dwelling capacity quality levels (Equation 4), calculated as follows. If the space is in the minimum dwelling capacity list of spaces, its dwelling capacity is 1 (minimum); if it is in the medium dwelling capacity list, but not in the minimum, its dwelling capacity is 2 (medium); and if it is in the maximum capacity list, but not in the minimum and medium, its dwelling capacity is 3 (maximum).

The space capacity average quality level is the weighted average of its spaces capacity quality levels (Equation 5). The rules for determining the quality level of a space's capacity are listed in Table 7.35. For instance, if the space is a bathroom, and if its capacity is lavatory, then the quality level is 1 (minimum); and if its capacity is shower,

then its quality level is 2 (medium); and if its capacity is bathtub, then its quality level is 3 (maximum).

The articulation average quality level is the weighted average of its spaces articulation quality levels (Equation 6). The rules for determining the quality level of a space's articulation are very straightforward. If the space's articulation is included, then its articulation quality level is 1 (minimum); if it is delimited, then its articulation quality level is 2 (medium), and if it is isolated, then it is 3 (maximum).

The spaciousness average quality level is the weighted average of its spaces spaciousness quality levels (Equation 7). The spaciousness quality level of a space is the average of its width, height, and area quality levels, whose individual values are shown in Tables 7.18-7.30.

The topology average quality level is the weighted average of the quality levels of the relations among all the dwelling spaces (Equation 8). The rules for determining the quality level of a relation between two spaces are listed in Tables 7.35. For instance, if the required relation among two spaces is door, and if the actual relation is door, then the quality level is 3 (maximum); if the actual relation is adjacent or window, then the quality level is 3 (medium); if it is close, then its is 1 (minimum); otherwise it is 0.

The aesthetic quality level is the weighted average of all the spaces aesthetic quality levels (Equation 9). The programmer assigns the same default weights to all the proportions, no matter the space. This means that in the definition of the housing program, proportion has no impact on the overall quality. The default quality level is 3 (maximum), independently from the space and the proportion. In the design of the

solution, however, the quality level of the proportion of a given space is lower than 3, if it does not have the specified proportion.

If the average dwelling quality level is in the interval [0, 1], then the quality level is said to be minimum; if it is in the interval ]0, 2], then it is medium; and if it is in ]2, 3], then it is maximum.

#### **7.4.4.5 Quality: function: *spatiality***

*Spatiality* describes the spatial features of the dwelling and its spaces. It includes the dwelling capacity (the list of spaces in the dwelling), the space capacity (the number of users that the space shelters), articulation (how the functions sheltered in the space relate to each other) and spaciousness (how large the space is).

A *function* is a behavioral activity performed with a certain goal (e.g. sleeping). A *space* is defined as the three-dimensional volume that is required to perform a given function.

A *room* is an enclosed space that shelters one or more functions. A dwelling is a set of rooms, sheltering several related dwelling functions. The relationship between functions, spaces, rooms, and dwelling is diagrammed in Figure 7.6.

The dwelling functions and rooms are listed in Table 7.7, after Portas (1969) and Herbert et al (1978), quoted in Pedro (2000). The Malagueira houses considered in the corpus (Chapter 4) sheltered all of these functions, with the exception of "exterior storage" and "parking car," mainly because the lot was small and the cars were parked in clusters of garages segregated from the houses. However, Type F (Section 6.2) did include exterior storage spaces, and people park their cars in front of their houses disregarding functional segregation. Therefore, it seems reasonable to accept that the functions

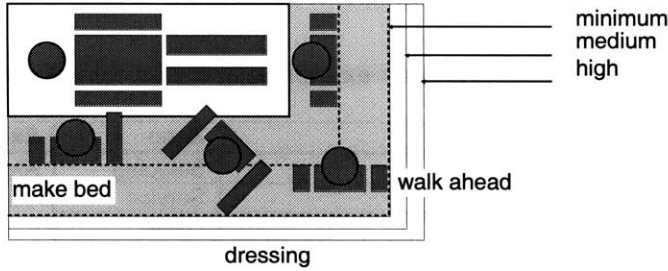


**Function / Space: SLEEPING**

Function - activities - furniture:

*bed* lay on bed walk sideways

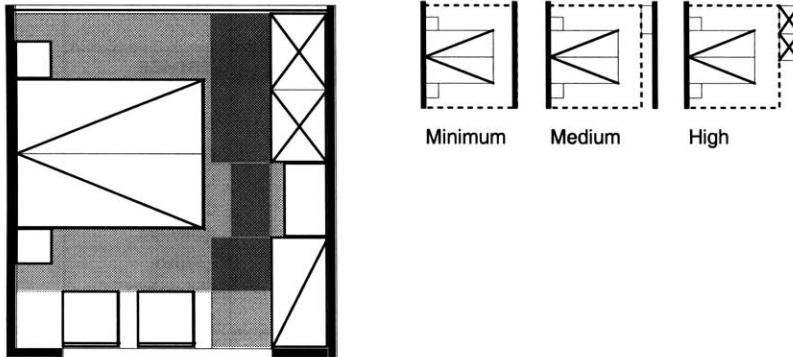
Quality level:



**Room: BEDROOM**

Function: sleeping  
storing personal clothes

Quality level:



**Dwelling: one-bedroom, two-dwellers**

Rooms: double bedroom, kitchen,  
bathroom, living room, hall

Quality:

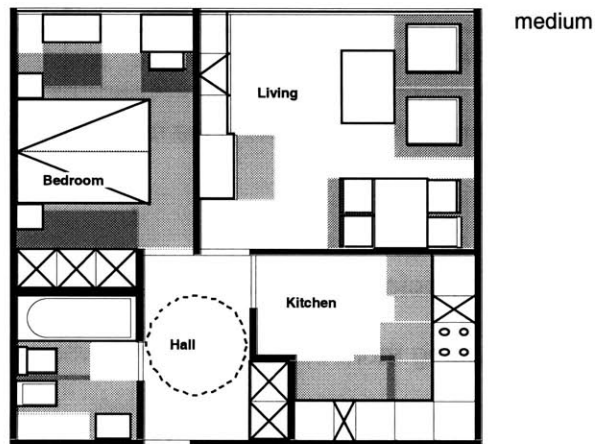


Figure 7.6 – From spaces to rooms, to dwellings. Top: the space required for performing the function sleeping by one person. Middle: the bedroom required for performing the function sleeping by two people. Bottom: a dwelling congregating several functions for two people.

“exterior storage” and “park car” can be included in the design of future Malagueira houses.

**Table 7.7 - Dwelling functions, rooms, and their existence in Malagueira houses**

	Function		Room	Zone	Existence *
1	<b>Sleeping</b>		<b>Bedroom</b>	Sleeping	Always
2	<b>Cooking</b>		<b>Kitchen</b>	Service	
5	<b>Living</b>	<b>Living</b>	<b>Living-room</b>	Living	
6		Hosting	<b>Living-room, bedroom</b>	Living	
13	<b>Bathing</b>		<b>Bathroom</b>	Sleeping	
14	<b>Being outside</b>		<b>Patio, terrace, balcony</b>	Patio	
15	<b>Circulation</b>		<b>Corridor, staircase</b>	Living, sleeping, service	
3	Dining	Informal dining	<b>Dining-room, Kitchen, living-room</b>	Service	Sometimes
4		Formal Dining	<b>Dining-room, Living-room</b>	Service	
7	Work/leisure	Playing (children)	<b>Studio, bedroom, living</b>	Living, sleeping, service	
8		Studying (youth)	<b>Studio, bedroom, living</b>	Living, sleeping, service	
9		Working (adult)	<b>Studio, bedroom, living</b>	Living, sleeping, service	
10	Clothing	Ironing clothes	<b>Laundry, kitchen, living-room, bedroom</b>	Service	
11		Washing clothes	<b>Laundry, kitchen</b>	Service	
12		Drying clothes	<b>Laundry, kitchen, patio, terrace, balcony</b>	Service	
16	Interior storage		<b>Closet, pantry, Living-room, circulation</b>	Living, sleeping, service	
16	<i>Exterior storage</i>		<i>Closet, garage</i>		Never
17	<i>Parking car</i>		<i>Garage</i>		

Notes: A function in bold is a main dwelling function. A rooms in bold is the room where a given function is most likely located in Malagueira houses.

#### 7.4.4.6 Quality: function: spatiality: **Dwelling Spaciousness - Area**

The programmer will prevent the user from adding more spaces or rooms when the combined area of all the spaces that have already been added exceeds the available area. Checking the area presents three difficulties. First, national regulations classify the area of different spaces into different categories, and impose different restrictions on

each category. Second, the Malagueira regulations cause the available area to depend on the morphology. Third, the area occupied by walls and other building elements is unknown at the specification time. There are, nevertheless, ways to overcome such difficulties as explained below.

The PAHPA, like other Portuguese regulations consider that a single-family house (*habitação*) is composed of the dwelling (*fogo*) and of the annex (*dependencias*). The classification of spaces into dwelling and annexes is shown in Table 7.8. In short, the dwelling consists of all the interior spaces, except garages and detached storage rooms; and the annex consists of exterior spaces, garages, and storage rooms. The Portuguese regulations also make a distinction between gross area (*área bruta*), useful area (*área útil*), and inhabitable area (*área habitável*), and specify limits for the ratios between these areas (Table 7.9). The Malagueira regulations impose limits on the gross interior area, depending on the morphology (Table 7.10).<sup>4</sup> Therefore, it is possible to calculate the available useful areas by applying those indexes to the gross areas for the selected morphology.

To check whether the available area has been exceeded, it is necessary (1) to match the sum of the useful interior areas against the available useful interior area to guarantee the respect for yard's minimum area; and (2) to match the sum of the interior and exterior useful areas to assure that the total available useful area is not exceeded.

Rules 49-53 capture the five morphological situations and determine the corresponding available areas as shown in Table 7.10. For instance, if the yard is at the back, there is a

---

<sup>4</sup> It was introduced a change to the Malagueira building regulations by considering that the exterior area on the second floor could be smaller if there is a street at the back.

street at the back, and there are no balconies on the second floor, then the first floor's interior, exterior, and total available areas are 68.50 m<sup>2</sup>, 27.50 m<sup>2</sup>, and 96.00 m<sup>2</sup>; the second floor's are 68.50 m<sup>2</sup>, 0.00 m<sup>2</sup>, and 72.50 m<sup>2</sup>; and the total are 137.00 m<sup>2</sup>, 27.50 m<sup>2</sup>, and 164.50 m<sup>2</sup>, respectively (Rule 53). These rules are the rules that transfer information from the designing grammar to the programming grammar, as mentioned in Section 7.3. More accurately, these rules are urban planning rules, and if the urban planning Malagueira grammar was developed, they would be transferred from this grammar to the programming and designing grammars.

**Table 7.8 - Classification of functions, spaces, and rooms from an area measurement viewpoint**

House	Gross area	Useful area	Inhabitable area	Functions	Rooms	
Dwelling	Gross area $A_{gd}$	Useful area $A_{ud} = A_{gd} \cdot i_{Aud/Agd}$	Inhabitable area $A_{id}$	1	Sleeping	Bedroom
				7	Playing	Studio, bedroom, living
				8	Studying	Studio, bedroom, living
				9	Working	Studio, bedroom, living
				2	Cooking	Kitchen
				3	Informal dining	Dining-room, Kitchen, living-room
				4	Formal Dining	Dining-room, living-room
				5	Living	Living-room
				6	Hosting	Living-room
				10	Ironing clothes	kitchen, living-room, bedroom
				11	Washing clothes	kitchen
				12	Drying clothes	kitchen
			Non-inhabitable area $A_{nd}$	10	Ironing clothes	Laundry
				11	Washing clothes	Laundry
				12	Drying clothes	Laundry, patio, terrace, balcony
				13	Hygiene	Bathroom
			15	Circulation	Corridor, staircase	
			16	Storage	Closets, storage, pantry	
		Non-useful area			Walls, ducts, etc.	
Annexes	Gross area $A_{ga}$	Useful area $A_{ud} = A_{ga} \cdot i_{Ada/Aga}$	Non-useful area $A_{na}$	14	Being outside	Patio, terrace, balcony
				16	Storage	
				17	Garage	Garage
						Walls, ducts, etc.

**Table 7.9 – Area indexes**

Document		Indexes		
		Aid / Aud	Aud / Agd	Aua/Aga
PH	Min	0.667 (T1/1 dweller) 0.741 (T5/9 dwellers)	0.769	0.769
	Ref	0.769	0.800 (T1/1 dweller) 0.820 (T5/9 dwellers)	0.820
RRTHS		0.77	0.77	
RTHS		0.81	0.75	
<p>Key:</p> <p>Aid – dwelling inhabitable area            Aud – dwelling useful area            Agd – dwelling gross area            Aua – annex useful area            Aga – annex gross area</p>				
<p>Note:</p> <p>The Aua/Aga index (annex useful / annex gross area) did not exist in the RTHS, which used the Agh/Aud index (house gross area / dwelling useful area). However the PH considered that this index could be misleading in assessing a house's quality because "the annex might contain spaces that compensate for some limitations found in the dwelling." This is, in fact, the case of the Malagueira patio houses. Therefore, the Aua/Aga was preferred.</p>				

**Table 7.10 – Gross areas in Malagueira lots (after Table 4.3)**

Yard Location	Urban context	Balconies	Gross Areas (m <sup>2</sup> )								
			1 <sup>st</sup> Floor			2 <sup>nd</sup> Floor			Total		
			Int.	Ext.	Total	Int.	Ext.	Total	Int.	Ext.	Total
Lot			96.00	0.00	96.00	96.00	0.00	96.00	192.00	0.00	192.00
Front	any	yes	72.50	23.50	96.00	68.50	27.50 *23.50 **04.00	96.00	141.00	51.00	192.00
		no	68.50	27.50	96.00	68.50	0.00	72.50	141.00	23.50	164.50
Back	no back street	yes or no	72.97	23.03	96.00	56.80	39.20 *23.03 **16.17	96.00	129.77	62.23	192.00
		yes	72.50	23.50	96.00	68.50	4.00	72.50	141.00	27.50	168.50
	no	68.50	27.50	96.00	68.50	0.00	72.50	137.00	27.50	164.50	
<p>Note: * Area above the 1<sup>st</sup> floor's patio , **minimum useful exterior area</p>											

**7.4.4.7 Quality: function: spatiality: Space capacity**

A space's capacity is related to the number of dwellers that will be involved in the function assigned to it, and/or to the pieces of equipment required to perform its function. In the PH grammar, the capacity of a room for common use, such as the living-room, the dining room, the pantry, and the house clothing and general storage spaces, is

measured in terms of the total number of dwellers. The capacity of a room for private use, such as a bedroom, a studio, and a balcony is measured in terms of the number of its users. The capacity of a room for common use where equipment has an important role, such as a kitchen, and a laundry, is measured both in terms of the total number of dwellers, and the required pieces of equipment. The capacity of bathrooms is given by the pieces of equipment that it contains. The rules for determining a space's capacity are embedded in the rules that add spaces.

#### 7.4.4.8 Quality: function: spatiality: **Space Articulation**

A space's articulation refers to the way in which the space is articulated with other spaces. The space is isolated if the associated function is the sole or the primary function in the room. The space is delimited if it is in a room primarily designed for another function, but clearly delimited from the rest of the room, otherwise it is included.

(Figure 7.7) In Malagueira houses, sleeping, cooking, living, bathing, and exterior spaces always constitute isolated rooms, whereas dining, working, and storage spaces can be isolated, delimited, or included.

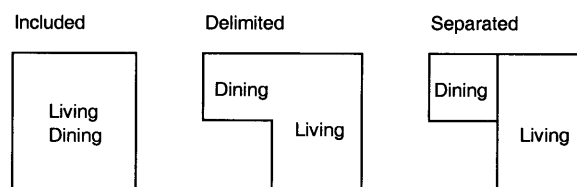


Fig. 7.7 - Articulation of spaces: included, delimited, and isolated.

#### 7.4.4.9 Quality: function: spatiality: **Dwelling capacity**

The dwelling capacity is the list of spaces included in the dwelling. The programmer provides a list of obligatory spaces required for the minimum quality level -- *the minimum*

*obligatory list*, based on the type and number of bedrooms, if the chosen degree of customization was *type*, or on detailed dweller information if it was *custom*. It also calculates the list of obligatory spaces -- *the initial obligatory list*, and the *list of optional spaces* for the quality level specified by the user. The user has then the possibility of changing that list to define the list of current spaces – *the current list*. The degree of freedom in making such changes is limited to respect the regulations and the available area. The user cannot delete or downgrade spaces in the desired list if they are in the minimum obligatory list, except in particular circumstances when the previous addition of an optional space covers up for the loss of that space, as explained further below. The user can upgrade obligatory spaces or add optional spaces as long as the area of all the desired spaces does not exceed the amount of available area. Such a restriction prevents the user from upgrading or adding rooms when the available area has already been exceeded. The exact changes that can be made regarding each space, as well the rules for determining the available area are explained in suit. Figure 7.8 shows the relation among the different lists of spaces.

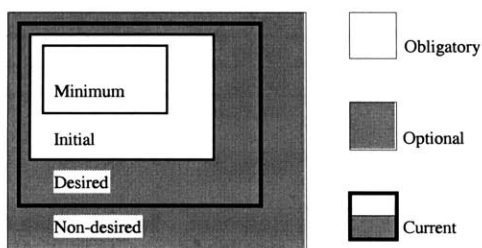


Figure 7.8 - List of spaces used in the generation of the housing program: obligatory (minimum, and initial), optional (desired, and non-desired), and current (obligatory, and desired).

### **Obligatory dwelling capacity**

The programmer determines the obligatory dwelling capacity depending on typological and morphological constraints, following Rules g54-g102, summarized in Table 7.11.

The programmer assigns weights to each space requirement, to express how important its inclusion in the house is. The designer will use these weights in the derivation of a solution. The assignment of weights follows the following rules. If a space is in the minimum list and in the current list for the selected quality level, its weight is 100. If the space is in both lists, but with different spatiality features in each, its weight is 90. If the user changes the spatiality features of such spaces, the user can assign a weight between 80 and 60. If the space is in the optional list, the user can assign a weight between 60 and 10.

**Table 7.11 - Obligatory spaces**

<b>Space</b>	<b>Articulation</b>	<b>Conditions</b>	<b>Room</b>
Patio	Isolated	none	own
Living-room	Isolated	none	own
Kitchen	Isolated	none	own
Bedroom *	Couple		own
	Double		own
	Single		own
Bathroom *	Bathtub	none	own
	Shower		own
	Lavatory		own
Formal dining *	Included	If quality min or med	living
	Delimited	If quality max	living
Informal dining *	Included	If quality min or med, bedrooms [2, 5], dwellers [3,9]	kitchen
	Delimited	If quality max, bedrooms [1, 5], dwellers [2,9]	kitchen
Children play *	Included	If custom	Bedroom, living, kitchen, circulation
Youth study *	Included	If quality min, med, or max	Bedroom, living
Adult work *	Included	If quality max	Bedroom, living
Laundry *	Included	If quality min	Kitchen
	Delimited	If quality med or max	
Pantry *	Included	If quality min, med, or max	Kitchen, circulation
Clothing storage *	Included	If quality min, med, or max	Circulation, bedrooms
Global storage *	Included	If quality min, med, or max	Circulation
Note: * See rules g54-102 or tables 7.12-7.17 to get the exact number and articulation of each space			



### Patio, kitchen, living room, and staircase

Rules 54-56 always add a patio, a kitchen, a living-room to the minimum ( $\alpha_{10}$ ) and initial obligatory lists of spaces ( $\alpha_{11}$ ), as well as to the current lists of spaces ( $\alpha_{13}$ ), and to the appropriate zones ( $\alpha_{14}$ ). Rule 57 adds a staircase if the number of floors ( $\alpha_8$ ) is two. These rules also calculate the dimensional requirements based on the number of dwellers and on the quality level and then update the areas ( $\alpha_{15}$ ,  $\alpha_{16}$ ), and the cost ( $\alpha_{24}$ ). They also introduce the corresponding topological requirements ( $\alpha_{16}$ ). The exact number and type of the remaining spaces varies according to typological and morphological constraints, as explained below.

### Bedrooms

If the user chooses to specify a customized house, the number and type of bedrooms is derived from the provided detailed dweller information as explained above. If the user chooses to specify a housetype, the programmer determines the number and type of bedrooms based on the total number of dwellers and bedrooms, according to Rules g58-71, summarized in Table 7.12. The rationale behind these rules derives from the traditional concept of nuclear family formed by the parents and their kids. A couple bedroom is always allocated, except in those cases in which the number of dwellers matches the number of bedrooms. The number and type of the remaining bedrooms is a matter of fitting the remaining dwellers into the remaining rooms as follows. The remaining dwellers are placed in single bedrooms until there are no more rooms available. Then, single bedrooms are turned into double bedrooms until all the dwellers are assigned to a bedroom. For instance, Rule g71 specifies that if the number of bedrooms is 5 and the number of users is 9, then there is 1 couple and 3 double and 1 individual bedrooms.

**Table 7.12 – Obligatory bedrooms**

<b>Bedrooms</b>	<b>Dwellers</b>	<b>Number</b>	<b>Type</b>
1	1	1	individual
	2	1	couple
2	2	2	individual
	3	1	couple
		1	individual
	4	1	couple
1		double	
3	4	1	couple
		2	individual
	5	1	couple
		1	double
		1	individual
	6	1	couple
2		double	
4	5	1	couple
		1	double
		1	individual
	6	1	couple
		1	double
		2	individual
	7	1	couple
		2	double
		1	individual
5	7	1	couple
		1	double
		3	individual
	8	1	couple
		2	double
		2	individual
	9	1	Couple
		3	Double
		1	individual

**Bathrooms**

The programmer adds bathrooms to the housing program according to rules g72-83, summarized in Table 7.14. Rules 72-74 add bathrooms to the minimum list of spaces based on the number of users and floors to ensure that the user cannot delete the required bathrooms. Rules 75-83 add bathrooms to the current list of spaces based on the number of dwellers and floors, but also on the quality level. There are three types of bathroom, depending on their capacity: bathtub, shower, and lavatory<sup>5</sup> (Table 7.13). All

<sup>5</sup> In rigor, a lavatory is not a bathroom. Nevertheless, the term bathroom is used in a broader sense in this work encompass all the spaces dedicated to personal hygiene functions.

the dwellings have a bathtub but they also can have a shower and a lavatory in some cases. For instance, rule g83 specifies that if the dwelling quality level is maximum and the number of users is between 6 and 9, then there is a bathtub, a shower, and a Lavatory.

**Table 7.9 - Type of bathrooms and their equipment**

Type (capacity)	Equipment
Bathtub	Bathtub, sink, toilette, bide
Shower	Shower, sink, toilette, bide
Lavatory	Sink, toilette

**Table 7.14 - Obligatory bathrooms**

Quality	Users	Floors	1st	2nd	3rd
any	1-9	any	bathtub		
min	2-6	2		lavatory	
min	7	any		lavatory	
min	8-9	any		shower	
med	2-5	2		lavatory	
med	6	any		lavatory	
med	7-9	any		shower	
med	8-9	any			lavatory
max	2-4	2		lavatory	
max	5	any		lavatory	
max	6-9	any		shower	
max	6-9	any			lavatory

Dining spaces

The programmer includes dining spaces in the housing program following rules g84-88, summarized in Table 7.15. Rule g84 adds an included informal dining space to the minimum list of spaces. Rules g85-88 add appropriate dining spaces to the initial obligatory and to current lists of spaces, as well as to the appropriate zone. There is no obligatory dining room. However, depending on the number of bedrooms and dwellers, there might be obligatory informal and formal dining spaces. These can be included in

the living or kitchen rooms, or constitute a delimited space adjacent to them. The formal dining space is always related to the living room, whereas the informal one is related to the kitchen. In case there is no informal dining space, the informal meals will take place in the formal dining space be it included, delimited or isolated. For instance, rule g85 specifies that if the quality level of the house is minimum or medium, and if the number of bedrooms is equal or bigger than 2, and the number of dwellers is equal or bigger than 3, then add an included informal dining space.

**Table 7.15 - Obligatory dining spaces**

Quality	Dining	Bedrooms	1		2		3			4			5			
		Dwellers	1	2	2	3	4	4	5	6	5	6	7	7	8	9
		Articulation														
min or med	Current	included				x	x	x	x	x	x	x	x	x	x	
		delimited														
	Formal	included	x	x	x	x	x	x	x	x	x	x	x	x	x	
		delimited														
max	Current	included														
		delimited		x	x	x	x	x	x	x	x	x	x	x	x	
	Formal	included														
		delimited	x	x	x	x	x	x	x	x	x	x	x	x	x	

Studio spaces

The programmer determines the inclusion of studio spaces following rules g89-96, summarized in Table 7.16. A studio shelters those activities that are secondary to the dwelling function such as the work of adults, the study of youngsters, and the play of children. There are, thus, three types of studio spaces: work, study, and play spaces. The rules to include these spaces in the housing program depend little on the degree of customization specified by the user. In the housetype option (rules 89-91) the programmer assumes that the occupants of the couple bedrooms are adults and the

occupants of the remaining are youngsters. In the customized option (rules g92-96), it checks the dwellers' ages to decide which type of studio space to include. For instance, rule g89 specifies that if the user chose to describe a housetype, and if the number of single bedrooms is smaller than the number of single youth study spaces, then include one single youth study space in a free single bedroom. For instance, rule g92 specifies that if the user chose to describe a customized house, and a dweller's age is smaller than 14, and the he/she does not share a bedroom, and he/she is not the user of an existing children play space, then add one included single children play space for this user in his bedroom.

**Table 7.16 – Obligatory studio spaces**

Dwelling capacity	Bedrooms	1		2			3			4			5		
		Dwellers	1	2	2	3	4	4	5	6	5	6	7	7	8
Space	single	1	-	2	1	-	2	1	2	3	2	1	3	2	1
capacity	double	-	-	-	-	1	-	1	-	-	1	2	1	2	3

Laundry space

The programmer adds laundry spaces following rules g97 and g98. The laundry space includes both wash and dry clothing spaces. The rules are very straightforward. Rule g97 specifies that if the housing quality level is minimum, then add one included laundry space to the minimum, initial obligatory, and current lists of spaces, as well as to the service zone. Rule g98 specifies that if the housing quality level is medium or maximum, then add one delimited laundry space to such lists.

Storage space

In the PH, the storage space required in the performance of functions is included in the area requirements of the corresponding spaces. This included the bedroom. In

addition, the PH required pantry, house clothing storage, and general storage spaces. In the RTHS, one of the documents that preceded the PH, the bedroom closet space was not included in the bedroom, but in the house clothing storage. Comparing the area demands of both documents, one can conclude that in the PH area demands are just slightly higher than those in the RTHS. (For instance, for a two-bedroom, three-dwellers house, the RTHS area requirement is 3.00-4.50m<sup>2</sup>, and the PH one is 3.20-5.54m<sup>2</sup>.) Although, Siza followed the RTHS after 1984, when they became mandatory, the analysis of his designs showed that the storage area allocated in the Malagueira designs is always above the RTHS minimum and around the PAHPA one. Therefore, to follow the PAHPA requirements in the programming grammar does not make it incompatible with the Malagueira design grammar.

The programmer adds pantry, house clothing storage, and general storage spaces following rules g99-101. The pantry can be a piece of furniture in the kitchen, or a room easily accessible from the kitchen, but it is not obligatory to include an isolated pantry. Rule g99 adds one included pantry space. The house clothing storage can be a furniture closet. Rule g100 adds one included house clothing storage space. The global storage is used to store larger items used in activities of the dwellers, such as bicycles, folding pieces of furniture, and so on. It is not obligatory to include a step-in closet. Rule g101 adds one included global storage space.

### Exterior spaces

The exterior spaces include balconies, exterior storage spaces, and garages. The programmer adds a balcony, only if the user chose this option in the description of the morphology (Rule g102). Exterior storage spaces and garages are not obligatory spaces.

### **Optional Dwelling Capacity**

The programmer automatically includes in the optional list those spaces that are not included in the obligatory list using rule g103. The user can then add such spaces to the current list. In addition, user can subtract spaces from the current list, provided that they are not included in the minimum list, if he/she is describing a customized house; or in the initial obligatory list, if he/she is describing a housetype. The user also can upgrade and downgrade the spaces in the current list. When the user manipulates this list, the programmer automatically updates the optional list to ensure that the maximum capacity that is allowed by the PH is not exceeded. The PH imposes such an upper limit because they target state-subsidized social housing, and therefore, they want to guarantee that the state is not subsidizing upper-class houses. This restriction was respected in the PH programming grammar because they also were respected in the design of Malagueira houses, and therefore, in the Malagueira designing grammar. Moreover, even when the PH did not impose an upper limit on the requirements, these were introduced in the grammar to prevent the user from specifying an over-constrained housing program. Nevertheless, these upper limits could be relaxed in the programming grammar if the designing grammar was changed.

The rules to add and subtract spaces from the current list are rules g104 through g136, summarized in Table 7.17. As shown in this table, the optional list includes only bathrooms, dining spaces, studios, storage spaces, and balconies. The rules to manipulate these optional spaces are described below.

**Table 7.17 - Optional spaces**

Operation	Spaces	Articulation	Conditions	Rooms where are located
Upgrade (obligatory spaces)	Bathroom *	bathtub		
		shower		
	Formal dining	Delimited	If quality min or med	
		Isolated		living
	Informal dining	Delimited	If quality min, med If quality max, bedrooms [1], users [1]	kitchen
		Isolated		
	Children play	Delimited		Bedroom, living, kitchen, circulation
		Isolated		Bedroom, living, kitchen, circulation
	Youth study	Delimited		Bedroom, living
		Isolated		
	Adult work	Delimited	If quality min or med	Bedroom, living
		Isolated	If quality max	
	Laundry	Delimited	If quality min	Kitchen
		Isolated		
Pantry	Isolated		<i>Kitchen, circulation</i>	
House clothing storage	Isolated		<i>Circulation, bedrooms</i>	
Global storage	Isolated		<i>Circulation</i>	
Delete (obligatory)	Informal dining	Included	If it exists formal dining isolated	
		Delimited		
Add (optional spaces)	Bathroom	bathtub	$n_{\text{bathrooms}} < n_{\text{bedrooms}+2}$	
		shower		
		lavatory		
	Informal dining	Included	If min or med, bedrooms [1], users [1,2]	kitchen
	Adult work	Included	$n_{\text{adult studio}} < n_{\text{adult dwellers}}$	
	Clothing storage	Included		Circulation, bedrooms
	Global storage	Included		Circulation
		Isolated		
Balcony	--	If $n_{\text{balcony}} < 2$		
Terrace	--	If $n_{\text{terrace}} < 2$		

### Bathrooms

The user can either upgrade any of the existing bathrooms, by turning a lavatory into a shower bathroom, or a shower bathroom into a bathtub bathroom (rule 104). Bathrooms can be downgraded and deleted, as long as the requirements of the minimum obligatory list are fulfilled. The user also can add new bathrooms, as long as the total number of bathrooms does not exceed the number of bedrooms plus two (rule 105). The rationale behind this rule is to consider that all the bedrooms can have a private bathroom, and



that there can be additional bathrooms linked the service and the living zones. The user also can subtract bathrooms when the number in the current list is above the minimum (rule g106). The programmer removes the bathroom from the optional list, if the number of bathrooms in the current list equals the maximum allowed (rule 107), and adds if it is below (rule 108).

### Dining spaces

The formal dining can be upgraded from included to delimited, and from delimited to isolated (rule g109). The formal dining cannot be deleted. The informal dining space can be only upgraded from included to adjacent, and if one delimited formal dining space exists, then the informal dining space can be downgraded to included (rule g110). An informal dining space can be added, if there is none (g111). The informal dining space can be deleted, if one isolated formal dining space exists (rule g112).

### Studio spaces

All the studio spaces -- children play, youth study, and adult workspace -- can be upgraded from included to delimited, and from delimited to isolated and later downgraded if the minimum obligatory list is respected (rules g114-115). The user can add adult workspaces as long as their number does not exceed the number of adult dwellers (rule g116), and then delete them (rule g117). The programmer removes the adult workspace from the optional list, if the number of adult spaces in the current list is equal to the number of adult dwellers (g118), and adds it back, if it is below (g119).

### Laundry spaces

The laundry can be upgraded from included to delimited, and from delimited to isolated and then downgraded, to the initial type (rule g122). The laundry cannot be deleted.

### Storage spaces

The storage spaces—pantry, house clothing storage, and global storage—can be upgraded from included to isolated and then downgraded (rules g121-g123). The user also can add and delete house clothing and global storage spaces (g124-127). The programmer deletes the house clothing storage and the global storage spaces from the optional list, if the current capacity reaches the maximum, which equals the number of dwellers plus two, and adds them back, if it gets below (g128-131).

### Exterior spaces

The user can upgrade and downgrade a balcony by changing its capacity, as long as it does not exceed the number of dwellers plus two (g132). The user can add a balcony to the current list, if he specified such in the morphology and there are no more than two in the current list (g133). The user also can delete a balcony, if there are two (g134). The programmer deletes balcony from the optional list, if the current number equals two (g135), and adds it back if it is below (g136). No rules were included to add garages and exterior storage spaces, but they could be easily added.

#### **7.4.4.10 Quality: function: spatiality: Space spaciousness – width, height, and area**

The spaciousness of a space depends on the dimensional requirements of that space, namely its width, height, and area. When a space is introduced in the housing program, the programmer determines its dimensional requirements based on its capacity (dwellers or equipment), articulation, and quality level. It does so by using functions with the spaces' name, capacity, articulation, and quality level as arguments to retrieve the appropriate values from tables  $\beta_1$  (width),  $\beta_2$  (height), and  $\beta_3$  (area). Tables  $\beta_1$  and  $\beta_2$  are rearranged into Tables 7.18 through 7.30. The dimensional requirements shown

correspond to the PH requirements with the few exceptions indicated below the tables. The analysis of the Malagueira houses confirmed that their dimensions respect the PH requirements with very few exceptions. Therefore, the compatibility between the PAHPA grammar and the Malagueira grammar also is guaranteed at this level.

**Table 7.18 Dimensional requirements of bedrooms**

Feature ( $\beta$ ) (units)	Capacity	Quality		
		Min	Med	Max
Width ( $\beta_1$ ) (m)	Couple	2.70	3.00	3.30
	Double	2.10	2.40	2.70
	Single	2.10	2.40	2.70
Height ( $\beta_2$ ) (m)	Any	2.40	2.60	2.70
Area ( $\beta_3$ ) (m <sup>2</sup> )	Couple	10.5	11.5	12.0
	Double	9	10.0	11.0
	Single	5.5	6.0	6.5

Note: The area of bedrooms in the Malagueira houses are in accordance with the PH requirements.

**Table 7.19 - Dimensional requirements of living rooms**

Feature ( $\beta$ ) (units)	Capacity (dwellers)	Quality		
		Min	Med	Max
Width ( $\beta_1$ ) (m)	1-3	2.70	2.85	3.00
	4-5	2.85	3.00	3.30
	6-7	3.00	3.30	3.60
	8-9	3.30	3.60	3.90
Height ( $\beta_2$ )	1-9	2.40	2.60	2.70
Area ( $\beta_3$ ) (m <sup>2</sup> )	1-2	6.00 (< 10.00)	7.5 (< 10.00)	9.00 (< 10.00)
	3	7.50 (< 10.00)	9.00 (< 10.00)	10.50
	4-5	9.00 (< 10.00)	10.5	12.00
	6	10.50	12.00	15.5
	7	12.00 (16.00)	15.50	19.00
	8-9	15.50 (16.00)	19.00	22.5

Note: (< x.xx) the RGEU requirements.

The Malagueira houses are expandable, but the areas of those rooms that are common to all the variations (1-5 bedroom) of a particular type do not change from variation to variation. Therefore, the areas of living-rooms of the smallest variation are as large as those of the biggest variation. The smallest living room area in a Malagueira house was 16.35m<sup>2</sup> (Subtype Da), which is above the 15.50 minimum required for a five-bedroom, nine-dweller house.

**Table 7.20 - Dimensional requirements of informal dining spaces**

Feature ( $\beta$ ) (units)	Articulation	Capacity (Dwellers)	Quality		
			Min	Med	Max
Width ( $\beta_1$ ) (m)		1-3	2.15	2.35	2.55
		4-5	2.15	2.35	2.75
		6-7	2.15	2.55	2.75
		8-9	2.35	2.55	2.95
Height ( $\beta_3$ ) (m)		1-9	2.40	2.60	2.70
Area ( $\beta_3$ ) (m <sup>2</sup> )	Included	1	-	-	-
		2	-	-	2.50
		3	2.00	2.50	3.00
		4	2.00	3.00	4.00
		5	2.50	3.50	4.50
		6	3.00	4.00	5.00
		7	3.50	4.50	5.50
		8	4.00	5.00	6.00
		9	4.50	5.50	6.50
	Delimited Isolated*	1	-	-	-
		2	-	-	3.00
		3	2.50	3.00	3.50
		4	2.50	3.50	4.50
		5	3.00	4.00	5.00
		6	3.50	4.50	5.50
		7	4.00	5.00	6.00
		8	4.50	5.50	6.50
		9	5.00	6.00	7.00

Note: The PH did not indicate the area requirements of isolated informal dining spaces. In this work, they equal those of delimited spaces.

The area of the dining spaces (transitional spaces) in the Malagueira houses is above the limits of the requirements considered in this table. The smallest dining space's area is 5.51 m<sup>2</sup>, which is above the required 5.00 m<sup>2</sup>.

**Table 7.21 - Dimensional requirements of formal dining spaces**

Feature ( $\beta$ ) (units)	Articulation	Capacity (dwellers)	Quality		
			Min	Med	Max
Width ( $\beta_1$ ) (m)	(delimited / isolated)	1-3	2.40	2.70	3.10
		4	2.40	2.70	3.30
		5	2.40	2.90	3.30
		6	2.60	2.90	3.50
		7	2.60	3.10	3.50
		8	2.80	3.10	3.70
		9	2.80	3.30	3.70
Height ( $\beta_2$ ) (m)		1-9	2.40	2.60	2.70

Area ( $\beta_3$ ) (m <sup>2</sup> )	Included	1	5.00	6.00	6.50
		2	5.00	6.00	6.50
		3	5.00	6.00	7.50
		4	5.00	7.00	8.50
		5	6.00	8.00	9.50
		6	7.00	9.00	10.50
		7	8.00	10.00	11.50
		8	9.00	11.00	12.50
		9	10.00	12.00	13.00
	Delimited	1	5.50	6.50	7.00
		2	5.50	6.50	7.00
		3	5.50	6.50	8.00
		4	5.50	7.50	9.00
		5	6.50	8.50	10.00
		6	7.50	9.50	11.00
		7	8.50	10.50	12.00
		8	9.50	11.50	13.00
		9	10.5	12.50	14.00
	Isolated	1	-	-	-
		2	-	-	7.50
		3	-	-	8.50
		4	-	8.00	9.50
		5	7.00	9.00	10.50
		6	8.00	10.00	11.50
		7	9.00	11.00	12.50
		8	10.00	12.00	13.50
		9	11.00	13.00	14.00

### 7.22 – Dimensional requirements of kitchens

Feature ( $\beta$ ) (units)	Capacity		Quality		
	Bedrooms	Dwellers	Minimum	Medium	Maximum
Width ( $\beta_1$ ) (m)	I or L-shaped counter		1.70	1.70	1.80
	Double I-shaped counter		2.30	2.30	2.40
	U-shaped counter		2.30	2.30	2.40
Height( $\beta_2$ )(m)	0-5	1-9	2.40	2.60	2.70
Area ( $\beta_3$ )(m <sup>2</sup> )	0	1	4.50	5.00	5.50
	1	1	4.50	5.00	5.50
	1	2	4.50	5.50	6.00
	2	2	4.50	5.50	6.00
	2	3	4.50	5.50	6.00
	2	4	5.00	6.00	6.50
	3	4	5.00	6.00	6.50
	3	5	5.00	6.50	7.00
	3	6	5.00	6.50	7.00
	4	5	5.00	6.50	7.00
	4	6	5.00	6.50	7.00
	4	7	5.50	7.00	7.50
	5	7	5.50	7.00	7.50
	5	8	5.50	7.00	7.50
	5	9	5.50	7.00	7.50

**Table 7.23 – Dimensional requirements of bathrooms**

Feature ( $\beta$ ) (units)	Capacity	Quality		
		Min	Med	Max
Width ( $\beta_1$ ) (m)	Bathtub	1.50	1.70	1.90
	Shower	1.30	1.40	1.60
	Lavatory	0.80	0.80	0.90
Height ( $\beta_2$ ) (m)	Any	2.20	2.40	2.60
Area ( $\beta_3$ ) (m <sup>2</sup> )	Bathtub	4.00 (3.50)	4.50	5.00
	Shower	2.50 (2.50)	2.50	3.00
	Lavatory	1.50 (1.00)	1.50	2.00

Note: (x.xx) the RGEU requirements.

In some cases, the dimensions of bathrooms in Malagueira houses are above the RGEU requirements, but below the PH requirements.

**Table 7.24 - Dimensional requirements of studio spaces**

Space	Feature ( $\beta$ ) (units)	Articulation	Capacity (dwellers)	Quality		
				Min	Med	Max
Playing (children)	Width ( $\beta_1$ ) (m)			1.30	1.40	1.60
	Height ( $\beta_2$ ) (m)			2.40	2.60	2.70
	Area ( $\beta_3$ ) (m <sup>2</sup> )	Included	Single	1.00	1.50	2.00
			Double	1.50	2.50	3.50
		Delimited	Single	3.00	3.50	4.00
Double			3.50	4.00	4.50	
Studying (youth)	Width ( $\beta_1$ ) (m)			1.20	1.30	1.40
	Height ( $\beta_2$ ) (m)			2.40	2.60	2.70
	Area ( $\beta_3$ ) (m <sup>2</sup> )	Included	Single	1.00	1.50	2.00
			Double	1.50	2.50	3.50
		Delimited	Single	1.50	2.00	2.50
Double			3.00	4.00	5.00	
Working (adult)	Width ( $\beta_1$ ) (m)			1.30	1.40	1.60
	Height ( $\beta_2$ ) (m)			2.40	2.60	2.70
	Area ( $\beta_3$ ) (m <sup>2</sup> )	Included		1.00	2.00	3.00
		Delimited		2.00	3.00	4.00

**Table 7.25 – Dimensional requirements of laundry spaces**

Feature ( $\beta$ ) (units)	Articulation	Dwellers	Quality		
			Min	Med	Max
Width ( $\beta_1$ ) (m) (parallel to the counter)	Included	Single counter	0.90	1.20	1.50
		Opposed counters	0.60	0.60	0.90
	Delimited	Single counter	1.20	1.20	1.50
		Opposed counters	0.60	0.60	0.90
	Isolated	Single counter	1.20	1.20	1.50
		Opposed counters	0.90	1.00	1.10

Length ( $\beta_1$ ) (m) (perpendicular to the counter)	Included	Single counter	1.30	1.40	1.50
		Opposed counters	2.30	2.30	2.40
	Delimited	Single counter	1.40	1.50	1.50
		Opposed counters	2.30	2.40	2.40
	Isolated	Single counter	1.50	1.50	1.60
		Opposed counters	2.30	2.30	2.40
Height ( $\beta_2$ ) (m)	Any	1-9	2.20	2.40	2.60
Area ( $\beta_3$ ) (m <sup>2</sup> )	Included	1-3	1.17 1.0 + 1.0	1.5 + 1.0	2.0 + 1.0
		4-6	1.0 + 1.0	1.5 + 1.0	2.0 + 1.0
		7-9	1.0 + 1.0	2.0 + 1.0	2.5 + 1.0
	Delimited Isolated	1-3	-	-	-
		4-6	1.5 + 1.0	2.0 + 1.0	2.5 + 1.0
		7-9	1.5 + 1.0	2.5 + 1.0	3.0 + 1.0
Note: The dimensions of laundries in the Malagueira houses are in accordance with the PH requirements, except for Type Bb's laundry whose width is slightly below (2.26 < 2.30).					

**Table 7.26 – Dimensional requirements of pantry spaces**

Feature ( $\beta$ ) (units)	Articulation	Capacity (dwellers)	Quality		
			Min	Med	Max
Width ( $\beta_1$ ) (m)		Any	0.90	1.20	1.50
Height ( $\beta_2$ ) (m)		Any	2.20	2.40	2.60
Area ( $\beta_3$ ) (m <sup>2</sup> )	Included Isolated*	1	0.50	0.75	1.00
		2	0.50	0.75	1.25
		3-4	0.75	1.00	1.50
		5	1.00	1.25	1.75
		6	1.00	1.25	1.75
		7	1.25	1.25	1.75
		8	1.25	1.50	2.00
		9	1.50	1.75	2.00
Note: The dimensions of isolated pantries in Malagueira houses are in accordance with the PH requirements.					

**Table 7.27 – Dimensional requirements of general storage spaces (step-in closet)**

Feature ( $\beta$ ) (units)	Articulation	Capacity (dwellers)	Quality		
			Min	Med	Max
Width ( $\beta_1$ ) (m)	Isolated	1-9	0.90*	1.20	1.50
Height ( $\beta_3$ ) (m)	Isolated	1-9	2.20	2.40	2.60
Area ( $\beta_3$ ) (m <sup>2</sup> )	Isolated	1-2	1.00	1.00	1.25
		3	1.00	1.00	1.50
		4	1.00	1.25	1.50
		5-6	1.00	1.25	1.75
		7-8	1.00	1.50	2.00
		9	1.00	1.75	2.25

**Table 7.28 - Dimensional requirements of house Clothing storage (furniture)**

Feature ( $\beta$ ) (units)	Articulation	Capacity (dwellers)	Quality		
			Min	Med	Max
Width ( $\beta_1$ ) (m)	Included	1-9	0.90	1.20	1.50
Height ( $\beta_2$ ) (m)	Included	1-9	2.20	2.40	2.60
Area ( $\beta_3$ ) (m <sup>2</sup> )	Included	1	0.50	0.75	0.75
		2	0.50	0.75	1.00
		3	0.75	1.00	1.25
		4	0.75	1.25	1.50
		5	1.00	1.50	1.75
		6	1.00	1.50	2.00
		7	1.25	1.75	2.25
		8	1.25	2.00	2.50
		9	1.50	2.00	2.75

Note: The analysis of Malagueira houses showed that Siza designed furniture closets whose smallest dimension was 0.45 m and isolated closets whose smallest dimension was 0.75 m.

**Table 7.29 - Dimensional requirements of balconies**

Feature ( $\beta$ ) (units)	Capacity (dwellers)	Quality		
		Min	Med	Max
Width ( $\beta_1$ ) (m)	1	-	0.60	1.25
	2	-	0.60	1.25
	3	-	0.60	1.25
	4	-	1.25	1.50
	5	-	1.25	1.50
	6	-	1.25	1.50
	7	-	1.50	1.80
	8	-	1.50	1.80
	9	-	1.50	1.80
Height ( $\beta_2$ ) (m)	1-9	-	-	-
Area ( $\beta_3$ ) (m <sup>2</sup> )	1	-	1.50	2.50
	2	-	1.50	2.50
	3	-	1.50	2.50
	4	-	2.50	3.50
	5	-	2.50	3.50
	6	-	2.50	3.50
	7	-	3.00	4.00
	8	-	3.00	4.00
	9	-	3.00	4.00

**Table 7.30 – Dimensional requirements of stairs**

Feature ( $\beta$ ) (units)	Type	Quality		
		Minimum	Medium	Maximum
Width ( $\beta_1$ ) (m)	Any	0.70	0.80	0.90
Height ( $\beta_2$ ) (m)	Any	1.90	1.90	1.90
Area ( $\beta_3$ ) (m <sup>2</sup> )	I-shaped	3.88	4.55	5.25
	L-shaped	4.08	4.77	5.55
	U-shaped	4.88	5.87	6.94



#### 7.4.4.11 Quality: function: **Topology**

Topology qualifies the relations between any two spaces in terms of their distance, and communication. It affects privacy and accessibility. If a space is added to the current list of spaces, the programmer adds to the housing program the required and recommended topological relations involving that space and other included spaces. The required and recommended topological relations are shown in Tables 7.31 and 7.32.

Rules g137-146 add required topological requirements. For instance, rule g151 specifies the required topological requirement between the kitchen and the laundry, depending on their specified articulation (isolated, delimited, and included) and on the dwelling quality level (minimum, medium, and maximum). For instance, if a kitchen and an isolated laundry are included in the current list of spaces, and if the quality level is minimum, then the laundry should be preferably close to, adjacent to, or communicate with the kitchen through a door in this order. If the quality level is maximum, then the order of preference is reversed. This means that if the quality level is maximum, then designer will first try to place the laundry in a way they communicate through a door. If it fails, then it will try to place them adjacent to, and then close to each other.

Rules 147-152 add recommended topological requirements. The user can change recommended requirements, or create new ones between any two spaces included in the current list of spaces (rule g156). If a space is subtracted from the current list of spaces, the programmer removes the topological relations involving that space (rule g157).

**Table 7.31 – Required topological relations among spaces**

Space 1	Space 2	Articulation	Quality level	Relation	Weight	
Kitchen	Laundry	Isolated	Minimum	Close, adjacent, door	100	
			Medium	Adjacent, close, door	100	
			Maximum	Door, adjacent, close	100	
		Delimited	Any	Passage	100	
		Included	Any	Merged	100	
		Informal dining	Isolated	Minimum	Close, adjacent, window, door	100
	Medium			Adjacent, close, window, door	100	
	Maximum			Door, window, adjacent, close	100	
	Delimited		Any	Passage	100	
	Included		Any	Merged	100	
	Formal dining		Isolated	Minimum	Close, adjacent, window, door	100
		Medium		Adjacent, close, window, door	100	
		Maximum		Door, window, adjacent, close	100	
		Delimited	Any	Passage	100	
		Included	Any	Merged	100	
		Pantry	Isolated	Minimum	Close, adjacent, door	100
	Medium			Adjacent, close, door	100	
	Maximum			Door, adjacent, close	100	
	Included		Any	Merged	100	
	Patio, street		Isolated	Minimum	Adjacent, close, window, door	100
				Medium	Window, close, adjacent, door	100
		Maximum		Door, window, adjacent, close	100	
		Living-room	Isolated	Minimum	Close, adjacent, window, door	100
				Medium	Adjacent, close, window, door	100
Maximum				Door, window, adjacent, close	100	
Living-room	Formal dining		Minimum	Close, adjacent, window, door	100	
			Medium	Adjacent, close, window, door	100	
			Maximum	Door, window, adjacent, close	100	
	Delimited	Any	Passage	100		
	Included	Any	Merged	100		
	Living-room	Patio	Minimum	Adjacent, close, window, door	100	
Medium			Window, close, adjacent, door	100		
Maximum			Door, window, adjacent, close	100		
Laundry		Minimum	Close, adjacent, window, door	100		
		Medium	Adjacent, close, window, door	100		
		Maximum	Door, window, adjacent, close	100		
Global storage	Staircase, corridor	Minimum	Close, adjacent, merged, door	100		
		Medium	Adjacent, merged, close, door	100		
		Maximum	Merged, door, adjacent, close	100		

**Table 7.32 – Recommended topological relations among spaces**

Space 1	Space 2	Relation	Weight
Bedroom	First floor	on	80
	Living-room	same floor	80
	Kitchen	same floor	80
	Bathtub	same floor	80
Lavatory	Living-room	same floor	80
	Kitchen	same floor	80

#### 7.4.4.12 Quality: Aesthetics: proportion

The only included aesthetic feature is proportion because the experimental results showed that it was not embedded into the Malagueira grammar. However, unlike other dimensional features such as width, height, and area, proportion impacts aesthetics more than function. Therefore, it was included in the programming grammar as a sub-feature of aesthetics. It also would be possible to extend the grammar and include other categories such as symmetry, rhythm, and so on.

Proportion is the ratio between two dimensions of shapes. In architecture, several proportions are usually considered. Two of these proportions are the one between the width and the length of rooms and the one between the width and the height of rooms. Other common proportions are the ones between the dimensions of façade elements. Proportioning systems refer to the relations among the proportion of shapes in a composition. For instance, it might refer to the proportion among windows or among windows and walls or other elements in the design of a façade. The use of certain proportions or the use of a proportioning system is said to contribute to the beauty of a design. Andrea Palladio, for instance, recommended rooms with proportions  $\sqrt{2}:1$ ,  $1:1$ ,  $4:3$ ,  $3:2$ ,  $5:3$ , and  $2:1$  in his *Quattro Libri* (Palladio, 1980). Table 7.33 shows some commonly suggested proportions in architectural treaties. (March 1998)

**Table 7.33 - Proportions commonly used in architecture**

Ordered by <i>divider</i>		Ordered by ratio	
Proportion	Ratio	Proportion	Ratio
1:1*	1.0	1:1	1.0
1: $\sqrt{2}$ *	0.7071	5:6	0.83(3)
1:2*	0.5	4:5	0.8
1:3	0.33(3)	3:4	0.75
1:4	0.25	1: $\sqrt{2}$	0.7071
2:3*	0.66(6)	2:3	0.66(6)
2:5	0.4	3:5	0.6
3:4*	0.75	1:2	0.5

3:5 *	0.6	2:5	0.4
4:5	0.8	1:3	0.33(3)
5:6	0.83(3)	1:4	0.25
Note: * Proportions cited by Palladio in the Quattro Libri.			

Siza confirmed proportion as an important design quality in the design of Malagueira houses, in our conversations with him. Siza mentioned that there was the concern to design rooms with certain proportions, such as 1:1, 2:3, 3:4, and so on. A careful analysis of the floor plans was undertaken to confirm this assertion. In this analysis, two issues needed to be considered: which proportions to take into account, how to measure the dimensions to calculate the proportions.

As the Malagueira floor plans were generated by a dissection process, two proportions needed to be considered. One was the proportion of the dissection, and the other was the proportion of the resulting rectangles (Figure 7.9). The adequate proportioning of the dissection assures a harmonious relation among rooms, whereas the adequate proportioning of the resulting rectangles influences the beauty of each individual room.

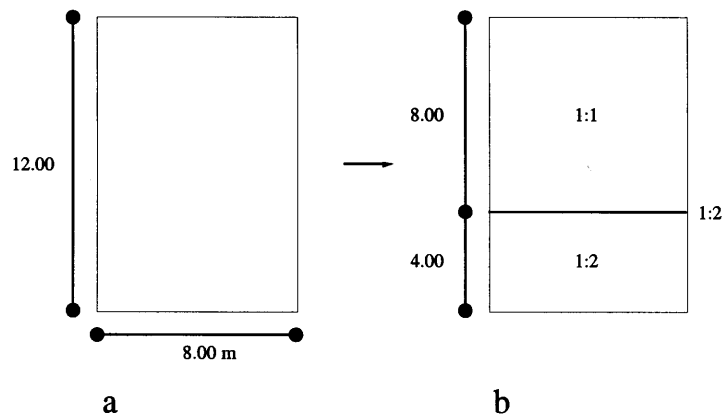


Figure 7.9 - Two proportions considered in the dissection of rectangles: the proportion of the dissection, and the proportion of the resulting rooms.

The measurement issue consisted of two problems. The first was to choose whether to consider preliminary or construction drawings. The preliminary drawings are closer to the initial stages of the design and, therefore, they are theoretically better to identify the original ideas behind the genesis of form. The construction drawings, on the other hand, potentially represent a compromise between aesthetic, functional (e.g. the space required to perform a function) and construction requirements (e.g. modular coordination), and so they can identify other forces that contributed to the final form. Comparative analyses of both drawings provided important clues in modeling the process of generating Malagueira plans. The second problem was how to take into account wall thickness. That is, should the dimensions of rooms be measured from the axes of the walls (Figure 7.10a), or should they be measured from their surfaces (Figure 7.10b)? Furthermore, if rooms were measured from their surface, should the thickness of the finishing material be considered (Figure 7.11)? In preliminary studies, the degree of abstraction of the drawings does not pose such a problem, but in construction drawings, there is a difference between drawings showing the superstructure (for instance, non-plastered walls), and drawings showing the finished materials (plastered walls). The former drawings are used at an early construction stage for building the walls, pillars, pavements, and other structural elements, whereas the latter are used for finishing the building (plastering, tiling, and so on). The different measurement strategies yield different results, especially when small rooms are involved.

Preliminary drawings were not available for all the housetypes, and therefore a complete comparison between preliminary and construction drawing was not possible.

Preliminary design drawings were available for the first frontyard and backyard houses that Siza designed, namely subtypes Aa and Ba. These houses were never built and they were, in fact, preliminary studies of types A and B houses.

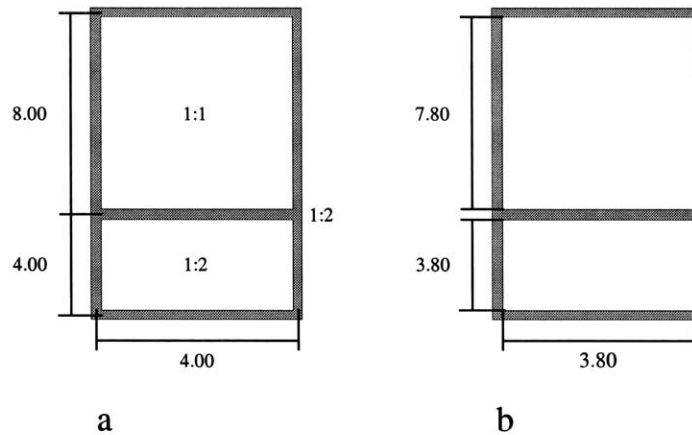


Figure 7.10 - Two ways of measuring the dimensions in a dissected rectangle: from the axes, or from the surfaces of walls.

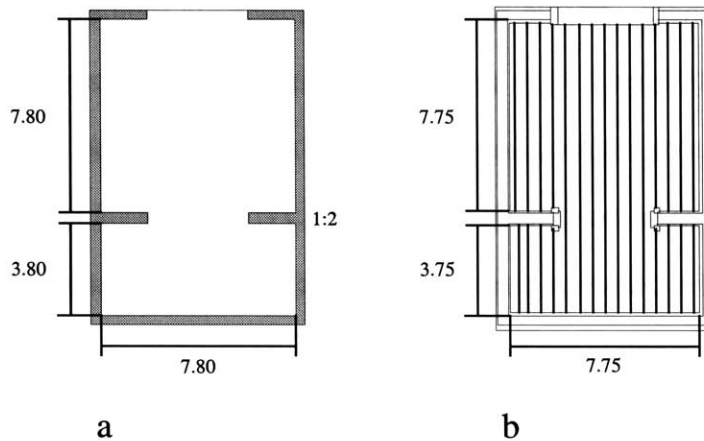


Figure 7.11 - Schematic representation of the two types of construction drawings, one showing just the elements of the superstructure (a), and the other the same elements with finishing materials applied (b).

The first step in the search for revealing Siza's strategy for wall placement was to overlay the floor plans on a one by one meter grid (Figure 7.12). This procedure revealed that the wall placement did not strictly follow the grid. The only wall consistently placed on the grid in all the housetypes, with the grid line cutting through the middle of the wall, was the first level dissection wall (dissection into inside and outside zones). In the majority of cases, the second level dissection walls (dissection into functional zones) is also placed on the grid line, but sometimes in the middle, and some



But did the same happen to the proportioning requirement? The survey of the construction drawings revealed that rare are the cases in which there is an exact match of the proportions in the drawing with any of the canonic types, if the rooms are measured from the walls surface.

In housetypes Aa and Ba, there is a strict compliance with proportioning rules, if the measurements are made from the grid lines to which the walls are related. (Fig. 7.14) In housetype Aa, the wall between the corridor and the patio recess is on the "wrong" place, but that is because building regulations are very strict regarding the corridor width, which needs to be 1.10 m and, therefore, could not be related to the grid at all. All the other proportions are one of Siza's favorite. The main theme is the dissection into 2:3 (the same proportion as the lot) and 3:4 rectangles through 1:2 and 2:3 dissections. In housetype Ba, there is a similar use of canonic proportions although the grid is not followed as strictly as in Aa. The predominant theme, however, is the dissection into 1:1 and 1:2 rectangles through 1:2 and 3:5 dissections. The analyses of Aa and Ba drawings (preliminary drawings) suggest that Siza's strategy comprised two steps: (1) use canonic dissections to obtain canonic rectangles by following the grid and, (2) place the walls in the middle, on the left, or on the right of the dissecting lines, to conform with functional requirements.

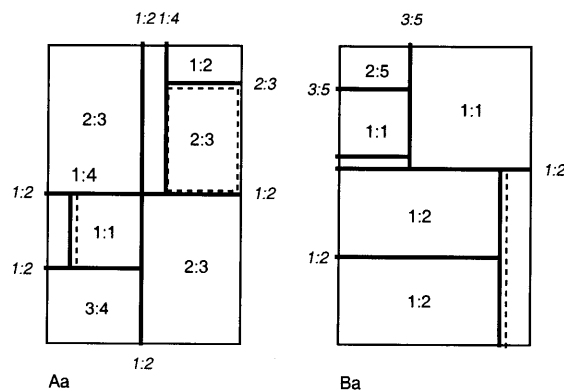


Figure 7.14 - The canonic proportions of the dissections (italics) and the rooms in types Aa and Ba, prior to the consideration of wall thickness.



The analysis of the construction drawings of types Ab, Ac (Figure 7.15), and Bb reveal a concern for obtaining rooms with final good proportions, more than for respecting canonic dissections obtained by following the grid. The deviations of the final proportions of rooms from the canonic types is smaller in these types (0.0070), than in Aa (0.0131) and Ba (0.0150). In types Ca, Da (Figure 7.16), and E, the situations differ. In Ca and E there is a bigger emphasis on following the grid, whereas in Da, the emphasis is on the final proportion of rooms, which deviate very little from the canonic proportions (0.0053). These results suggest two additional steps in Siza's proportioning strategy: (3) once walls have been placed, shift them around to better accommodate functional requirements and (4) shift them again to bring the final proportions of rooms closer to the canonical proportions.

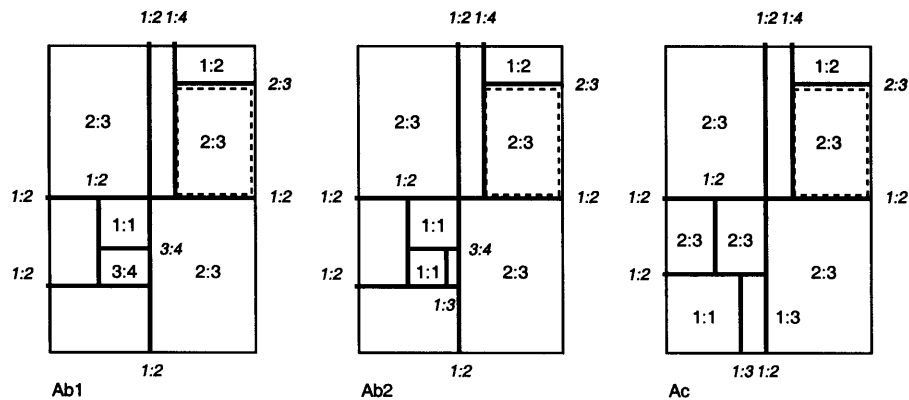


Figure 7.15 – The dissection and room proportions in subtypes Ab, and Ac.

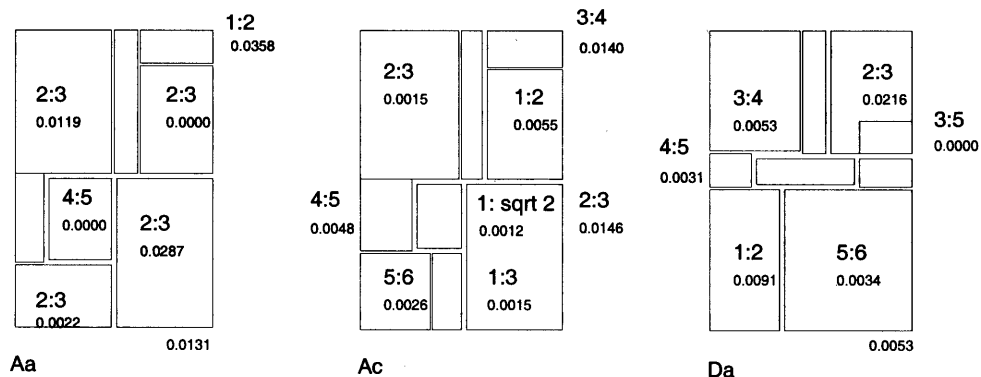


Figure 7.16 - The proportions of the final rooms in types Aa, Ac, and Da, and their deviations from the strictly canonic types.

In summary, results suggest that room dimensioning stems from interaction between two types of requirements, one related to function, and the other related to aesthetics (proportion), thereby supporting the inclusion of proportion as a design requirement. The programmer introduces the default weights assign to proportions through Rule g164. The user can then change such weights using rule g165.

#### **7.4.4.13 Cost**

In the PH three different costs are considered: construction, exploitation, and maintenance. The land cost also is frequently considered in the design of housing. However, the only cost considered in the programming grammar is the construction cost ( $\alpha_{24}$ ) because it is relatively easy to estimate, and to budget. Moreover, many institutions regularly publish tables with construction costs, thereby making it easier to make accurate estimations. The land cost is not important in the context of the PAHPA-Malagueira grammar because there is only one type of land plot, and the lot is an independent variable in the top-down approach followed by Siza. (Once a lot is selected by the user, changing the lot is not a design option.)

The programmer estimates the construction cost by multiplying the area of each space by the cost per area unit (square meter) of that space, and then adding up the cost of all spaces. The cost per area unit depends on whether the space is a service space (kitchen or bathroom), an enclosed non-service space (bedroom, living-room, etc.), a covered space (balcony or patio with overhangs), or an external space (balcony or patio). (Table 7.34) There are no specific cost rules. The cost is updated by the rules that add, subtract, upgrade, and downgrade spaces. The user controls the total cost associated with a housing program by controlling the added spaces.

**Table 7.34 – Cost per area unit**

	<b>Space</b>			
	Service	Enclosed	Covered	Exterior
Cost (USD/m <sup>2</sup> )	\$600.00	\$500.00	\$400.00	\$300.00

#### **7.4.4.14 Activating the designing grammar**

When the user hits the 'send' button on the interface, the programmer creates an empty design description (Rule 166), except for the inclusion of the appropriate available area ( $\alpha_{15}$ ), the weights that the user assigned to the proportions ( $\alpha_{23}$ ) and to the qualities ( $\alpha_{23}$ ), and the inclusion of rule R0 in the history of the design derivation ( $\alpha_{25}$ ). The features in the design description are the same in the housing program, but they are identified with the Greek letter  $\delta$  to differentiate them. The rules in the designing grammar, manipulated both descriptions as explained below.

Table 7.35 - The programming grammar rules

**START**

**g0: Initializing description**

$\alpha_1 \leftarrow \langle \text{nil}, \text{nil}, \text{nil}, \text{nil} \rangle$   
 $\alpha_2 \leftarrow \langle \text{nil}, \text{nil}, \text{nil}, \text{nil} \rangle$   
 $\alpha_3 \leftarrow \langle \text{nil} \rangle$   
 $\alpha_4 \leftarrow \emptyset$   
 $\alpha_5 \leftarrow \emptyset$   
 $\alpha_6 \leftarrow \langle \text{nil} \rangle$   
 $\alpha_7 \leftarrow \langle \text{nil} \rangle$   
 $\alpha_8 \leftarrow \langle \text{nil} \rangle$   
 $\alpha_9 \leftarrow \langle \text{nil} \rangle$   
 $\alpha_{10} \leftarrow \emptyset$   
 $\alpha_{11} \leftarrow \emptyset$   
 $\alpha_{12} \leftarrow \emptyset$   
 $\alpha_{13} \leftarrow \emptyset$   
 $\alpha_{14} \leftarrow \langle [\text{patio}, \emptyset], [\text{service}, \emptyset], [\text{living}, \emptyset], [\text{sleeping}, \emptyset], [f2, \emptyset] \rangle$   
 $\alpha_{15} \leftarrow \langle (\text{available}, (0, 0, 0, 0)), (0, 0, 0, 0), (0, 0, 0, 0), 0 \rangle$   
 $\alpha_{16} \leftarrow \langle (\text{used}, (0, 0, 0, 0)), (0, 0, 0, 0), (0, 0, 0, 0), 0 \rangle$   
 $\alpha_{17} \leftarrow \emptyset$   
 $\alpha_{18} \leftarrow 0$   
 $\alpha_{19} \leftarrow \emptyset$   
 $\alpha_{20} \leftarrow \emptyset$   
 $\alpha_{21} \leftarrow \emptyset$   
 $\alpha_{22} \leftarrow \langle [1:1, 0], [1:\sqrt{2}, 0], [1:2, 0], [1:3, 0], [1:4, 0], [2:3, 0], [2:5, 0], [3:4, 0], [3:5, 0], [4:5, 0], [5:6, 0] \rangle$   
 $\alpha_{23} \leftarrow \langle [\text{function}, 0], [\text{spatiality}, 0], [\text{capacity}, 0], [\text{articulation}, 0], [\text{spaciousness}, 0], [\text{topology}, 0], [\text{aesthetics}, 0] \rangle$   
 $\alpha_{24} \leftarrow 0$   
 $\alpha_{25} \leftarrow \emptyset$   
 $\beta_1 \leftarrow \beta_1$   
 $\beta_2 \leftarrow \beta_2$   
 $\beta_3 \leftarrow \beta_3$   
 $\beta_4 \leftarrow \beta_4$   
 $\beta_5 \leftarrow \beta_5$   
 $\beta_5 \leftarrow \beta_5$

**CONTEXT**

**Lot: specifying the lot features**

**g1: default lot context (programmer)**

$\alpha_0 \leftarrow \alpha_0 + \langle 8, 12, 96 \rangle$

**Urban context: specifying the urban context**

**g2: default urban context (programmer)**

$\alpha_1 \leftarrow \alpha_1 - \langle ?\text{context}_r, ?\text{context}_l, ?\text{context}_b, ?\text{context}_r \rangle,$   
 $\quad ?\text{context}_l, ?\text{context}_l, ?\text{context}_b, ?\text{context}_r \in \{\text{street}, \text{house}\}$   
 $+ \langle \text{street}, \text{house}, \text{house}, \text{house} \rangle$   
 $\alpha_{13} \leftarrow \alpha_{13} - \langle [\text{front}, ?id, ?users, ?functions, (?capacity, ?w), (?articulation, ?w), (?qlevel, ?w_q), ?w, ?h, ?a],$   
 $\quad [\text{left}, ?id, ?users, ?functions, (?capacity, ?w), (?articulation, ?w), (?qlevel, ?w_q), ?w, ?h, ?a],$   
 $\quad [\text{back}, ?id, ?users, ?functions, (?capacity, ?w), (?articulation, ?w), (?qlevel, ?w_q), ?w, ?h, ?a],$   
 $\quad [\text{right}, ?id, ?users, ?functions, (?capacity, ?w), (?articulation, ?w), (?qlevel, ?w_q), ?w, ?h, ?a] \rangle$   
 $+ \langle [\text{front}, 1, \emptyset, (\text{street}), \emptyset, \emptyset, \emptyset, 0, 0, 0],$   
 $\quad [\text{left}, 2, \emptyset, (\text{house}), \emptyset, \emptyset, \emptyset, 0, 0, 0],$   
 $\quad [\text{back}, 3, \emptyset, (\text{house}), \emptyset, \emptyset, \emptyset, 0, 0, 0],$

[right, 4, ∅, (house), ∅, ∅, ∅, 0, 0, 0] >

**g3: houses on both sides and at the back (user)**

$\alpha_1 \leftarrow \alpha_1$  - < ?context<sub>i</sub>, ?context<sub>i</sub>, ?context<sub>b</sub>, ?context<sub>r</sub> > ,  
?context<sub>i</sub>, ?context<sub>i</sub>, ?context<sub>b</sub>, ?context<sub>r</sub> ∈ {street, house}  
+ < street, house, house, house >  
 $\alpha_{13} \leftarrow \alpha_{13}$  - < [front, ?id, ?users, ?functions, (?capacity, ?w<sub>i</sub>), (?articulation, ?w<sub>i</sub>), (?qlevel, ?w<sub>q</sub>), ?w, ?h, ?a],  
[left, ?id, ?users, ?functions, (?capacity, ?w<sub>i</sub>), (?articulation, ?w<sub>i</sub>), (?qlevel, ?w<sub>q</sub>), ?w, ?h, ?a],  
[back, ?id, ?users, ?functions, (?capacity, ?w<sub>i</sub>), (?articulation, ?w<sub>i</sub>), (?qlevel, ?w<sub>q</sub>), ?w, ?h, ?a],  
[right, ?id, ?users, ?functions, (?capacity, ?w<sub>i</sub>), (?articulation, ?w<sub>i</sub>), (?qlevel, ?w<sub>q</sub>), ?w, ?h, ?a]>  
+ < [front, 1, ∅, (street), ∅, ∅, ∅, 0, 0, 0],  
[left, 2, ∅, (house), ∅, ∅, ∅, 0, 0, 0],  
[back, 3, ∅, (house), ∅, ∅, ∅, 0, 0, 0],  
[right, 4, ∅, (house), ∅, ∅, ∅, 0, 0, 0] >

**g4: houses on one side and at the back (user)**

$\alpha_1 \leftarrow \alpha_1$  - < ?context<sub>i</sub>, ?context<sub>i</sub>, ?context<sub>b</sub>, ?context<sub>r</sub> > ,  
?context<sub>i</sub>, ?context<sub>i</sub>, ?context<sub>b</sub>, ?context<sub>r</sub> ∈ {street, house}  
+ < street, house, house, street >  
 $\alpha_{13} \leftarrow \alpha_{13}$  - < [front, ?id, ?users, ?functions, (?capacity, ?w<sub>i</sub>), (?articulation, ?w<sub>i</sub>), (?qlevel, ?w<sub>q</sub>), ?w, ?h, ?a],  
[left, ?id, ?users, ?functions, (?capacity, ?w<sub>i</sub>), (?articulation, ?w<sub>i</sub>), (?qlevel, ?w<sub>q</sub>), ?w, ?h, ?a],  
[back, ?id, ?users, ?functions, (?capacity, ?w<sub>i</sub>), (?articulation, ?w<sub>i</sub>), (?qlevel, ?w<sub>q</sub>), ?w, ?h, ?a],  
[right, ?id, ?users, ?functions, (?capacity, ?w<sub>i</sub>), (?articulation, ?w<sub>i</sub>), (?qlevel, ?w<sub>q</sub>), ?w, ?h, ?a]>  
+ < [front, 1, ∅, (street), ∅, ∅, ∅, 0, 0, 0],  
[left, 2, ∅, (house), ∅, ∅, ∅, 0, 0, 0],  
[back, 3, ∅, (house), ∅, ∅, ∅, 0, 0, 0],  
[right, 4, ∅, (street), ∅, ∅, ∅, 0, 0, 0] >

**g5: houses on one side (user)**

$\alpha_1 \leftarrow \alpha_1$  - < ?context<sub>i</sub>, ?context<sub>i</sub>, ?context<sub>b</sub>, ?context<sub>r</sub> > ,  
?context<sub>i</sub>, ?context<sub>i</sub>, ?context<sub>b</sub>, ?context<sub>r</sub> ∈ {street, house}  
+ < street, house, street, street >  
 $\alpha_{13} \leftarrow \alpha_{13}$  - < [front, ?id, ?users, ?functions, (?capacity, ?w<sub>i</sub>), (?articulation, ?w<sub>i</sub>), (?qlevel, ?w<sub>q</sub>), ?w, ?h, ?a],  
[left, ?id, ?users, ?functions, (?capacity, ?w<sub>i</sub>), (?articulation, ?w<sub>i</sub>), (?qlevel, ?w<sub>q</sub>), ?w, ?h, ?a],  
[back, ?id, ?users, ?functions, (?capacity, ?w<sub>i</sub>), (?articulation, ?w<sub>i</sub>), (?qlevel, ?w<sub>q</sub>), ?w, ?h, ?a],  
[right, ?id, ?users, ?functions, (?capacity, ?w<sub>i</sub>), (?articulation, ?w<sub>i</sub>), (?qlevel, ?w<sub>q</sub>), ?w, ?h, ?a]>  
+ < [front, 1, ∅, (street), ∅, ∅, ∅, 0, 0, 0],  
[left, 2, ∅, (house), ∅, ∅, ∅, 0, 0, 0],  
[back, 3, ∅, (street), ∅, ∅, ∅, 0, 0, 0],  
[right, 4, ∅, (street), ∅, ∅, ∅, 0, 0, 0] >

**g6: houses on both sides (user)**

$\alpha_1 \leftarrow \alpha_1$  - < ?context<sub>i</sub>, ?context<sub>i</sub>, ?context<sub>b</sub>, ?context<sub>r</sub> > ,  
?context<sub>i</sub>, ?context<sub>i</sub>, ?context<sub>b</sub>, ?context<sub>r</sub> ∈ {street, house}  
+ < street, house, street, house >  
 $\alpha_{13} \leftarrow \alpha_{13}$  - < [front, ?id, ?users, ?functions, (?capacity, ?w<sub>i</sub>), (?articulation, ?w<sub>i</sub>), (?qlevel, ?w<sub>q</sub>), ?w, ?h, ?a],  
[left, ?id, ?users, ?functions, (?capacity, ?w<sub>i</sub>), (?articulation, ?w<sub>i</sub>), (?qlevel, ?w<sub>q</sub>), ?w, ?h, ?a],  
[back, ?id, ?users, ?functions, (?capacity, ?w<sub>i</sub>), (?articulation, ?w<sub>i</sub>), (?qlevel, ?w<sub>q</sub>), ?w, ?h, ?a],  
[right, ?id, ?users, ?functions, (?capacity, ?w<sub>i</sub>), (?articulation, ?w<sub>i</sub>), (?qlevel, ?w<sub>q</sub>), ?w, ?h, ?a]>  
+ < [front, 1, ∅, (street), ∅, ∅, ∅, 0, 0, 0],  
[left, 2, ∅, (house), ∅, ∅, ∅, 0, 0, 0],  
[back, 3, ∅, (street), ∅, ∅, ∅, 0, 0, 0],  
[right, 4, ∅, (house), ∅, ∅, ∅, 0, 0, 0] >

**Solar orientation: specifying the solar orientation (user)**

**g7: default urban context (programmer)**

$\alpha_2 \leftarrow \alpha_2$  - < ?orientation<sub>r</sub>, ?orientation<sub>i</sub>, ?orientation<sub>b</sub>, ?orientation<sub>r</sub> >  
+ < South, East, North, West >  
 $\alpha_{13} \leftarrow \alpha_{13}$  - function (?orientation<sub>r</sub>, ?orientation<sub>i</sub>, ?orientation<sub>b</sub>, ?orientation<sub>r</sub>),

+ function (south, east, north, right, west)  
 ?orientation<sub>i</sub>, ?orientation<sub>i</sub>, ?orientation<sub>b</sub>, ?orientation<sub>r</sub>  
 ∈ {nil, south, west, north, east, southwest, southeast, northwest, northeast}

**g8: front elevation facing South (user)**

$\alpha_2 \leftarrow \alpha_2$  - < ?orientation<sub>i</sub>, ?orientation<sub>i</sub>, ?orientation<sub>b</sub>, ?orientation<sub>r</sub> >  
 + < South, East, North, West >  
 $\alpha_{13} \leftarrow \alpha_{13}$  - function (?orientation<sub>i</sub>, ?orientation<sub>i</sub>, ?orientation<sub>b</sub>, ?orientation<sub>r</sub>),  
 + function (south, east, north, right, west)  
 ?orientation<sub>i</sub>, ?orientation<sub>i</sub>, ?orientation<sub>b</sub>, ?orientation<sub>r</sub>  
 ∈ {nil, south, west, north, east, southwest, southeast, northwest, northeast}

**g9: front elevation facing West (user)**

$\alpha_2 \leftarrow \alpha_2$  - < ?orientation<sub>i</sub>, ?orientation<sub>i</sub>, ?orientation<sub>b</sub>, ?orientation<sub>r</sub> >  
 + < West, South, East, North >  
 $\alpha_{13} \leftarrow \alpha_{13}$  - function (?orientation<sub>i</sub>, ?orientation<sub>i</sub>, ?orientation<sub>b</sub>, ?orientation<sub>r</sub>),  
 + function (west, south, east, right, north)  
 ?orientation<sub>i</sub>, ?orientation<sub>i</sub>, ?orientation<sub>b</sub>, ?orientation<sub>r</sub>  
 ∈ {nil, south, west, north, east, southwest, southeast, northwest, northeast}

**g10: front elevation facing North (user)**

$\alpha_2 \leftarrow \alpha_2$  - < ?orientation<sub>i</sub>, ?orientation<sub>i</sub>, ?orientation<sub>b</sub>, ?orientation<sub>r</sub> >  
 + < North, West, South, East >  
 $\alpha_{13} \leftarrow \alpha_{13}$  - function (?orientation<sub>i</sub>, ?orientation<sub>i</sub>, ?orientation<sub>b</sub>, ?orientation<sub>r</sub>),  
 + function (north, west, south, east)  
 ?orientation<sub>i</sub>, ?orientation<sub>i</sub>, ?orientation<sub>b</sub>, ?orientation<sub>r</sub>  
 ∈ {nil, south, west, north, east, southwest, southeast, northwest, northeast}

**g11: front elevation facing East (user)**

$\alpha_2 \leftarrow \alpha_2$  - < ?orientation<sub>i</sub>, ?orientation<sub>i</sub>, ?orientation<sub>b</sub>, ?orientation<sub>r</sub> >  
 + < East, South, West, North >  
 $\alpha_{13} \leftarrow \alpha_{13}$  - function (?orientation<sub>i</sub>, ?orientation<sub>i</sub>, ?orientation<sub>b</sub>, ?orientation<sub>r</sub>),  
 + function (east, west, south, north)  
 ?orientation<sub>i</sub>, ?orientation<sub>i</sub>, ?orientation<sub>b</sub>, ?orientation<sub>r</sub>  
 ∈ {nil, south, west, north, east, southwest, southeast, northwest, northeast}

**g12: front elevation facing Southwest (user)**

$\alpha_2 \leftarrow \alpha_2$  - < ?orientation<sub>i</sub>, ?orientation<sub>i</sub>, ?orientation<sub>b</sub>, ?orientation<sub>r</sub> >  
 + < Southwest, Southeast, Northeast, Northwest >  
 $\alpha_{13} \leftarrow \alpha_{13}$  - function (?orientation<sub>i</sub>, ?orientation<sub>i</sub>, ?orientation<sub>b</sub>, ?orientation<sub>r</sub>),  
 + function (southwest, southeast, northeast, northwest)  
 ?orientation<sub>i</sub>, ?orientation<sub>i</sub>, ?orientation<sub>b</sub>, ?orientation<sub>r</sub>  
 ∈ {nil, south, west, north, east, southwest, southeast, northwest, northeast}

**g13: front elevation facing Northwest (user)**

$\alpha_2 \leftarrow \alpha_2$  - < ?orientation<sub>i</sub>, ?orientation<sub>i</sub>, ?orientation<sub>b</sub>, ?orientation<sub>r</sub> >  
 + < Northwest, Southwest, Southeast, Northeast >  
 $\alpha_{13} \leftarrow \alpha_{13}$  - function (?orientation<sub>i</sub>, ?orientation<sub>i</sub>, ?orientation<sub>b</sub>, ?orientation<sub>r</sub>),  
 + function (northwest, southwest, southeast, northeast)  
 ?orientation<sub>i</sub>, ?orientation<sub>i</sub>, ?orientation<sub>b</sub>, ?orientation<sub>r</sub>  
 ∈ {nil, south, west, north, east, southwest, southeast, northwest, northeast}

**g14: front elevation facing Northeast (user)**

$\alpha_2 \leftarrow \alpha_2$  - < ?orientation<sub>i</sub>, ?orientation<sub>i</sub>, ?orientation<sub>b</sub>, ?orientation<sub>r</sub> >  
 + < Northeast, Northwest, Southwest, Southeast >  
 $\alpha_{13} \leftarrow \alpha_{13}$  - function (?orientation<sub>i</sub>, ?orientation<sub>i</sub>, ?orientation<sub>b</sub>, ?orientation<sub>r</sub>),  
 + function (northeast, northwest, southwest, southeast)  
 ?orientation<sub>i</sub>, ?orientation<sub>i</sub>, ?orientation<sub>b</sub>, ?orientation<sub>r</sub>  
 ∈ {nil, south, west, north, east, southwest, southeast, northwest, northeast}

**g15: front elevation facing Southeast (user)**  
 $\alpha_2 \leftarrow \alpha_2 - < ?front, ?left, ?back, ?right >$   
 $+ < Southeast, Northeast, Northwest, Southwest >$   
 $\alpha_{13} \leftarrow \alpha_{13} - \text{function} (?orientation_r, ?orientation_l, ?orientation_b, ?orientation_r)$   
 $+ \text{function} (southeast, northeast, northwest, southwest)$   
 $?orientation_r, ?orientation_l, ?orientation_b, ?orientation_r$   
 $\in \{nil, south, west, north, east, southwest, southeast, northwest, northeast\}$

## TYPOLOGY

### Customization: Selecting the degree of customization

**g16: Deafault degree (programmer)**  
 $\alpha_3 \leftarrow \alpha_3 - < ?degree >, ?degree \in \{custom, type\}$   
 $+ < type >$

**g17: Customized house (user)**  
 $\alpha_3 \leftarrow \alpha_3 - < ?degree >, ?degree \in \{custom, type\}$   
 $+ < custom >$

**g18: Housetype (user)**  
 $\alpha_3 \leftarrow \alpha_3 - < ?degree > ?degree \in \{custom, type\}$   
 $+ < type >$

### Customized house

#### Dwellers:

#### g19: Specifying dwellers information (user), and adding required bedrooms (programmer)

$\alpha_3 \leftarrow \alpha_3, \alpha_3 = \text{custom}$   
 $\alpha_4 \leftarrow \alpha_4 + < 1, (name_{n+1}, age_{n+1}, gender_{n+1}) >,$   
 $age_i \in \{ '0-1', '2-5', '6-13', '14-17', '18-23', '23-65', '> 65' \},$   
 $gender_i \in \{ male, female \}$   
 $\alpha_5 \leftarrow \alpha_5 + < 1, [(couple, 0), (double, 0), (single, 1)] >$   
 $\alpha_6 \leftarrow \alpha_6, q_{level} = \alpha_6$   
 $\alpha_{13} \leftarrow \alpha_{13} + < be, id_{be}, (name_{n+1}), (sleeping), (single, 100), (q_{level}, 100), w_{si}, h_{si}, a_{si} >$   
 $id_{be} = \max(id) + 1$   
 $\sim \exists < ?id_{be}, ?id_{f1}, on, ?w > \in \alpha_{17}$   
 $\Rightarrow \alpha_{14} \leftarrow \alpha_{14}, + \text{sleeping} (id_{be}, a_{si})$   
 $\alpha_{15} \leftarrow \alpha_{15}, + < \text{available}, (0, -a_{si}, 0, -a_{si}), (0, 0, 0, 0), (0, -a_{si}, 0, -a_{si}), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16}, + < \text{used}, (a_{si}, a_{si}, 0, a_{si}), (0, 0, 0, 0), (a_{si}, a_{si}, 0, a_{si}), -a_i / a_u + (a_i +$   
 $a_{si} / a_u + a_{si}) >$   
 $\alpha_{17} \leftarrow \alpha_{17}, + < [id_{be}, id_{f1}, on, 100] >$   
 $\exists < ?id_{be}, ?id_{f1}, on, ?w >$   
 $\Rightarrow \alpha_{14} \leftarrow \alpha_{14}, + f_2 (id_{be}, a_{si})$   
 $\alpha_{15} \leftarrow \alpha_{15}, + < \text{available}, (0, 0, 0, 0), (0, -a_{si}, 0, -a_{si}), (0, -a_{si}, 0, -a_{si}), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16}, + < \text{used}, (0, 0, 0, 0), (a_{si}, a_{si}, 0, a_{si}), (a_{si}, a_{si}, 0, a_{si}), -a_i / a_u + (a_i +$   
 $a_{si} / a_u + a_{si}) >$   
 $\alpha_{17} \leftarrow \alpha_{17}, + < [id_{be}, id_{f2}, on, 100] >$   
 $\alpha_{24} \leftarrow \alpha_{24} + a_{si} \cdot cm_2$   
 $\beta_1 \leftarrow \beta_1, w_{si} = w (q_{level}, si)$   
 $\beta_2 \leftarrow \beta_2, h_{si} = h (q_{level}, si)$   
 $\beta_3 \leftarrow \beta_3, a_{si} = a (q_{level}, si)$   
 $\beta_6 \leftarrow \beta_6, cm_2 = \text{cost\_m2} (q_{level}, \text{covered})$

#### Share:

#### g20: Specifying that two dwellers share a bedroom (user), and update description (programmer.)

$\alpha_3 \leftarrow \alpha_3, \alpha_3 = \text{custom}$   
 $\alpha_5 \leftarrow \alpha_5 + < -1, [(couple, 0), (double, 1), (single, -2)] >$   
 $\alpha_6 \leftarrow \alpha_6, q_{level} = \alpha_6$

$\alpha_{13} \leftarrow \alpha_{13}$   
- < [be, ?id<sub>1</sub>, (?name<sub>1</sub>), (sleeping), (single, ?w<sub>c1</sub>), (isolated, 100), (?q<sub>level1</sub>, ?w<sub>q1</sub>), ?w<sub>1</sub>, ?h<sub>1</sub>, ?a<sub>1</sub>],  
[be, ?id<sub>2</sub>, (?name<sub>2</sub>), (sleeping), (single, ?w<sub>c2</sub>), (isolated, 100), (?q<sub>level2</sub>, ?w<sub>q2</sub>), ?w<sub>2</sub>, ?h<sub>2</sub>, ?a<sub>2</sub>]  
+ < [be, ?id<sub>1</sub>, (?name<sub>i</sub>, ?name<sub>j</sub>), (double, 100), (?articulation, 100), (?q<sub>leveldo</sub>, ?w<sub>qdo</sub>), w<sub>do</sub>, h<sub>do</sub>, a<sub>do</sub>]  
?q<sub>leveldo</sub> = max (?q<sub>level1</sub>, ?q<sub>level2</sub>), ?w<sub>qdo</sub> = max (?w<sub>q1</sub>, ?w<sub>q2</sub>)  
 $\neg \exists < ?id_n, ?id_{f1}, on, ?w > \in \alpha_{17}$   
 $\Rightarrow \alpha_{14} \leftarrow \alpha_{14}, - \text{sleeping}(id_n, a_{si})$   
+ sleeping(id<sub>1</sub>, a<sub>do</sub>)  
 $\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, 2 \cdot a_{si} - a_{do}, 0, 2 \cdot a_{si} - a_{do}), (0, 0, 0, 0), (0, 2 \cdot a_{si} - a_{do}, 0, 2 \cdot a_{si} - a_{do}), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (a_{do} - 2 \cdot a_{si}, a_{do} - 2 \cdot a_{si}, 0, a_{do} - 2 \cdot a_{si}),$   
(0, 0, 0, 0),  
(a<sub>do</sub> - 2 · a<sub>si</sub>, a<sub>do</sub> - 2 · a<sub>si</sub>, 0, a<sub>do</sub> - 2 · a<sub>si</sub>),  
- a<sub>i</sub> / a<sub>u</sub> + (a<sub>i</sub> + a<sub>do</sub> - 2 · a<sub>si</sub> / a<sub>u</sub> + a<sub>do</sub> - 2 · a<sub>si</sub>) >  
 $\alpha_{17} \leftarrow \alpha_{17} + < [?id_{be}, id_{f1}, on, 100] >$   
 $\exists < ?id_n, ?id_{f1}, on, ?w > \in \alpha_{17}$   
 $\Rightarrow \alpha_{14} \leftarrow \alpha_{14} - \text{sleeping}(id_n, a_{si})$   
+ f2(id<sub>1</sub>, a<sub>do</sub>)  
 $\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, 0, 0, 0), (0, 2 \cdot a_{si} - a_{do}, 0, 2 \cdot a_{si} - a_{do}), (0, 2 \cdot a_{si} - a_{do}, 0, 2 \cdot a_{si} - a_{do}), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (0, 0, 0, 0),$   
(a<sub>do</sub> - 2 · a<sub>si</sub>, a<sub>do</sub> - 2 · a<sub>si</sub>, 0, a<sub>do</sub> - 2 · a<sub>si</sub>),  
(a<sub>do</sub> - 2 · a<sub>si</sub>, a<sub>do</sub> - 2 · a<sub>si</sub>, 0, a<sub>do</sub> - 2 · a<sub>si</sub>),  
- a<sub>i</sub> / a<sub>u</sub> + (a<sub>i</sub> + a<sub>do</sub> - 2 · a<sub>si</sub> / a<sub>u</sub> + a<sub>do</sub> - 2 · a<sub>si</sub>) >  
 $\alpha_{17} \leftarrow \alpha_{17} + < [?id_{be}, id_2, on, 100] >$   
 $\alpha_{24} \leftarrow \alpha_{24} + a_{do} \cdot cm2 - 2 \cdot a_{si} \cdot cm2$   
 $\beta_1 \leftarrow \beta_1, w_{do} = w(q_{level}, do)$   
 $\beta_2 \leftarrow \beta_2, h_{do} = h(q_{level}, do)$   
 $\beta_3 \leftarrow \beta_3, a_{do} = a(q_{level}, do)$   
 $\beta_6 \leftarrow \beta_6, cm2 = cost\_m2(q_{level}, covered)$

**g21: Specifying that two dwellers share a bed (user), and update description (programmer)**

$\alpha_3 \leftarrow \alpha_3, \alpha_3 = \text{custom}$   
 $\alpha_5 \leftarrow \alpha_5 + < 1, [(couple, 1), (double, 0), (single, -2)] >$   
 $\alpha_6 \leftarrow \alpha_6, q_{level} = \alpha_6$   
 $\alpha_{13} \leftarrow \alpha_{13}$   
- < [be, ?id<sub>1</sub>, (?name<sub>1</sub>), (sleeping), (single, ?w<sub>c1</sub>), (isolated, 100), (?q<sub>level1</sub>, ?w<sub>q1</sub>), ?w<sub>1</sub>, ?h<sub>1</sub>, ?a<sub>1</sub>],  
[be, ?id<sub>2</sub>, (?name<sub>2</sub>), (sleeping), (single, ?w<sub>c2</sub>), (isolated, 100), (?q<sub>level2</sub>, ?w<sub>q2</sub>), ?w<sub>2</sub>, ?h<sub>2</sub>, ?a<sub>2</sub>]  
+ < [be, ?id<sub>1</sub>, (?name<sub>i</sub>, ?name<sub>j</sub>), (sleeping), (double, 100), (?articulation, 100), (?q<sub>leveldo</sub>, ?w<sub>qdo</sub>),  
w<sub>cu</sub>, h<sub>cu</sub>, a<sub>cu</sub>] >  
?q<sub>leveldo</sub> = max (?q<sub>level1</sub>, ?q<sub>level2</sub>), ?w<sub>qdo</sub> = max (?w<sub>q1</sub>, ?w<sub>q2</sub>)  
 $\neg \exists < ?id_n, ?id_{f1}, on, ?w > \in \alpha_{17}, n \in \{1, 2\}$   
 $\Rightarrow \alpha_{14} \leftarrow \alpha_{14} - \text{sleeping}(id_n, a_{si})$   
+ sleeping(id<sub>1</sub>, a<sub>cu</sub>)  
 $\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, 2 \cdot a_{si} - a_{cu}, 0, 2 \cdot a_{si} - a_{cu}), (0, 0, 0, 0), (0, 2 \cdot a_{si} - a_{cu}, 0, 2 \cdot a_{si} - a_{cu}), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (a_{cu} - 2 \cdot a_{si}, a_{cu} - 2 \cdot a_{si}, 0, a_{cu} - 2 \cdot a_{si}),$   
(0, 0, 0, 0),  
(a<sub>cu</sub> - 2 · a<sub>si</sub>, a<sub>cu</sub> - 2 · a<sub>si</sub>, 0, a<sub>cu</sub> - 2 · a<sub>si</sub>),  
- a<sub>i</sub> / a<sub>u</sub> + (a<sub>i</sub> + a<sub>cu</sub> - 2 · a<sub>si</sub> / a<sub>u</sub> + a<sub>cu</sub> - 2 · a<sub>si</sub>) >  
 $\alpha_{17} \leftarrow \alpha_{17} + < [?id_{be}, id_{f1}, on, 100] >$   
 $\exists < ?id_n, ?id_{f1}, on, ?w > \in \alpha_{17}, n \in \{1, 2\}$   
 $\Rightarrow \alpha_{14} \leftarrow \alpha_{14} - \text{sleeping}(id_n, a_{si})$   
+ sleeping(id<sub>1</sub>, a<sub>cu</sub>)  
 $\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, 0, 0, 0), (0, 2 \cdot a_{si} - a_{cu}, 0, 2 \cdot a_{si} - a_{cu}), (0, 2 \cdot a_{si} - a_{cu}, 0, 2 \cdot a_{si} - a_{cu}), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (0, 0, 0, 0),$   
(a<sub>cu</sub> - 2 · a<sub>si</sub>, a<sub>cu</sub> - 2 · a<sub>si</sub>, 0, a<sub>cu</sub> - 2 · a<sub>si</sub>),  
(a<sub>cu</sub> - 2 · a<sub>si</sub>, a<sub>cu</sub> - 2 · a<sub>si</sub>, 0, a<sub>cu</sub> - 2 · a<sub>si</sub>),  
- a<sub>i</sub> / a<sub>u</sub> + (a<sub>i</sub> + a<sub>cu</sub> - 2 · a<sub>si</sub> / a<sub>u</sub> + a<sub>cu</sub> - 2 · a<sub>si</sub>) >  
 $\alpha_{17} \leftarrow \alpha_{17} + < [?id_{be}, id_2, on, 100] >$   
 $\alpha_{24} \leftarrow \alpha_{24} + a_{cu} \cdot cm2 - 2 \cdot a_{si} \cdot cm2$   
 $\beta_1 \leftarrow \beta_1, w_{cu} = w(q_{level}, cu)$



$\beta_2 \leftarrow \beta_2, h_{cu} = h(q_{level}, cu)$   
 $\beta_3 \leftarrow \beta_3, a_{cu} = a(q_{level}, cu)$   
 $\beta_6 \leftarrow \beta_6, cm2 = cost\_m2(q_{level}, covered)$

## Housetype

### Bedrooms: setting the range for the possible number of dwellers (programmer)

#### g22: The number of bedrooms is 0 (user)

$\alpha_3 \leftarrow \alpha_3, \quad \alpha_3 = type$   
 $\alpha_4 \leftarrow \alpha_4 + \langle \Pi_{dwellers}, \emptyset \rangle, \quad \Pi_{users} \in \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$   
 $\alpha_5 \leftarrow \alpha_5 + \langle 0, \emptyset \rangle$

#### g23: The number of bedrooms is 1 (user)

$\alpha_3 \leftarrow \alpha_3, \quad \alpha_3 = type$   
 $\alpha_4 \leftarrow \alpha_4 + \langle \Pi_{dwellers}, \emptyset \rangle, \quad \Pi_{users} \in \{1, 2\}$   
 $\alpha_5 \leftarrow \alpha_5 + \langle 1, \emptyset \rangle$

#### g24: The number of bedrooms is 2 (user)

$\alpha_3 \leftarrow \alpha_3, \quad \alpha_3 = type$   
 $\alpha_4 \leftarrow \alpha_4 + \langle \Pi_{dwellers}, \emptyset \rangle, \quad \Pi_{users} \in \{2, 3, 4\}$   
 $\alpha_5 \leftarrow \alpha_5 + \langle 2, \emptyset \rangle$

#### g25: The number of bedrooms is 3 (user)

$\alpha_3 \leftarrow \alpha_3, \quad \alpha_3 = type$   
 $\alpha_4 \leftarrow \alpha_4 + \langle \Pi_{dwellers}, \emptyset \rangle, \quad \Pi_{users} \in \{4, 5, 6\}$   
 $\alpha_5 \leftarrow \alpha_5 + \langle 3, \emptyset \rangle$

#### g26: The number of bedrooms is 4 (user)

$\alpha_3 \leftarrow \alpha_3, \quad \alpha_3 = type$   
 $\alpha_4 \leftarrow \alpha_4 + \langle \Pi_{dwellers}, \emptyset \rangle, \quad \Pi_{users} \in \{5, 6, 7\}$   
 $\alpha_5 \leftarrow \alpha_5 + \langle 4, \emptyset \rangle$

#### g27: The number of bedrooms is 5 (user)

$\alpha_3 \leftarrow \alpha_3, \quad \alpha_3 = type$   
 $\alpha_4 \leftarrow \alpha_4 + \langle \Pi_{dwellers}, \emptyset \rangle, \quad \Pi_{users} \in \{7, 8, 9\}$   
 $\alpha_5 \leftarrow \alpha_5 + \langle 5, \emptyset \rangle$

### Dwellers: setting the range for the possible number of bedrooms (programmer)

#### g28: The number of dwellers is 0 (user)

$\alpha_3 \leftarrow \alpha_3, \quad \alpha_3 = type$   
 $\alpha_4 \leftarrow \alpha_4 + \langle 0, \emptyset \rangle$   
 $\alpha_5 \leftarrow \alpha_5 + \langle \Pi_{bedrooms}, \emptyset \rangle, \quad \Pi_{bedrooms} \in \{1, 2, 3, 4, 5\}$

#### g29: The number of dwellers is 1 (user)

$\alpha_3 \leftarrow \alpha_3, \quad \alpha_3 = type$   
 $\alpha_4 \leftarrow \alpha_4 + \langle 1, \emptyset \rangle$   
 $\alpha_5 \leftarrow \alpha_5 + \langle 1, \emptyset \rangle$

#### g30: The number of dwellers is 2 (user)

$\alpha_3 \leftarrow \alpha_3, \quad \alpha_3 = type$   
 $\alpha_4 \leftarrow \alpha_4 + \langle 2, \emptyset \rangle$   
 $\alpha_5 \leftarrow \alpha_5 + \langle \Pi_{bedrooms}, \emptyset \rangle, \quad \Pi_{bedrooms} \in \{1, 2\}$

#### g31: The number of dwellers is 3 (user)

$\alpha_3 \leftarrow \alpha_3, \quad \alpha_3 = type$   
 $\alpha_4 \leftarrow \alpha_4 + \langle 3, \emptyset \rangle$   
 $\alpha_5 \leftarrow \alpha_5 + \langle \Pi_{bedrooms}, \emptyset \rangle, \quad \Pi_{bedrooms} \in \{2, 3\}$

#### g32: The number of dwellers is 4 (user)

$\alpha_3 \leftarrow \alpha_3, \quad \alpha_3 = \text{type}$   
 $\alpha_4 \leftarrow \alpha_4 + < 4, \emptyset >$   
 $\alpha_5 \leftarrow \alpha_5 < \text{Nbedrooms}, \emptyset >, \quad \text{Nbedrooms} \in \{2, 3\}$

**g33: The number of dwellers is 5 (user)**

$\alpha_3 \leftarrow \alpha_3, \quad \alpha_3 = \text{type}$   
 $\alpha_4 \leftarrow \alpha_4 + < 5, \emptyset >$   
 $\alpha_5 \leftarrow \alpha_5 < \text{Nbedrooms}, \emptyset >, \quad \text{Nbedrooms} \in \{3, 4\}$

**g34: The number of dwellers is 6 (user)**

$\alpha_3 \leftarrow \alpha_3, \quad \alpha_3 = \text{type}$   
 $\alpha_4 \leftarrow \alpha_4 + < 6, \emptyset >$   
 $\alpha_5 \leftarrow \alpha_5 < \text{Nbedrooms}, \emptyset >, \quad \text{Nbedrooms} \in \{3, 4\}$

**g35: The number of dwellers is 7 (user)**

$\alpha_3 \leftarrow \alpha_3, \quad \alpha_3 = \text{type}$   
 $\alpha_4 \leftarrow \alpha_4 + < 7, \emptyset >$   
 $\alpha_5 \leftarrow \alpha_5 < \text{Nbedrooms}, \emptyset >, \quad \text{Nbedrooms} \in \{4, 5\}$

**g36: The number of dwellers is 8 (user)**

$\alpha_3 \leftarrow \alpha_3, \quad \alpha_3 = \text{type}$   
 $\alpha_4 \leftarrow \alpha_4 + < 8, \emptyset >$   
 $\alpha_5 \leftarrow \alpha_5 < 5, \emptyset >$

**g37: The number of dwellers is 9 (user)**

$\alpha_3 \leftarrow \alpha_3, \quad \alpha_3 = \text{type}$   
 $\alpha_4 \leftarrow \alpha_4 + < 9, \emptyset >$   
 $\alpha_5 \leftarrow \alpha_5 < 5, \emptyset >$

**MORPHOLOGY**

**Yard:**

**g38: Specifying the yard location (programmer)**

$\alpha_7 \leftarrow \alpha_7 - < ?\text{yard} >, ?\text{yard} \in \{\emptyset, \text{default}\}$   
 $+ < \text{front} >$

**g39: Specifying the yard location (user)**

$\alpha_7 \leftarrow \alpha_7 - < ?\text{yard} >$   
 $+ < \text{yard} >, ?\text{yard}, \text{yard} \in \{\text{front}, \text{back}, \text{default}\}$

**Floors: specifying the number of floors (user/programmer)**

**g40: Setting the number of floors to 1, when the number of bedrooms is 1**

$\alpha_5 \leftarrow \alpha_5, \quad \text{Nbedrooms} = 1$   
 $\alpha_8 \leftarrow \alpha_8 - < ?\text{Nfloors} >$   
 $+ < 1 >$

**g41: Setting the possible number of floors to 1, when the number of bedrooms is 2**

$\alpha_5 \leftarrow \alpha_5, \quad \text{Nbedrooms} = 2$   
 $\alpha_8 \leftarrow \alpha_8 - < ?\text{Nfloors} >$   
 $+ < n >, \quad n \in \{1, 2\}$

**g42: Setting the number of floors to 2, when the number of bedrooms is 2**

$\alpha_5 \leftarrow \alpha_5, \quad \text{Nbedrooms} > 2$   
 $\alpha_8 \leftarrow \alpha_8 - < ?\text{Nfloors} >$   
 $+ < 2 >$

**Balconies:**

**g43: Indicating whether balconies are desired (programmer)**

$\alpha_9 \leftarrow \alpha_9 - < ?\text{balconies} >, ?\text{balconies} \in \{\emptyset, \text{default}\}$

+ < true >

**g44: Indicating whether balconies are desired (user)**

$\alpha_9 \leftarrow \alpha_9$  - < ?balconies >  
 + < balconies >, ?balconies, balconies  $\in$  {true, false}

**QUALITY**

**g45: setting the default housing quality to minimum (programmer)**

$\alpha_6 \leftarrow \alpha_6$  + < minimum >  
 $\alpha_{23} \leftarrow \alpha_{23}$   
 + < [minimum, 50], [minimum, 50], [minimum, 34], [minimum, 33], [minimum, 33],  
 [minimum, 50], [minimum, 50] >

**g46: specifying the housing quality (user)**

$\alpha_6 \leftarrow \alpha_6$  - < ?quality >  
 + < quality >, ?quality, quality  $\in$  {minimum, medium, maximum}

**g47: setting the qualities weights**

$\alpha_{23} \leftarrow \alpha_{23}$  + set\_qLevel\_weight ( $v_i$ ,  $w_i$ ),  
 $v_i \in$  {function, spatiality, capacity, articulation, spaciousness, topology, aesthetics},  
 $w_i \in$  {5, 10, 15, ..., 100}

**g48: calculate the current quality level**

$\alpha_6 \leftarrow \alpha_6$  + < qlevel >  
 $\alpha_{13} \leftarrow \alpha_{13}$   
 $\alpha_{23} \leftarrow \alpha_{23}$  - < [Vfunction, Wfunction], [Vspatiality, Wspatiality], [Vcapacity, Wcapacity],  
 [Varticulation, Warticulation], [Vspaciousness, Wspaciousness],  
 [Vtopology, Wtopology], [Vaesthetics, Waesthetics] >  
 + < [Vfunction, Wfunction], [Vspatiality, Wspatiality], [Vcapacity, Wcapacity],  
 [Varticulation, Warticulation], [Vspaciousness, Wspaciousness],  
 [Vtopology, Wtopology], [Vaesthetics, Waesthetics] >

$$(1) V_{\text{housing}} = \frac{V_{\text{function}} \cdot W_{\text{function}} + V_{\text{aesthetics}} \cdot W_{\text{aesthetics}}}{W_{\text{function}} + W_{\text{aesthetics}}}$$

$$(2) V_{\text{function}} = \frac{V_{\text{spatiality}} \cdot W_{\text{spatiality}} + V_{\text{topology}} \cdot W_{\text{topology}}}{W_{\text{spatiality}} + W_{\text{topology}}}$$

$$(3) V_{\text{spatiality}} = \frac{V_{\text{dwelling capacity}} \cdot W_{\text{dwelling capacity}} + V_{\text{capacity}} \cdot W_{\text{capacity}} + V_{\text{articulation}} \cdot W_{\text{articulation}} + V_{\text{spaciousness}} \cdot W_{\text{spaciousness}}}{W_{\text{dwelling capacity}} + W_{\text{capacity}} + W_{\text{articulation}} + W_{\text{spaciousness}}}$$

$$(4) V_{\text{dwelling capacity}} = \frac{\sum V_{\text{dwelling capacity}}(\text{space}_i) \cdot W_{\text{dwelling capacity}}(\text{space}_i)}{\sum W_{\text{dwelling capacity}}(\text{space}_i)}$$

$$(5) V_{\text{capacity}} = \frac{\sum V_{\text{capacity}}(\text{space}_i) \cdot W_{\text{capacity}}(\text{space}_i)}{\sum W_{\text{capacity}}(\text{space}_i)}$$

$$(6) V_{\text{articulation}} = \frac{\sum V_{\text{articulation}}(\text{space}_i) \cdot W_{\text{articulation}}(\text{space}_i)}{\sum W_{\text{articulation}}(\text{space}_i)}$$

$$(7) V_{\text{spaciousness}} = \frac{\sum V_{\text{spaciousness}}(\text{space}_i) \cdot W_{\text{spaciousness}}(\text{space}_i)}{\sum W_{\text{spaciousness}}(\text{space}_i)}$$

$$(8) V_{\text{topology}} = \frac{\sum V_{\text{topology}}(\text{relation}_i) \cdot W_{\text{topology}}(\text{relation}_i)}{\sum W_{\text{topology}}(\text{relation}_i)}$$

$$(9) V_{\text{aesthetics}} = \frac{\sum V_{\text{aesthetics}}(\text{space}_i) \cdot W_{\text{aesthetics}}(\text{space}_i)}{\sum W_{\text{aesthetics}}(\text{space}_i)}$$

$$V_q \in [0, 1] \Rightarrow q_{\text{level}} = \text{minimum}$$

$$V_q \in ]1, 2] \Rightarrow q_{\text{level}} = \text{medium}$$

$$V_q \in ]2, 3] \Rightarrow q_{\text{level}} = \text{maximum}$$

$$V_q \in \{\text{function, aesthetics, spatiality, topology, dwelling capacity, capacity, articulation, spaciousness}\}$$

### Getting the dwelling capacity average quality level

$V_{\text{dwelling capacity}}(\text{space}_i)$ :

$\text{space}_i \in \text{minimum\_list} \Rightarrow V_{\text{dwelling capacity}}(\text{space}_i) = 1$

$\text{space}_i \in \text{medium\_list} - \text{minimum\_list} \Rightarrow V_{\text{dwelling capacity}}(\text{space}_i) = 2$

$\text{space}_i \in \text{maximum\_list} - (\text{minimum\_list} + \text{medium\_list}) \Rightarrow V_{\text{dwelling capacity}}(\text{space}_i) = 3$

### Getting the capacity average quality level

$V_{\text{capacity}}(\text{space}_i)$ :

$\text{space}_i = \text{bedroom}$ :  $\text{capacity}(\text{space}_i) = 2 \wedge \text{share}(\text{users}(\text{space}_i)) = \text{room} \Rightarrow V_{\text{capacity}}(\text{space}_i) = 1$

$\text{capacity}(\text{space}_i) = 2 \wedge \text{share}(\text{users}(\text{space}_i)) = \text{bed} \Rightarrow V_{\text{capacity}}(\text{space}_i) = 2$

$\text{capacity}(\text{space}_i) = 1 \Rightarrow V_{\text{capacity}}(\text{space}_i) = 3$

$\text{space}_i \in \{\text{kitchen, livingroom, laundry, pantry, global storage, patio, terrace}\}$

$\text{capacity}(\text{space}_i) = n_{\text{users}} \Rightarrow V_{\text{capacity}}(\text{space}_i) = 1$

$\text{capacity}(\text{space}_i) = n_{\text{users}} + 1 \Rightarrow V_{\text{capacity}}(\text{space}_i) = 2$

$\text{capacity}(\text{space}_i) \geq n_{\text{users}} + 2 \Rightarrow V_{\text{capacity}}(\text{space}_i) = 3$

$\text{space}_i = \text{bathroom}$   $\text{capacity}(\text{space}_i) = \text{lavatory} \Rightarrow V_{\text{capacity}}(\text{space}_i) = 1$

$\text{capacity}(\text{space}_i) = \text{shower} \Rightarrow V_{\text{capacity}}(\text{space}_i) = 2$

$\text{capacity}(\text{space}_i) = \text{bathtub} \Rightarrow V_{\text{capacity}}(\text{space}_i) = 3$

$\text{space}_i \in \{\text{formal diningroom, informal diningroom}\}$

$\text{capacity}(\text{space}_{i1}) + \text{capacity}(\text{space}_{i2}) = n_{\text{users}} \Rightarrow V_{\text{capacity}}(\text{space}_i) = 1$

$\text{capacity}(\text{space}_i) + \text{capacity}(\text{space}_i) = n_{\text{users}} + 1 \Rightarrow V_{\text{capacity}}(\text{space}_i) = 2$

$\text{capacity}(\text{space}_i) + \text{capacity}(\text{space}_i) \geq n_{\text{users}} + 2 \Rightarrow V_{\text{capacity}}(\text{space}_i) = 3$

$\text{space}_i \in \{\text{playspace, studyspace, workspace}\}$

$\text{capacity}(\text{space}_i) = 2 \Rightarrow V_{\text{capacity}}(\text{space}_i) = 1$

$\text{capacity}(\text{space}_i) = 1 \Rightarrow V_{\text{capacity}}(\text{space}_i) = 2$

$\text{capacity}(\text{space}_i) = 1 \wedge$

$\text{capacity}_{\text{current}}(\text{function}(\text{space}_i)) > \text{capacity}_{\text{current}}(\text{function}(\text{space}_i)) \Rightarrow V_{\text{capacity}}(\text{space}_i) = 3$

$\text{space}_i \in \{\text{balcony, terrace, patio}\}$

$\text{space}_i \in \{\text{hall, corridor, staircase}\}$

### Getting the articulation average quality level

$V_{\text{articulation}}(\text{space}_i)$ :

$\text{Articulation} = \text{included} \Rightarrow V_{\text{articulation}}(\text{space}_i) = 1$

$\text{Articulation} = \text{delimited} \Rightarrow V_{\text{articulation}}(\text{space}_i) = 2$

$\text{Articulation} = \text{isolated} \Rightarrow V_{\text{articulation}}(\text{space}_i) = 3$

### Getting the spaciousness average quality level

$V_{\text{spaciousness}} = (\text{quality}(\text{area}(\text{space}_i)) + \text{quality}(\text{width}(\text{space}_i)) + \text{quality}(\text{height}(\text{space}_i))) / 3$

### Getting the topology average quality level

$V_{\text{topology}}(\text{space}_i, \text{space}_j)$ :

$r_{\text{req}}(\text{space}_i, \text{space}_j) = \text{door}$

$\wedge r(\text{space}_i, \text{space}_j) \neq \text{close, adjacent, window, door} \Rightarrow q_{\text{level}} = 0$

$\wedge r(\text{space}_i, \text{space}_j) = \text{close} \Rightarrow q_{\text{level}} = 1$

$\wedge r \in \{\text{adjacent, window}\} \Rightarrow q_{\text{level}} = 2$

$\wedge r = \text{door} \Rightarrow q_{\text{level}} = 3$

$r_{\text{req}}(\text{space}_i, \text{space}_j) = \text{merged}$

$\wedge r(\text{space}_i, \text{space}_j) \neq \text{close, adjacent, window, door, merged} \Rightarrow q_{\text{level}} = 0$

$\wedge r(\text{space}_i, \text{space}_j) \in \{\text{adjacent, close}\} \Rightarrow q_{\text{level}} = 1$

$\wedge r(\text{space}_i, \text{space}_j) \in \{\text{door, window}\} \Rightarrow q_{\text{level}} = 2$

$\wedge r(\text{space}_i, \text{space}_j) = \text{merged} \Rightarrow q_{\text{level}} = 3$

$r_{req}(\text{space}_i, \text{space}_j) = \text{any}$   
 $\forall r(\text{space}_i, \text{space}_j) \Rightarrow q_{\text{level}} = 3$

### Getting the aesthetic average quality level

$\forall \text{proportion, space} \Rightarrow q_{\text{level}} = 3$

## SPACIALITY

### Area: setting the available area (programmer)

#### g49: the housetype is frontyard, and there are balconies on the second floor

$\alpha_0 \leftarrow \alpha_0, Ag = \text{lot\_area}(\alpha_0) = 96.00 \text{ m}^2$   
 $\alpha_7 \leftarrow \alpha_7, \alpha_7 = \text{frontyard}$   
 $\alpha_9 \leftarrow \alpha_9, \alpha_9 = \text{true}$   
 $\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (72.50, 23.50, Ag), (68.50, 4.00, 72.50), (0, 141.00, 27.50, 168.50), 0.77 >$

#### g50: the housetype is frontyard, and there are no balconies on the second floor

$\alpha_0 \leftarrow \alpha_0, Ag = \text{lot\_area}(\alpha_0) = 96.00 \text{ m}^2$   
 $\alpha_7 \leftarrow \alpha_7, \alpha_7 = \text{frontyard}$   
 $\alpha_9 \leftarrow \alpha_9, \alpha_9 = \text{false}$   
 $\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (68.50, 27.50, Ag), (68.50, 0.00, 72.50), (0, 141.00, 27.50, 164.50), 0.77 >$

#### g51: the housetype is backyard, and there is a house at the back

$\alpha_0 \leftarrow \alpha_0, Ag = \text{lot\_area}(\alpha_0) = 96.00 \text{ m}^2$   
 $\alpha_1 \leftarrow \alpha_1, \alpha_1 = < \text{street}, ?\text{context}_l, \text{house}, ?\text{context}_r >, ?\text{context}_l, ?\text{context}_r \in \{\text{house}, \text{street}\}$   
 $\alpha_7 \leftarrow \alpha_7, \alpha_7 = \text{backyard}$   
 $\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (72.97, 23.03, Ag), (56.80, 16.17, 72.97), (0, 129.77, 39.20, 168.97), 0.77 >$

#### g52: the housetype is backyard, there is a street at the back, and there are balconies on the second floor

$\alpha_0 \leftarrow \alpha_0, Ag = \text{lot\_area}(\alpha_0) = 96.00 \text{ m}^2$   
 $\alpha_1 \leftarrow \alpha_1, \alpha_1 = < \text{street}, ?\text{context}_l, \text{street}, ?\text{context}_r >, ?\text{context}_l, ?\text{context}_r \in \{\text{house}, \text{street}\}$   
 $\alpha_7 \leftarrow \alpha_7, \alpha_7 = \text{backyard}$   
 $\alpha_9 \leftarrow \alpha_9, \alpha_9 = \text{true}$   
 $\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (72.50, 23.50, Ag), (68.50, 4.00, 72.50), (0, 141.00, 27.50, 168.50), 0.77 >$

#### g53: the housetype is backyard, there is a street at the back, and there are no balconies on the second floor

$\alpha_0 \leftarrow \alpha_0, Ag = \text{lot\_area}(\alpha_0) = 96.00 \text{ m}^2$   
 $\alpha_1 \leftarrow \alpha_1, \alpha_1 = < \text{street}, ?\text{context}_l, \text{street}, ?\text{context}_r >, ?\text{context}_l, \text{house}, ?\text{context}_r \in \{\text{house}, \text{street}\}$   
 $\alpha_7 \leftarrow \alpha_7, \alpha_7 = \text{backyard}$   
 $\alpha_9 \leftarrow \alpha_9, \alpha_9 = \text{false}$   
 $\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (68.50, 27.50, Ag), (68.50, 0.00, 72.50), (0, 137.00, 27.50, 164.50), 0.77 >$

### Obligatory Dwelling capacity (programmer)

#### g54: Adding a patio (programmer)

$\alpha_4 \leftarrow \alpha_4, n_{\text{dwellers}} = \alpha_4$   
 $\alpha_6 \leftarrow \alpha_6, q_{\text{level}} = \alpha_6$   
 $\alpha_{10} \leftarrow \alpha_{10} + < [y, 1] >$   
 $\alpha_{11} \leftarrow \alpha_{11} + < [y, \text{isolated}, 100] >$   
 $\alpha_{13} \leftarrow \alpha_{13} + < [y, \text{id}_y, \emptyset, (\text{being outside}), (n_{\text{dwellers}}, 100), (\text{isolated}, 100), (q_{\text{level}}, 0), w_y, h_y, a_y] >$   
 $\text{id}_y = \max(\text{id}) + 1$   
 $\alpha_{14} \leftarrow \alpha_{14} + \text{patio}(\text{id}_y, a_y)$   
 $\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, 0, -a_y, -a_y), (0, 0, 0, 0), (0, 0, -a_y, -a_y), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (0, 0, a_y, a_y), (0, 0, 0, 0), (0, 0, a_y, a_y), -a_i / a_u + (a_i / a_u + a_y) >$   
 $\alpha_{17} \leftarrow \alpha_{17} + < [\text{id}_y, \text{id}_{f1}, \text{on}, 100] >$   
 $\alpha_{24} \leftarrow \alpha_{24} + a_y \cdot \text{cm}2$   
 $\beta_1 \leftarrow \beta_1, w_y = w(y, q_{\text{level}})$   
 $\beta_2 \leftarrow \beta_2, h_y = h(y, q_{\text{level}})$   
 $\beta_3 \leftarrow \beta_3, a_y = a(y, q_{\text{level}})$

$\beta_6 \leftarrow \beta_6, cm2 = cost\_m2 (q_{level}, uncovered)$

**g55: Adding a kitchen (programmer)**

$\alpha_4 \leftarrow \alpha_4, n_{dwellers} = \alpha_4$   
 $\alpha_6 \leftarrow \alpha_6, q_{level} = \alpha_6$   
 $\alpha_{10} \leftarrow \alpha_{10} + < [ki, 1] >$   
 $\alpha_{11} \leftarrow \alpha_{11} + < [ki, isolated, 100] >$   
 $\alpha_{13} \leftarrow \alpha_{13} + < [ki, id_{ki}, \emptyset, (cooking), (n_{dwellers}, 100), (isolated, 100), (q_{level}, 0), w_{ki}, h_{ki}, a_{ki}] >$   
 $id_{ki} = \max(id) + 1$   
 $\alpha_{14} \leftarrow \alpha_{14} + service (id_{ki}, a_{ki})$   
 $\alpha_{15} \leftarrow \alpha_{15} + < available, (0, -a_{ki}, 0, -a_{ki}), (0, 0, 0, 0), (0, -a_{ki}, 0, -a_{ki}), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + < used, (a_{ki}, a_{ki}, 0, a_{ki}), (0, 0, 0, 0), (a_{ki}, a_{ki}, 0, a_{ki}), -a_i / a_u + (a_i + a_{ki} / a_u + a_{ki}) >$   
 $\alpha_{17} \leftarrow \alpha_{17} + < [id_{ki}, id_{r1}, on, 100] >$   
 $\alpha_{24} \leftarrow \alpha_{24} + a_{ki} \cdot cm2$   
 $\beta_1 \leftarrow \beta_1, w_{ki, n} = w (ki, n_{dwellers}, isolated, q_{level})$   
 $\beta_2 \leftarrow \beta_2, h_{ki, n} = h (ki, n_{dwellers}, isolated, q_{level})$   
 $\beta_3 \leftarrow \beta_3, a_{ki, n} = a (ki, n_{dwellers}, isolated, q_{level})$   
 $\beta_6 \leftarrow \beta_6, cm2 = cost\_m2 (q_{level}, covered)$

**g56: Adding a living room (programmer)**

$\alpha_4 \leftarrow \alpha_4, n_{dwellers} = \alpha_4$   
 $\alpha_6 \leftarrow \alpha_6, q_{level} = \alpha_6$   
 $\alpha_{10} \leftarrow \alpha_{10} + < [lr, 1] >$   
 $\alpha_{11} \leftarrow \alpha_{11} + < [lr, isolated, 100] >$   
 $\alpha_{13} \leftarrow \alpha_{13} + < [lr, id_{lr}, \emptyset, (living, receiving), (n_{dwellers}, 100), (isolated, 100), (q_{level}, 0), w_{lr}, h_{lr}, a_{lr}] >$   
 $id_{lr} = \max(id) + 1$   
 $\alpha_{14} \leftarrow \alpha_{14} + living (id_{lr}, a_{lr})$   
 $\alpha_{15} \leftarrow \alpha_{15} + < available, (0, -a_{lr}, 0, -a_{lr}), (0, 0, 0, 0), (0, -a_{lr}, 0, -a_{lr}), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + < used, (a_{lr}, a_{lr}, 0, a_{lr}), (0, 0, 0, 0), (a_{lr}, a_{lr}, 0, a_{lr}), -a_i / a_u + (a_i + a_{lr} / a_u + a_{lr}) >$   
 $\alpha_{17} \leftarrow \alpha_{17} + < [id_{lr}, id_{r1}, on, 100] >$   
 $\alpha_{24} \leftarrow \alpha_{24} + a_{lr} \cdot cm2$   
 $\beta_1 \leftarrow \beta_1, w_{lr} = w (lr, n_{dwellers}, isolated, q_{level})$   
 $\beta_2 \leftarrow \beta_2, h_{lr} = h (lr, n_{dwellers}, isolated, q_{level})$   
 $\beta_3 \leftarrow \beta_3, a_{lr} = a (lr, n_{dwellers}, isolated, q_{level})$   
 $\beta_6 \leftarrow \beta_6, cm2 = cost\_m2 (q_{level}, covered)$

**g57: Adding a staircase when the number of floors is 2 (programmer)**

$\alpha_4 \leftarrow \alpha_4, n_{dwellers} = \alpha_4$   
 $\alpha_6 \leftarrow \alpha_6, q_{level} = \alpha_6$   
 $\alpha_8 \leftarrow \alpha_8, \alpha_9 = 2$   
 $\alpha_{11} \leftarrow \alpha_{11} + < [st, isolated, 100] >$   
 $\alpha_{13} \leftarrow \alpha_{13} + < [st, id_{st}, \emptyset, (circulation), (n_{dwellers}, 100), (isolated, 100), (q_{level}, \emptyset), w_{st}, h_{st}, a_{st}] >$   
 $id_{st} = \max(id) + 1$   
 $\alpha_{14} \leftarrow \alpha_{14} + living (id_{st}, a_{st})$   
 $\alpha_{15} \leftarrow \alpha_{15} + < available, (0, -a_{st}, 0, -a_{st}), (0, 0, 0, 0), (0, -a_{st}, 0, -a_{st}), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + < used, (0, a_{st}, 0, a_{st}), (0, a_{st}, 0, a_{st}), (0, 2 \cdot a_{st}, 0, 2 \cdot a_{st}), -a_i / a_u + (a_i / a_u + 2 \cdot a_{st}) >$   
 $\alpha_{17} \leftarrow \alpha_{17} + < [id_{st}, id_{r1}, on, 100], [id_{st}, id_{r2}, on, 100] >$   
 $\alpha_{24} \leftarrow \alpha_{24} + a_{st} \cdot cm2$   
 $\beta_1 \leftarrow \beta_1, w_{st} = w (st, q_{level})$   
 $\beta_2 \leftarrow \beta_2, h_{st} = h (st, q_{level})$   
 $\beta_3 \leftarrow \beta_3, A_{q, st} = a (st, q_{level})$   
 $\beta_6 \leftarrow \beta_6, cm2 = cost\_m2 (q_{level}, covered)$

---

**Bedrooms: couple / double / single (programmer)**

**g58: Adding a single bedroom when there is 1 bedroom and 1 dweller**

$\alpha_3 \leftarrow \alpha_3, \alpha_3 = \text{type}$   
 $\alpha_4 \leftarrow \alpha_4 + < 1, \emptyset >$   
 $\alpha_5 \leftarrow \alpha_5 + < 1, [(\text{couple}, 0), (\text{double}, 0), (\text{single}, 1)] >$   
 $\alpha_6 \leftarrow \alpha_6, q_{\text{level}} = \alpha_6$   
 $\alpha_{10} \leftarrow \alpha_{10} + < [\text{be}, 1, ((\text{couple}, 0), (\text{double}, 0), (\text{single}, 1))] >$   
 $\alpha_{11} \leftarrow \alpha_{11} + < [\text{single}, \emptyset, 0] >$   
 $\alpha_{13} \leftarrow \alpha_{13} + < [\text{be}, \text{id}_{\text{be}}, \emptyset, (\text{sleeping}), (\text{single}, 100), (\text{isolated}, 100), (q_{\text{level}}, 0), w_{\text{si}}, h_{\text{si}}, a_{\text{si}}] >$   
 $\text{id}_{\text{be}} = \max(\text{id}) + 1$   
 $\alpha_{14} \leftarrow \alpha_{14} + \text{sleeping}(\text{id}_{\text{be}}, a_{\text{si}})$   
 $\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, -a_{\text{si}}, 0, -a_{\text{si}}), (0, 0, 0, 0), (0, -a_{\text{si}}, 0, -a_{\text{si}}), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (a_{\text{si}}, a_{\text{si}}, 0, a_{\text{si}}), (0, 0, 0, 0), (a_{\text{si}}, a_{\text{si}}, 0, a_{\text{si}}), -a_i / a_u + (a_i + a_{\text{si}} / a_u + a_{\text{si}}) >$   
 $\alpha_{17} \leftarrow \alpha_{17} + < [\text{id}_{\text{be}}, \text{id}_{\text{r1}}, \text{on}, 100] >$   
 $\alpha_{24} \leftarrow \alpha_{24} + a_{\text{si}} \cdot \text{cm2}$   
 $\beta_1 \leftarrow \beta_1, w_{\text{si}} = w(\text{be}, \text{si}, \text{isolated}, q_{\text{level}})$   
 $\beta_2 \leftarrow \beta_2, h_{\text{si}} = h(\text{be}, \text{si}, \text{isolated}, q_{\text{level}})$   
 $\beta_3 \leftarrow \beta_3, a_{\text{si}} = a(\text{be}, \text{si}, \text{isolated}, q_{\text{level}})$   
 $\beta_6 \leftarrow \beta_6, \text{cm2} = \text{cost\_m2}(q_{\text{level}}, \text{covered})$

**g59: Adding a couple bedroom when there is only 1 bedroom and 2 dwellers**

$\alpha_3 \leftarrow \alpha_3, \alpha_3 = \text{type}$   
 $\alpha_4 \leftarrow \alpha_4 + < 1, \emptyset >$   
 $\alpha_5 \leftarrow \alpha_5 + < 1, [(\text{couple}, 1), (\text{double}, 0), (\text{single}, 0)] >$   
 $\alpha_6 \leftarrow \alpha_6, q_{\text{level}} = \alpha_6$   
 $\alpha_{10} \leftarrow \alpha_{10} + < [(\text{be}, 1, ((\text{couple}, 1), (\text{double}, 0), (\text{single}, 0)))] >$   
 $\alpha_{11} \leftarrow \alpha_{11} + < [\text{couple}, \emptyset, 0] >$   
 $\alpha_{13} \leftarrow \alpha_{13} + < [(\text{be}, \text{id}_{\text{be}}, \emptyset, (\text{sleeping}), (\text{couple}, 100), (\text{isolated}, 100), (q_{\text{level}}, 0), w_{\text{cu}}, h_{\text{cu}}, a_{\text{cu}}) >$   
 $\text{id}_{\text{be}} = \max(\text{id}) + 1$   
 $\alpha_{14} \leftarrow \alpha_{14} + \text{sleeping}(\text{id}_{\text{be}}, a_{\text{cu}})$   
 $\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, -a_{\text{cu}}, 0, -a_{\text{cu}}), (0, 0, 0, 0), (0, -a_{\text{cu}}, 0, -a_{\text{cu}}), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (a_{\text{cu}}, a_{\text{cu}}, 0, a_{\text{cu}}), (0, 0, 0, 0), (a_{\text{cu}}, a_{\text{cu}}, 0, a_{\text{cu}}), -a_i / a_u + (a_i + a_{\text{cu}} / a_u + a_{\text{cu}}) >$   
 $\alpha_{17} \leftarrow \alpha_{17} + < [\text{id}_{\text{be}}, \text{id}_{\text{r1}}, \text{on}, 100] >$   
 $\alpha_{24} \leftarrow \alpha_{24} + a_{\text{cu}} \cdot \text{cm2}$   
 $\beta_1 \leftarrow \beta_1, w_{\text{cu}} = w(\text{be}, \text{cu}, \text{isolated}, q_{\text{level}})$   
 $\beta_2 \leftarrow \beta_2, h_{\text{cu}} = h(\text{be}, \text{cu}, \text{isolated}, q_{\text{level}})$   
 $\beta_3 \leftarrow \beta_3, a_{\text{cu}} = a(\text{be}, \text{cu}, \text{isolated}, q_{\text{level}})$   
 $\beta_6 \leftarrow \beta_6, \text{cm2} = \text{cost\_m2}(q_{\text{level}}, \text{covered})$

**g60: Adding bedrooms with the appropriate capacity when there are 2 bedrooms and 2 dwellers**

$\alpha_3 \leftarrow \alpha_3, \alpha_3 = \text{type}$   
 $\alpha_4 \leftarrow \alpha_4 + < 2, \emptyset >$   
 $\alpha_5 \leftarrow \alpha_5 + < 2, [(\text{couple}, 0), (\text{double}, 0), (\text{single}, 2)] >$   
 $\alpha_6 \leftarrow \alpha_6, q_{\text{level}} = \alpha_6$   
 $\alpha_9 \leftarrow \alpha_9, n_{\text{floors}} = \alpha_9$   
 $\alpha_{10} \leftarrow \alpha_{10} + < [\text{be}, 2, ((\text{couple}, 0), (\text{double}, 0), (\text{single}, 2))] >$   
 $\alpha_{11} \leftarrow \alpha_{11} + < [\text{single}, \emptyset, 0], [\text{single}, \emptyset, 0] >$   
 $\alpha_{13} \leftarrow \alpha_{13} + < [\text{be}, \text{id}_{\text{be}1}, \emptyset, (\text{sleeping}), (\text{single}, 100), (\text{isolated}, 100), (q_{\text{level}}, 0), w_{\text{si}}, h_{\text{si}}, a_{\text{si}}],$   
 $\quad [\text{be}, \text{id}_{\text{be}2}, \emptyset, (\text{sleeping}), (\text{single}, 100), (\text{isolated}, 100), (q_{\text{level}}, 0), w_{\text{si}}, h_{\text{si}}, a_{\text{si}}] >$   
 $\text{id}_{\text{ben}} = \max(\text{id}) + 1, n \in \{1, 2\}$   
 $\alpha_{14} \leftarrow \alpha_{14} + \text{sleeping}(\text{id}_{\text{be}1}, a_{\text{si}})$   
 $\quad + \text{sleeping}(\text{id}_{\text{be}2}, a_{\text{si}})$   
 $n_{\text{floors}} = 1 \Rightarrow$   
 $\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, -2 \cdot a_{\text{si}}, 0, -2 \cdot a_{\text{si}}), (0, 0, 0, 0), (0, -2 \cdot a_{\text{si}}, 0, -2 \cdot a_{\text{si}}), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (2 \cdot a_{\text{si}}, 2 \cdot a_{\text{si}}, 0, 2 \cdot a_{\text{si}}),$   
 $\quad (0, 0, 0, 0),$   
 $\quad (2 \cdot a_{\text{si}}, 2 \cdot a_{\text{si}}, 0, 2 \cdot a_{\text{si}}),$   
 $\quad -a_i / a_u + (a_i + 2 \cdot a_{\text{si}} / a_u + 2 \cdot a_{\text{si}}) >$   
 $\alpha_{17} \leftarrow \alpha_{17} + < [\text{id}_{\text{be}1}, \text{id}_{\text{r1}}, \text{on}, 100], [\text{id}_{\text{be}2}, \text{id}_{\text{r1}}, \text{on}, 100] >$   
 $n_{\text{floors}} = 2 \Rightarrow$   
 $\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, 0, 0, 0), (0, -2 \cdot a_{\text{si}}, 0, -2 \cdot a_{\text{si}}), (0, -2 \cdot a_{\text{si}}, 0, -2 \cdot a_{\text{si}}), 0 >$

$$\alpha_{16} \leftarrow \alpha_{16} + \langle \text{used}, (0, 0, 0, 0), \\ (2 \cdot a_{si}, 2 \cdot a_{si}, 0, 2 \cdot a_{si}), \\ (2 \cdot a_{si}, 2 \cdot a_{si}, 0, 2 \cdot a_{si}), \\ - a_i / a_u + (a_i + 2 \cdot a_{si} / a_u + 2 \cdot a_{si}) \rangle$$

$$\alpha_{17} \leftarrow \alpha_{17} + \langle [\text{id}_{be1}, \text{id}_{r2}, \text{on}, 100], [\text{id}_{be2}, \text{id}_{r2}, \text{on}, 100] \rangle$$

$$\alpha_{24} \leftarrow \alpha_{24} + 2 \cdot a_{si} \cdot \text{cm2}$$

$$\beta_1 \leftarrow \beta_1, w_{si} = w(\text{be}, \text{si}, \text{isolated}, q_{level})$$

$$\beta_2 \leftarrow \beta_2, h_{si} = h(\text{be}, \text{si}, \text{isolated}, q_{level})$$

$$\beta_3 \leftarrow \beta_3, a_{si} = a(\text{be}, \text{si}, \text{isolated}, q_{level})$$

$$\beta_6 \leftarrow \beta_6, \text{cm2} = \text{cost\_m2}(q_{level}, \text{covered})$$

**g61: Adding bedrooms with the appropriate capacity when there are 2 bedrooms and 3 dwellers**

$$\alpha_3 \leftarrow \alpha_3, \alpha_3 = \text{type}$$

$$\alpha_4 \leftarrow \alpha_4 + \langle 3, \emptyset \rangle$$

$$\alpha_5 \leftarrow \alpha_5 + \langle 2, [(\text{couple}, 1), (\text{double}, 0), (\text{single}, 1)] \rangle$$

$$\alpha_6 \leftarrow \alpha_6, q_{level} = \alpha_6$$

$$\alpha_{10} \leftarrow \alpha_{10} + \langle [\text{be}, 2, ((\text{couple}, 1), (\text{double}, 0), (\text{single}, 1))]] \rangle$$

$$\alpha_{11} \leftarrow \alpha_{11} + \langle [\text{couple}, \emptyset, 0], [\text{single}, \emptyset, 0] \rangle$$

$$\alpha_{13} \leftarrow \alpha_{13} + \langle [\text{be}, \text{id}_{be1}, \emptyset, (\text{sleeping}), (\text{couple}, 100), (\text{isolated}, 100), (q_{level}, 0), w_{cu}, h_{cu}, a_{cu}], \\ [\text{be}, \text{id}_{be2}, \emptyset, (\text{sleeping}), (\text{single}, 100), (\text{isolated}, 100), (q_{level}, 0), w_{si}, h_{si}, a_{si}] \rangle$$

$$\text{id}_{ben} = \max(\text{id}) + 1, n \in \{1, 2\}$$

$$\alpha_{14} \leftarrow \alpha_{14} + \text{sleeping}(\text{id}_{be1}, a_{cu}) \\ + \text{sleeping}(\text{id}_{be2}, a_{si})$$

$$n_{\text{floors}} = 1 \Rightarrow$$

$$\alpha_{15} \leftarrow \alpha_{15} + \langle \text{available}, (0, -2 \cdot a_{si}, 0, -2 \cdot a_{si}), (0, 0, 0, 0), (0, -2 \cdot a_{si}, 0, -2 \cdot a_{si}), 0 \rangle$$

$$\alpha_{16} \leftarrow \alpha_{16} + \langle \text{used}, (2 \cdot a_{si}, 2 \cdot a_{si}, 0, 2 \cdot a_{si}), \\ (0, 0, 0, 0), \\ (2 \cdot a_{si}, 2 \cdot a_{si}, 0, 2 \cdot a_{si}), \\ - a_i / a_u + (a_i + 2 \cdot a_{si} / a_u + 2 \cdot a_{si}) \rangle$$

$$\alpha_{17} \leftarrow \alpha_{17} + \langle [\text{id}_{be1}, \text{id}_{r1}, \text{on}, 100], [\text{id}_{be2}, \text{id}_{r1}, \text{on}, 100] \rangle$$

$$n_{\text{floors}} = 2 \Rightarrow$$

$$\alpha_{15} \leftarrow \alpha_{15} + \langle \text{available}, (0, 0, 0, 0), (0, -2 \cdot a_{si}, 0, -2 \cdot a_{si}), (0, -2 \cdot a_{si}, 0, -2 \cdot a_{si}), 0 \rangle$$

$$\alpha_{16} \leftarrow \alpha_{16} + \langle \text{used}, (0, 0, 0, 0), \\ (2 \cdot a_{si}, 2 \cdot a_{si}, 0, 2 \cdot a_{si}), \\ (2 \cdot a_{si}, 2 \cdot a_{si}, 0, 2 \cdot a_{si}), \\ - a_i / a_u + (a_i + 2 \cdot a_{si} / a_u + 2 \cdot a_{si}) \rangle$$

$$\alpha_{17} \leftarrow \alpha_{17} + \langle [\text{id}_{be1}, \text{id}_{r2}, \text{on}, 100], [\text{id}_{be2}, \text{id}_{r2}, \text{on}, 100] \rangle$$

$$\alpha_{24} \leftarrow \alpha_{24} + (a_{cu} + a_{si}) \cdot \text{cm2}$$

$$\beta_1 \leftarrow \beta_1, w_{cu} = w(\text{be}, \text{cu}, \text{isolated}, q_{level}), w_{q, si} = w(\text{be}, \text{si}, \text{isolated}, q_{level})$$

$$\beta_2 \leftarrow \beta_2, h_{cu} = h(\text{be}, \text{cu}, \text{isolated}, q_{level}), h_{q, si} = h(\text{be}, \text{si}, \text{isolated}, q_{level})$$

$$\beta_3 \leftarrow \beta_3, a_{cu} = a(\text{be}, \text{cu}, \text{isolated}, q_{level}), a_{q, si} = a(\text{be}, \text{si}, \text{isolated}, q_{level})$$

$$\beta_6 \leftarrow \beta_6, \text{cm2} = \text{cost\_m2}(q_{level}, \text{covered})$$

**g62: Adding bedrooms with the appropriate capacity when there are 2 bedrooms and 4 dwellers**

$$\alpha_3 \leftarrow \alpha_3, \alpha_3 = \text{type}$$

$$\alpha_4 \leftarrow \alpha_4 + \langle 4, \emptyset \rangle$$

$$\alpha_5 \leftarrow \alpha_5 + \langle 2, [(\text{couple}, 1), (\text{double}, 1), (\text{single}, 0)] \rangle$$

$$\alpha_6 \leftarrow \alpha_6, q_{level} = \alpha_6$$

$$\alpha_{10} \leftarrow \alpha_{10} + [\text{be}, 2, ((\text{couple}, 1), (\text{double}, 1), (\text{single}, 0))]$$

$$\alpha_{11} \leftarrow \alpha_{11} + \langle [\text{couple}, \emptyset, 0], [\text{double}, \emptyset, 0] \rangle$$

$$\alpha_{13} \leftarrow \alpha_{13} + \langle [\text{be}, \text{id}_{be1}, \emptyset, (\text{sleeping}), (\text{couple}, 100), (\text{isolated}, 100), (q_{level}, 0), w_{cu}, h_{cu}, a_{cu}], \\ [\text{be}, \text{id}_{be2}, \emptyset, (\text{sleeping}), (\text{double}, 100), (\text{isolated}, 100), (q_{level}, 0), w_{do}, h_{do}, a_{do}] \rangle$$

$$\text{id}_{ben} = \max(\text{id}) + 1, n \in \{1, 2\}$$

$$\alpha_{14} \leftarrow \alpha_{14} + \text{sleeping}(\text{id}_{be1}, a_{cu}) + \text{sleeping}(\text{id}_{be2}, a_{do})$$

$$n_{\text{floors}} = 1 \Rightarrow$$

$$\alpha_{15} \leftarrow \alpha_{15} + \langle \text{available}, (0, -2 \cdot a_{si}, 0, -2 \cdot a_{si}), (0, 0, 0, 0), (0, -2 \cdot a_{si}, 0, -2 \cdot a_{si}), 0 \rangle$$

$$\alpha_{16} \leftarrow \alpha_{16} + \langle \text{used}, (2 \cdot a_{si}, 2 \cdot a_{si}, 0, 2 \cdot a_{si}), \\ (0, 0, 0, 0), \\ (2 \cdot a_{si}, 2 \cdot a_{si}, 0, 2 \cdot a_{si}),$$



$$- a_i / a_u + (a_i + 2 \cdot a_{si} / a_u + 2 \cdot a_{si}) >$$

$$\alpha_{17} \leftarrow \alpha_{17} + < [id_{be1}, id_{r1}, on, 100], [id_{be2}, id_{r1}, on, 100] >$$

$n_{floors} = 2 \Rightarrow$

$$\alpha_{15} \leftarrow \alpha_{15} + < available, (0, 0, 0, 0), (0, -2 \cdot a_{si}, 0, -2 \cdot a_{si}), (0, -2 \cdot a_{si}, 0, -2 \cdot a_{si}), 0 >$$

$$\alpha_{16} \leftarrow \alpha_{16} + < used, (0, 0, 0, 0),$$

$$(2 \cdot a_{si}, 2 \cdot a_{si}, 0, 2 \cdot a_{si}),$$

$$(2 \cdot a_{si}, 2 \cdot a_{si}, 0, 2 \cdot a_{si}),$$

$$- a_i / a_u + (a_i + 2 \cdot a_{si} / a_u + 2 \cdot a_{si}) >$$

$$\alpha_{17} \leftarrow \alpha_{17} + < [id_{be1}, id_{l2}, on, 100], [id_{be2}, id_{l2}, on, 100] >$$

$$\alpha_{24} \leftarrow \alpha_{24} + (a_{cu} + a_{do}) \cdot cm2$$

$$\beta_1 \leftarrow \beta_1, w_{cu} = w (be, cu, isolated, q_{level}), w_{q, do} = w (be, do, isolated, q_{level})$$

$$\beta_2 \leftarrow \beta_2, h_{cu} = h (be, cu, isolated, q_{level}), h_{q, do} = h (be, do, isolated, q_{level})$$

$$\beta_3 \leftarrow \beta_3, a_{cu} = a (be, cu, isolated, q_{level}), a_{q, do} = a (be, do, isolated, q_{level})$$

$$\beta_6 \leftarrow \beta_6, cm2 = cost\_m2 (q_{level}, covered)$$

**g63: Adding bedrooms with the appropriate capacity when there are 3 bedrooms and 4 dwellers**

$$\alpha_3 \leftarrow \alpha_3, \alpha_3 = type$$

$$\alpha_4 \leftarrow \alpha_4 + < 4, \emptyset >$$

$$\alpha_5 \leftarrow \alpha_5 + < 3, [(couple, 1), (double, 0), (single, 2)] >$$

$$\alpha_6 \leftarrow \alpha_6, q_{level} = \alpha_6$$

$$\alpha_{10} \leftarrow \alpha_{10} + < [be, 3, ((couple, 1), (double, 0), (single, 2))] >$$

$$\alpha_{11} \leftarrow \alpha_{11} + < [couple, \emptyset, 0], [single, \emptyset, 0], [single, \emptyset, 0] >$$

$$\alpha_{13} \leftarrow \alpha_{13} + < [be, id_{be1}, \emptyset, (sleeping), (couple, 100), (isolated, 100), (q_{level}, 0), w_{cu}, h_{cu}, a_{cu}],$$

$$[be, id_{be2}, \emptyset, (sleeping), (single, 100), (isolated, 100), (q_{level}, 0), w_{si}, h_{si}, a_{si}],$$

$$[be, id_{be3}, \emptyset, (sleeping), (single, 100), (isolated, 100), (q_{level}, 0), w_{si}, h_{si}, a_{si}] >$$

$$id_{ben} = \max(id) + 1, n \in \{1, 2, 3\}$$

$$\alpha_{14} \leftarrow \alpha_{14} + sleeping (id_{be1}, a_{cu}) + f2 (id_{be2}, a_{si}) + f2 (id_{be3}, a_{si})$$

$$\alpha_{15} \leftarrow \alpha_{15} + < available, (0, -a_{cu}, 0, -a_{cu}),$$

$$(0, -2 \cdot a_{si}, 0, -2 \cdot a_{si}),$$

$$(0, - (a_{cu} + 2 \cdot a_{si}), 0, - (a_{cu} + 2 \cdot a_{si})),$$

$$0 >$$

$$\alpha_{16} \leftarrow \alpha_{16} + < used, (a_{cu}, a_{cu}, 0, a_{cu}),$$

$$(2 \cdot a_{si}, 2 \cdot a_{si}, 0, 2 \cdot a_{si}),$$

$$(a_{cu} + 2 \cdot a_{si}, a_{cu} + 2 \cdot a_{si}, 0, a_{cu} + 2 \cdot a_{si}),$$

$$- a_i / a_u + (a_i + a_{cu} + 2 \cdot a_{si} / a_u + a_{cu} + 2 \cdot a_{si}) >$$

$$\alpha_{17} \leftarrow \alpha_{17} + < [id_{be1}, id_{r1}, on, 100], [id_{be2}, id_{l2}, on, 100], [id_{be3}, id_{l2}, on, 100] >$$

$$\alpha_{24} \leftarrow \alpha_{24} + (a_{cu} + 2 \cdot a_{si}) \cdot cm2$$

$$\beta_1 \leftarrow \beta_1, w_{cu} = w (be, cu, isolated, q_{level}), w_{q, si} = w (be, si, isolated, q_{level})$$

$$\beta_2 \leftarrow \beta_2, h_{cu} = h (be, cu, isolated, q_{level}), h_{q, si} = h (be, si, isolated, q_{level})$$

$$\beta_3 \leftarrow \beta_3, a_{cu} = a (be, cu, isolated, q_{level}), a_{q, si} = a (be, si, isolated, q_{level})$$

$$\beta_6 \leftarrow \beta_6, cm2 = cost\_m2 (q_{level}, covered)$$

**g64: Adding bedrooms with the appropriate capacity when there are 3 bedrooms and 5 dwellers**

$$\alpha_3 \leftarrow \alpha_3, \alpha_3 = type$$

$$\alpha_4 \leftarrow \alpha_4 + < 5, \emptyset >$$

$$\alpha_5 \leftarrow \alpha_5 + < 3, [(couple, 1), (double, 1), (single, 1)] >$$

$$\alpha_6 \leftarrow \alpha_6, q_{level} = \alpha_6$$

$$\alpha_{10} \leftarrow \alpha_{10} + < [be, 3, ((couple, 1), (double, 2), (single, 2))] >$$

$$\alpha_{11} \leftarrow \alpha_{11} + < [couple, \emptyset, 0], [double, \emptyset, 0], [single, \emptyset, 0] >$$

$$\alpha_{13} \leftarrow \alpha_{13} + < [be, ?id_{be1}, \emptyset, (sleeping), (couple, 100), (isolated, 100), (q_{level}, 0), w_{cu}, h_{cu}, a_{cu}],$$

$$[be, ?id_{be2}, \emptyset, (sleeping), (double, 100), (isolated, 100), (q_{level}, 0), w_{do}, h_{do}, a_{do}],$$

$$[be, ?id_{be3}, \emptyset, (sleeping), (single, 100), (isolated, 100), (q_{level}, 0), w_{si}, h_{si}, a_{si}] >$$

$$id_{ben} = \max(id) + 1, n \in \{1, 2, 3\}$$

$$\alpha_{14} \leftarrow \alpha_{14} + sleeping (id_{be1}, a_{cu}) + f2 (id_{be2}, a_{do}) + f2 (id_{be3}, a_{si})$$

$$\alpha_{15} \leftarrow \alpha_{15} + < available, (0, -a_{cu}, 0, -a_{cu}), (0, -(a_{do} + a_{si}), 0, -(a_{do} + a_{si})), (0, - (a_{cu} + a_{do} + a_{si}), 0, - (a_{cu} + a_{do} + a_{si})), 0 >$$

$$\alpha_{16} \leftarrow \alpha_{16} + < used, (a_{cu}, a_{cu}, 0, a_{cu}),$$

$$(a_{do} + a_{si}, a_{do} + a_{si}, 0, a_{do} + a_{si}),$$

$$(a_{cu} + a_{do} + a_{si}, a_{cu} + a_{do} + a_{si}, 0, a_{cu} + a_{do} + a_{si}),$$

$$- a_i / a_u + (a_i + a_{cu} + a_{do} + a_{si} / a_u + a_{cu} + a_{do} + a_{si}) >$$

$\alpha_{17} \leftarrow \alpha_{17} + \langle [id_{be1}, id_{r1}, on, 100], [id_{be2}, id_{r2}, on, 100], [id_{be3}, id_{r2}, on, 100] \rangle$   
 $\alpha_{24} \leftarrow \alpha_{24} + (a_{cu} + a_{do} + a_{si}) \cdot cm2$   
 $\beta_1 \leftarrow \beta_1, w_{cu} = w(\text{be, cu, isolated, } q_{level}), w_{do} = w(\text{be, do, isolated, } q_{level}), w_{si} = w(\text{be, si, isolated, } q_{level})$   
 $\beta_2 \leftarrow \beta_2, h_{cu} = h(\text{be, cu, isolated, } q_{level}), h_{do} = h(\text{be, do, isolated, } q_{level}), h_{si} = h(\text{be, si, isolated, } q_{level})$   
 $\beta_3 \leftarrow \beta_3, a_{cu} = a(\text{be, cu, isolated, } q_{level}), a_{do} = w(\text{be, do, isolated, } q_{level}), a_{si} = w(\text{be, si, isolated, } q_{level})$   
 $\beta_6 \leftarrow \beta_6, cm2 = \text{cost\_m2}(q_{level}, \text{covered})$

**g65: Adding bedrooms with the appropriate capacity when there are 3 bedrooms and 6 dwellers**

$\alpha_3 \leftarrow \alpha_3, \alpha_3 = \text{type}$   
 $\alpha_4 \leftarrow \alpha_4 + \langle 6, \emptyset \rangle$   
 $\alpha_5 \leftarrow \alpha_5 + \langle 3, [(couple, 1), (double, 2), (single, 0)] \rangle$   
 $\alpha_6 \leftarrow \alpha_6, q_{level} = \alpha_6$   
 $\alpha_{10} \leftarrow \alpha_{10} + \langle [be, 3, ((couple, 1), (double, 2), (single, 0))] \rangle$   
 $\alpha_{11} \leftarrow \alpha_{11} + \langle [couple, \emptyset, 0], [double, \emptyset, 0], [double, \emptyset, 0] \rangle$   
 $\alpha_{13} \leftarrow \alpha_{13} + \langle [be, ?id_{be1}, \emptyset, (\text{sleeping}), (couple, 100), (\text{isolated}, 100), (q_{level}, 0), w_{cu}, h_{cu}, a_{cu}],$   
 $[be, ?id_{be2}, \emptyset, (\text{sleeping}), (double, 100), (\text{isolated}, 100), (q_{level}, 0), w_{do}, h_{do}, a_{do}],$   
 $[be, ?id_{be3}, \emptyset, (\text{sleeping}), (double, 100), (\text{isolated}, 100), (q_{level}, 0), w_{do}, h_{do}, a_{do}], \rangle$   
 $id_{ben} = \max(id) + 1, n \in \{1, 2, 3\}$   
 $\alpha_{14} \leftarrow \alpha_{14} + \text{sleeping}(id_{be1}, a_{cu}) + f2(id_{be2}, a_{do}) + f2(id_{be3}, a_{do})$   
 $\alpha_{15} \leftarrow \alpha_{15} + \langle \text{available}, (0, -a_{cu}, 0, -a_{cu}), (0, -2 \cdot a_{do}, 0, -2 \cdot a_{do}), (0, -(a_{cu} + 2 \cdot a_{do}), 0, -(a_{cu} + 2 \cdot a_{do})), 0 \rangle$   
 $\alpha_{16} \leftarrow \alpha_{16} + \langle \text{used}, (a_{cu}, a_{cu}, 0, a_{cu}),$   
 $(2 \cdot a_{do}, 2 \cdot a_{do}, 0, 2 \cdot a_{do}),$   
 $(a_{cu} + 2 \cdot a_{do}, a_{cu} + 2 \cdot a_{do}, 0, a_{cu} + 2 \cdot a_{do}),$   
 $-a_i / a_u + (a_i + a_{cu} + 2 \cdot a_{do} / a_u + a_{cu} + 2 \cdot a_{do}) \rangle$   
 $\alpha_{17} \leftarrow \alpha_{17} + \langle [id_{be1}, id_{r1}, on, 100], [id_{be2}, id_{r2}, on, 100], [id_{be3}, id_{r2}, on, 100] \rangle$   
 $\alpha_{24} \leftarrow \alpha_{24} + (a_{cu} + 2 \cdot a_{do}) \cdot cm2$   
 $\beta_1 \leftarrow \beta_1, w_{cu} = w(\text{be, cu, isolated, } q_{level}), w_{do} = w(\text{be, do, isolated, } q_{level}),$   
 $\beta_2 \leftarrow \beta_2, h_{cu} = h(\text{be, cu, isolated, } q_{level}), h_{do} = h(\text{be, do, isolated, } q_{level}),$   
 $\beta_3 \leftarrow \beta_3, a_{cu} = a(\text{be, cu, isolated, } q_{level}), a_{do} = a(\text{be, do, isolated, } q_{level})$   
 $\beta_6 \leftarrow \beta_6, cm2 = \text{cost\_m2}(q_{level}, \text{covered})$

**g66: Adding bedrooms with the appropriate capacity when there are 4 bedrooms and 5 dwellers**

$\alpha_3 \leftarrow \alpha_3, \alpha_3 = \text{type}$   
 $\alpha_4 \leftarrow \alpha_4 + \langle 5, \emptyset \rangle$   
 $\alpha_5 \leftarrow \alpha_5 + \langle 4, [(couple, 1), (double, 0), (single, 3)] \rangle$   
 $\alpha_6 \leftarrow \alpha_6, q_{level} = \alpha_6$   
 $\alpha_{10} \leftarrow \alpha_{10} + \langle [be, 3, ((couple, 1), (double, 0), (single, 3))] \rangle$   
 $\alpha_{11} \leftarrow \alpha_{11} + \langle [couple, \emptyset, 0], [single, \emptyset, 0], [single, \emptyset, 0], [single, \emptyset, 0] \rangle$   
 $\alpha_{13} \leftarrow \alpha_{13} + \langle [be, ?id_{be1}, \emptyset, (\text{sleeping}), (couple, 100), (\text{isolated}, 100), (q_{level}, 0), \emptyset], w_{cu}, h_{cu}, a_{cu}],$   
 $[be, ?id_{be2}, \emptyset, (\text{sleeping}), (single, 100), (\text{isolated}, 100), (q_{level}, 0), \emptyset], w_{si}, h_{si}, a_{si}],$   
 $[be, ?id_{be3}, \emptyset, (\text{sleeping}), (single, 100), (\text{isolated}, 100), (q_{level}, 0), \emptyset], w_{si}, h_{si}, a_{si}],$   
 $[be, ?id_{be4}, \emptyset, (\text{sleeping}), (single, 100), (\text{isolated}, 100), (q_{level}, 0), \emptyset], w_{si}, h_{si}, a_{si}], \rangle$   
 $id_{ben} = \max(id) + 1, n \in \{1, 2, 3, 4\}$   
 $\alpha_{14} \leftarrow \alpha_{14} + \text{sleeping}(id_{be1}, a_{cu}) + f2(id_{be2}, a_{si}) + f2(id_{be3}, a_{si}) + f2(id_{be4}, a_{si})$   
 $\alpha_{15} \leftarrow \alpha_{15} + \langle \text{available}, (0, -a_{cu}, 0, -a_{cu}), (0, -3 \cdot a_{si}, 0, -3 \cdot a_{si}), (0, -(a_{cu} + 3 \cdot a_{si}), 0, -(a_{cu} + 3 \cdot a_{si})), 0 \rangle$   
 $\alpha_{16} \leftarrow \alpha_{16} + \langle \text{used}, (a_{cu}, a_{cu}, 0, a_{cu}),$   
 $(3 \cdot a_{si}, 3 \cdot a_{si}, 0, 3 \cdot a_{si}),$   
 $(a_{cu} + 3 \cdot a_{si}, a_{cu} + 3 \cdot a_{si}, 0, a_{cu} + 3 \cdot a_{si}),$   
 $-a_i / a_u + (a_i + a_{cu} + 3 \cdot a_{si} / a_u + a_{cu} + 3 \cdot a_{si}) \rangle$   
 $\alpha_{17} \leftarrow \alpha_{17} + \langle [id_{be1}, id_{r1}, on, 100], [id_{be2}, id_{r2}, on, 100], [id_{be3}, id_{r2}, on, 100], [id_{be4}, id_{r2}, on, 100] \rangle$   
 $\alpha_{24} \leftarrow \alpha_{24} + (a_{cu} + 3 \cdot a_{si}) \cdot cm2$   
 $\beta_1 \leftarrow \beta_1, w_{cu} = w(\text{be, cu, isolated, } q_{level}), w_{si} = w(\text{be, si, isolated, } q_{level})$   
 $\beta_2 \leftarrow \beta_2, h_{cu} = h(\text{be, cu, isolated, } q_{level}), h_{si} = h(\text{be, si, isolated, } q_{level})$   
 $\beta_3 \leftarrow \beta_3, a_{cu} = a(\text{be, cu, isolated, } q_{level}), a_{si} = a(\text{be, si, isolated, } q_{level})$   
 $\beta_6 \leftarrow \beta_6, cm2 = \text{cost\_m2}(q_{level}, \text{covered})$

**g67: Adding bedrooms with the appropriate capacity when there are 4 bedrooms and 6 dwellers**

$\alpha_3 \leftarrow \alpha_3, \alpha_3 = \text{type}$   
 $\alpha_4 \leftarrow \alpha_4 + \langle 6, \emptyset \rangle$

$\alpha_5 \leftarrow \alpha_5 + < 4, [(couple, 1), (double, 1), (single, 2)] >$   
 $\alpha_6 \leftarrow \alpha_6, q_{level} = \alpha_6$   
 $\alpha_{10} \leftarrow \alpha_{10} + < [be, 4, ((couple, 1), (double, 1), (single, 2))] >$   
 $\alpha_{11} \leftarrow \alpha_{11} + < [couple, \emptyset, 0], [double, \emptyset, 0], [single, \emptyset, 0], [single, \emptyset, 0] >$   
 $\alpha_{13} \leftarrow \alpha_{13} + < [be, ?id_{be1}, \emptyset, (sleeping), (couple, 100), (isolated, 100), (q_{level}, 0), \emptyset], w_{cu}, h_{cu}, a_{cu},$   
 $[be, ?id_{be2}, \emptyset, (sleeping), (double, 100), (isolated, 100), (q_{level}, 0), \emptyset], w_{do}, h_{do}, a_{do},$   
 $[be, ?id_{be3}, \emptyset, (sleeping), (single, 100), (isolated, 100), (q_{level}, 0), \emptyset], w_{si}, h_{si}, a_{si},$   
 $[be, ?id_{be4}, \emptyset, (sleeping), (single, 100), (isolated, 100), (q_{level}, 0), \emptyset], w_{si}, h_{si}, a_{si} >$   
 $id_{ben} = \max(id) + 1, n \in \{1, 2, 3, 4\}$   
 $\alpha_{14} \leftarrow \alpha_{14} + sleeping(id_{be1}, a_{cu}) + f_2(id_{be2}, a_{do}) + f_2(id_{be3}, a_{si}) + f_2(id_{be4}, a_{si})$   
 $\alpha_{15} \leftarrow \alpha_{15} + < available, (0, -a_{cu}, 0, -a_{cu}), (0, -(a_{do} + 2 \cdot a_{si}), 0, -(a_{do} + 2 \cdot a_{si})), (0, -(a_{do} + 2 \cdot a_{si}), 0, -(a_{do} + 2 \cdot a_{si})), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + < used, (a_{cu}, a_{cu}, 0, a_{cu}),$   
 $(a_{do} + 2 \cdot a_{si}, a_{do} + 2 \cdot a_{si}, 0, a_{do} + 2 \cdot a_{si}),$   
 $(a_{cu} + a_{do} + 2 \cdot a_{si}, a_{cu} + a_{do} + 2 \cdot a_{si}, 0, a_{cu} + a_{do} + 2 \cdot a_{si}),$   
 $-a_i / a_u + (a_i + a_{cu} + a_{do} + 2 \cdot a_{si} / a_u + a_{cu} + a_{do} + 2 \cdot a_{si}) >$   
 $\alpha_{17} \leftarrow \alpha_{17} + < [id_{be1}, id_{f1}, on, 100], [id_{be2}, id_{f2}, on, 100], [id_{be3}, id_{f2}, on, 100], [id_{be4}, id_{f2}, on, 100] >$   
 $\alpha_{24} \leftarrow \alpha_{24} + (a_{cu} + a_{do} + 2 \cdot a_{si}) \cdot cm_2$   
 $\beta_1 \leftarrow \beta_1, w_{cu} = w(\text{be}, cu, isolated, q_{level}), w_{do} = w(\text{be}, do, isolated, q_{level}), w_{si} = w(\text{be}, si, isolated, q_{level})$   
 $\beta_2 \leftarrow \beta_2, h_{cu} = h(\text{be}, cu, isolated, q_{level}), h_{do} = h(\text{be}, do, isolated, q_{level}), h_{si} = h(\text{be}, si, isolated, q_{level})$   
 $\beta_3 \leftarrow \beta_3, a_{cu} = a(\text{be}, cu, isolated, q_{level}), a_{do} = a(\text{be}, do, isolated, q_{level}), a_{si} = a(\text{be}, si, isolated, q_{level})$   
 $\beta_6 \leftarrow \beta_6, cm_2 = cost\_m2(q_{level}, covered)$

**g68: Adding bedrooms with the appropriate capacity when there are 4 bedrooms and 7 dwellers**

$\alpha_3 \leftarrow \alpha_3, \alpha_3 = \text{type}$   
 $\alpha_4 \leftarrow \alpha_4 + < 7, \emptyset >$   
 $\alpha_5 \leftarrow \alpha_5 + < 4, [(couple, 1), (double, 2), (single, 1)] >$   
 $\alpha_6 \leftarrow \alpha_6, q_{level} = \alpha_6$   
 $\alpha_{10} \leftarrow \alpha_{10} + < [be, 4, ((couple, 1), (double, 2), (single, 1))] >$   
 $\alpha_{11} \leftarrow \alpha_{11} + < [couple, \emptyset, 0], [double, \emptyset, 0], [double, \emptyset, 0], [single, \emptyset, 0] >$   
 $\alpha_{13} \leftarrow \alpha_{13} + < [be, ?id_{be1}, \emptyset, (sleeping), (couple, 100), (isolated, 100), (q_{level}, 0), \emptyset], w_{cu}, h_{cu}, a_{cu},$   
 $[be, ?id_{be2}, \emptyset, (sleeping), (double, 100), (isolated, 100), (q_{level}, 0), \emptyset], w_{do}, h_{do}, a_{do},$   
 $[be, ?id_{be3}, \emptyset, (sleeping), (double, 100), (isolated, 100), (q_{level}, 0), \emptyset], w_{do}, h_{do}, a_{do},$   
 $[be, ?id_{be4}, \emptyset, (sleeping), (single, 100), (isolated, 100), (q_{level}, 0), \emptyset], w_{si}, h_{si}, a_{si} >$   
 $id_{ben} = \max(id) + 1, n \in \{1, 2, 3, 4\}$   
 $\alpha_{14} \leftarrow \alpha_{14} + sleeping(id_{be1}, a_{cu}) + f_2(id_{be2}, a_{do}) + f_2(id_{be3}, a_{do}) + f_2(id_{be4}, a_{si})$   
 $\alpha_{15} \leftarrow \alpha_{15} + < available, (0, -a_{cu}, 0, -a_{cu}), (0, -(2 \cdot a_{do} + a_{si}), 0, -(2 \cdot a_{do} + a_{si})), (0, -(2 \cdot a_{do} + a_{si}), 0, -(2 \cdot a_{do} + a_{si})), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + < used, (a_{cu}, a_{cu}, 0, a_{cu}),$   
 $(2 \cdot a_{do} + a_{si}, 2 \cdot a_{do} + a_{si}, 0, 2 \cdot a_{do} + a_{si}),$   
 $(a_{cu} + 2 \cdot a_{do} + a_{si}, a_{cu} + 2 \cdot a_{do} + a_{si}, 0, a_{cu} + 2 \cdot a_{do} + a_{si}),$   
 $-a_i / a_u + (a_i + a_{cu} + 2 \cdot a_{do} + a_{si} / a_u + a_{cu} + 2 \cdot a_{do} + a_{si}) >$   
 $\alpha_{17} \leftarrow \alpha_{17} + < [id_{be1}, id_{f1}, on, 100], [id_{be2}, id_{f2}, on, 100], [id_{be3}, id_{f2}, on, 100], [id_{be4}, id_{f2}, on, 100] >$   
 $\alpha_{24} \leftarrow \alpha_{24} + (a_{cu} + 2 \cdot a_{do} + a_{si}) \cdot cm_2$   
 $\beta_1 \leftarrow \beta_1, w_{cu} = w(\text{be}, cu, isolated, q_{level}), w_{do} = w(\text{be}, do, isolated, q_{level}), w_{si} = w(\text{be}, si, isolated, q_{level})$   
 $\beta_2 \leftarrow \beta_2, h_{cu} = h(\text{be}, cu, isolated, q_{level}), h_{do} = h(\text{be}, do, isolated, q_{level}), h_{si} = h(\text{be}, si, isolated, q_{level})$   
 $\beta_3 \leftarrow \beta_3, a_{cu} = a(\text{be}, cu, isolated, q_{level}), a_{do} = a(\text{be}, do, isolated, q_{level}), a_{si} = a(\text{be}, si, isolated, q_{level})$   
 $\beta_6 \leftarrow \beta_6, cm_2 = cost\_m2(q_{level}, covered)$

**g69: Adding bedrooms with the appropriate capacity when there are 5 bedrooms and 7 dwellers**

$\alpha_3 \leftarrow \alpha_3, \alpha_3 = \text{type}$   
 $\alpha_4 \leftarrow \alpha_4 + < 7, \emptyset >$   
 $\alpha_5 \leftarrow \alpha_5 + < 5, [(couple, 1), (double, 1), (single, 3)] >$   
 $\alpha_6 \leftarrow \alpha_6, q_{level} = \alpha_6$   
 $\alpha_{10} \leftarrow \alpha_{10} + < [be, 5, ((couple, 1), (double, 1), (single, 3))] >$   
 $\alpha_{11} \leftarrow \alpha_{11} + < [couple, \emptyset, 0], [double, \emptyset, 0], [single, \emptyset, 0], [single, \emptyset, 0], [single, \emptyset, 0] >$   
 $\alpha_{13} \leftarrow \alpha_{13} + < [be, ?id_{be1}, \emptyset, (sleeping), (couple, 100), (isolated, 100), (q_{level}, 0), \emptyset], w_{cu}, h_{cu}, a_{cu},$   
 $[be, ?id_{be2}, \emptyset, (sleeping), (double, 100), (isolated, 100), (q_{level}, 0), \emptyset], w_{do}, h_{do}, a_{do},$   
 $[be, ?id_{be3}, \emptyset, (sleeping), (single, 100), (isolated, 100), (q_{level}, 0), \emptyset], w_{si}, h_{si}, a_{si},$   
 $[be, ?id_{be4}, \emptyset, (sleeping), (single, 100), (isolated, 100), (q_{level}, 0), \emptyset], w_{si}, h_{si}, a_{si},$   
 $[be, ?id_{be5}, \emptyset, (sleeping), (single, 100), (isolated, 100), (q_{level}, 0), \emptyset], w_{si}, h_{si}, a_{si} >$

$$id_{ben} = \max(id) + 1, n \in \{1, 2, 3, 4\}$$

$$\alpha_{14} \leftarrow \alpha_{14} + \text{sleeping}(id_{be1}, a_{cu}) + f2(id_{be2}, a_{do}) + f2(id_{be3}, a_{si}) + f2(id_{be4}, a_{si}) + f2(id_{be5}, a_{si})$$

$$\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, -a_{cu}, 0, -a_{cu}), (0, -(a_{do} + 3 \cdot a_{si}), 0, -(a_{do} + 3 \cdot a_{si})), (0, -(a_{do} + 3 \cdot a_{si}), 0, -(a_{do} + 3 \cdot a_{si})), 0 >$$

$$\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (a_{cu}, a_{cu}, 0, a_{cu}),$$

$$(a_{do} + 3 \cdot a_{si}, a_{do} + 3 \cdot a_{si}, 0, a_{do} + 3 \cdot a_{si}),$$

$$(a_{cu} + a_{do} + 3 \cdot a_{si}, a_{cu} + a_{do} + 3 \cdot a_{si}, 0, a_{cu} + a_{do} + 3 \cdot a_{si}),$$

$$- a_i / a_u + (a_i + a_{cu} + a_{do} + 3 \cdot a_{si}) / a_u + a_{cu} + a_{do} + 3 \cdot a_{si} >$$

$$\alpha_{17} \leftarrow \alpha_{17} + < [id_{be1}, id_{f1}, on, 100], [id_{be2}, id_{f2}, on, 100], [id_{be3}, id_{f2}, on, 100], [id_{be4}, id_{f2}, on, 100], [id_{be5}, id_{f2}, on, 100] >$$

$$\alpha_{24} \leftarrow \alpha_{24} + (a_{cu} + a_{do} + 3 \cdot a_{si}) \cdot cm2$$

$$\beta_1 \leftarrow \beta_1, W_{cu} = w(\text{be}, cu, \text{isolated}, q_{level}), W_{do} = w(\text{be}, do, \text{isolated}, q_{level}), W_{si} = w(\text{be}, si, \text{isolated}, q_{level})$$

$$\beta_2 \leftarrow \beta_2, h_{cu} = h(\text{be}, cu, \text{isolated}, q_{level}), h_{do} = h(\text{be}, do, \text{isolated}, q_{level}), h_{si} = h(\text{be}, si, \text{isolated}, q_{level})$$

$$\beta_3 \leftarrow \beta_3, a_{cu} = a(\text{be}, cu, \text{isolated}, q_{level}), a_{do} = a(\text{be}, do, \text{isolated}, q_{level}), a_{si} = a(\text{be}, si, \text{isolated}, q_{level})$$

$$\beta_6 \leftarrow \beta_6, cm2 = \text{cost\_m2}(q_{level}, \text{covered})$$

**g70: Adding bedrooms with the appropriate capacity when there are 5 bedrooms and 8 dwellers**

$$\alpha_3 \leftarrow \alpha_3, \alpha_3 = \text{type}$$

$$\alpha_4 \leftarrow \alpha_4 + < 8, \emptyset >$$

$$\alpha_5 \leftarrow \alpha_5 + < 5, [(couple, 1), (double, 2), (single, 2)] >$$

$$\alpha_6 \leftarrow \alpha_6, q_{level} = \alpha_6$$

$$\alpha_{10} \leftarrow \alpha_{10} + < [\text{be}, 5, ((couple, 1), (double, 2), (single, 2))] >$$

$$\alpha_{11} \leftarrow \alpha_{11} + < [\text{couple}, \emptyset, 0], [\text{double}, \emptyset, 0], [\text{double}, \emptyset, 0], [\text{single}, \emptyset, 0] [\text{single}, \emptyset, 0] >$$

$$\alpha_{13} \leftarrow \alpha_{13} + < [\text{be}, ?id_{be1}, \emptyset, (\text{sleeping}), (couple, 100), (\text{isolated}, 100), (q_{level}, 0), \emptyset], W_{cu}, h_{cu}, a_{cu},$$

$$[\text{be}, ?id_{be2}, \emptyset, (\text{sleeping}), (double, 100), (\text{isolated}, 100), (q_{level}, 0), \emptyset], W_{do}, h_{do}, a_{do},$$

$$[\text{be}, ?id_{be3}, \emptyset, (\text{sleeping}), (double, 100), (\text{isolated}, 100), (q_{level}, 0), \emptyset], W_{do}, h_{do}, a_{do},$$

$$[\text{be}, ?id_{be4}, \emptyset, (\text{sleeping}), (single, 100), (\text{isolated}, 100), (q_{level}, 0), \emptyset], W_{si}, h_{si}, a_{si},$$

$$[\text{be}, ?id_{be5}, \emptyset, (\text{sleeping}), (single, 100), (\text{isolated}, 100), (q_{level}, 0), \emptyset], W_{si}, h_{si}, a_{si} >$$

$$id_{ben} = \max(id) + 1, n \in \{1, 2, 3, 4\}$$

$$\alpha_{14} \leftarrow \alpha_{14} + \text{sleeping}(id_{be1}, a_{cu}) + f2(id_{be2}, a_{do}) + f2(id_{be3}, a_{do}) + f2(id_{be4}, a_{si}) + f2(id_{be5}, a_{si})$$

$$\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, -a_{cu}, 0, -a_{cu}),$$

$$(0, -(2 \cdot (a_{do} + a_{si})), 0, -(2 \cdot (a_{do} + a_{si}))),$$

$$(0, -(2 \cdot (a_{do} + a_{si})), 0, -(2 \cdot (a_{do} + a_{si}))),$$

$$0 >$$

$$\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (a_{cu}, a_{cu}, 0, a_{cu}),$$

$$(2 \cdot (a_{do} + a_{si}), 2 \cdot (a_{do} + a_{si}), 0, 2 \cdot (a_{do} + a_{si})),$$

$$(a_{cu} + 2 \cdot (a_{do} + a_{si}), a_{cu} + 2 \cdot (a_{do} + a_{si}), 0, a_{cu} + 2 \cdot (a_{do} + a_{si})),$$

$$- a_i / a_u + (a_i + a_{cu} + 2 \cdot (a_{do} + a_{si})) / a_u + a_{cu} + 2 \cdot (a_{do} + a_{si}) >$$

$$\alpha_{17} \leftarrow \alpha_{17} + < [id_{be1}, id_{f1}, on, 100], [id_{be2}, id_{f2}, on, 100], [id_{be3}, id_{f2}, on, 100], [id_{be4}, id_{f2}, on, 100], [id_{be5}, id_{f2}, on, 100] >$$

$$\alpha_{24} \leftarrow \alpha_{24} + (a_{cu} + 2 \cdot (a_{do} + a_{si})) \cdot cm2$$

$$\beta_1 \leftarrow \beta_1, W_{cu} = w(\text{be}, do, \text{isolated}, q_{level}), W_{do} = w(q_{level}, do), W_{si} = w(\text{be}, si, \text{isolated}, q_{level})$$

$$\beta_2 \leftarrow \beta_2, h_{cu} = h(\text{be}, do, \text{isolated}, q_{level}), h_{do} = h(q_{level}, do), h_{si} = h(\text{be}, si, \text{isolated}, q_{level})$$

$$\beta_3 \leftarrow \beta_3, a_{cu} = a(\text{be}, do, \text{isolated}, q_{level}), a_{do} = a(q_{level}, do), a_{si} = a(\text{be}, si, \text{isolated}, q_{level})$$

$$\beta_6 \leftarrow \beta_6, cm2 = \text{cost\_m2}(q_{level}, \text{covered})$$

**g71: Adding bedrooms with the appropriate capacity when there are 5 bedrooms and 9 dwellers**

$$\alpha_3 \leftarrow \alpha_3, \alpha_3 = \text{type}$$

$$\alpha_4 \leftarrow \alpha_4 + < 9, \emptyset >$$

$$\alpha_5 \leftarrow \alpha_5 + < 5, [(couple, 1), (double, 3), (single, 1)] >$$

$$\alpha_6 \leftarrow \alpha_6, q_{level} = \alpha_6$$

$$\alpha_{10} \leftarrow \alpha_{10} + < [\text{be}, 5, ((couple, 1), (double, 3), (single, 1))] >$$

$$\alpha_{11} \leftarrow \alpha_{11} + < [\text{couple}, \emptyset, 0], [\text{double}, \emptyset, 0], [\text{double}, \emptyset, 0], [\text{double}, \emptyset, 0] [\text{single}, \emptyset, 0] >$$

$$\alpha_{13} \leftarrow \alpha_{13} + < [\text{be}, ?id_{be1}, \emptyset, (\text{sleeping}), (couple, 100), (\text{isolated}, 100), (q_{level}, 0), \emptyset], W_{cu}, h_{cu}, a_{cu},$$

$$[\text{be}, ?id_{be2}, \emptyset, (\text{sleeping}), (double, 100), (\text{isolated}, 100), (q_{level}, 0), \emptyset], W_{do}, h_{do}, a_{do},$$

$$[\text{be}, ?id_{be3}, \emptyset, (\text{sleeping}), (double, 100), (\text{isolated}, 100), (q_{level}, 0), \emptyset], W_{do}, h_{do}, a_{do},$$

$$[\text{be}, ?id_{be4}, \emptyset, (\text{sleeping}), (double, 100), (\text{isolated}, 100), (q_{level}, 0), \emptyset], W_{do}, h_{do}, a_{do},$$

$$[\text{be}, ?id_{be5}, \emptyset, (\text{sleeping}), (single, 100), (q_{level}, 0), \emptyset], W_{si}, h_{si}, a_{si} >$$

$$id_{ben} = \max(id) + 1, n \in \{1, 2, 3, 4\}$$

$$\alpha_{14} \leftarrow \alpha_{14} + \text{sleeping}(id_{be1}, a_{cu}) + f2(id_{be2}, a_{do}) + f2(id_{be3}, a_{do}) + f2(id_{be4}, a_{do}) + f2(id_{be5}, a_{si})$$

$$\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, -a_{cu}, 0, -a_{cu}), (0, -(3 \cdot a_{do} + a_{si}), 0, -(3 \cdot a_{do} + a_{si})), (0, -(3 \cdot a_{do} + a_{si}), 0, -(3 \cdot a_{do} + a_{si})), 0 >$$

$$\alpha_{16} \leftarrow \alpha_{16} + \text{used}, (a_{cu}, a_{cu}, 0, a_{cu}),$$

$$(3 \cdot a_{do} + a_{si}, 3 \cdot a_{do} + a_{si}, 0, 3 \cdot a_{do} + a_{si}),$$

$$(a_{cu} + 3 \cdot a_{do} + a_{si}, a_{cu} + 3 \cdot a_{do} + a_{si}, 0, a_{cu} + 3 \cdot a_{do} + a_{si}),$$

$$- a_i / a_u + (a_i + a_{cu} + 3 \cdot a_{do} + a_{si} / a_u + a_{cu} + 3 \cdot a_{do} + a_{si}) >$$

$$\alpha_{17} \leftarrow \alpha_{17} + \text{[id}_{be1}, \text{id}_{r1}, \text{on}, 100], [\text{id}_{be2}, \text{id}_{l2}, \text{on}, 100], [\text{id}_{be3}, \text{id}_{l2}, \text{on}, 100], [\text{id}_{be4}, \text{id}_{l2}, \text{on}, 100], [\text{id}_{be5}, \text{id}_{l2}, \text{on}, 100] >$$

$$\alpha_{24} \leftarrow \alpha_{24} + (a_{cu} + 3 \cdot a_{do} + a_{si}) \cdot \text{cm2}$$

$$\beta_1 \leftarrow \beta_1, w_{cu} = w(\text{be}, \text{cu}, \text{isolated}, q_{level}), w_{do} = w(\text{be}, \text{do}, \text{isolated}, q_{level}), w_{si} = w(\text{be}, \text{si}, \text{isolated}, q_{level})$$

$$\beta_2 \leftarrow \beta_2, h_{cu} = h(\text{be}, \text{cu}, \text{isolated}, q_{level}), h_{do} = h(\text{be}, \text{do}, \text{isolated}, q_{level}), h_{si} = h(\text{be}, \text{si}, \text{isolated}, q_{level})$$

$$\beta_3 \leftarrow \beta_3, a_{cu} = a(\text{be}, \text{cu}, \text{isolated}, q_{level}), a_{do} = a(\text{be}, \text{do}, \text{isolated}, q_{level}), a_{si} = a(\text{be}, \text{si}, \text{isolated}, q_{level})$$

$$\beta_6 \leftarrow \beta_6, \text{cm2} = \text{cost\_m2}(q_{level}, \text{covered})$$

### Bathrooms: bathtub / shower / lavatory bathrooms (programmer)

#### g72: Adding a bathtub bathroom to the minimum list of spaces

$$\alpha_{10} \leftarrow \alpha_{10} + \text{[ba}, 1, \text{((bathtub}, 1))]} >$$

#### g73: Adding a lavatory bathroom to the minimum list of spaces

$$\alpha_4 \leftarrow \alpha_4, n_{users} = \alpha_4$$

$$\alpha_8 \leftarrow \alpha_8, n_{floors} = \alpha_8$$

$$\alpha_{10} \leftarrow \alpha_{10} + \text{[ba}, 1, \text{((lavatory}, 1))]} >$$

$$(n_{users} \in \{2, 3, 4, 5, 6\} \wedge n_{floors} = 2) \vee n_{users} = 7$$

#### g74: Adding a shower bathroom to the minimum list of spaces

$$\alpha_4 \leftarrow \alpha_4, n_{users} \in \{8, 9\}$$

$$\alpha_{10} \leftarrow \alpha_{10} + \text{[ba}, 1, \text{((shower}, 1))]} >$$

#### g75: Adding a bathtub bathroom as the main bathroom to the current list of spaces

$$\alpha_6 \leftarrow \alpha_6, q_{level} = \alpha_6$$

$$\alpha_8 \leftarrow \alpha_8, n_{floors} = \alpha_8$$

$$\alpha_{11} \leftarrow \alpha_{11} + \text{[bathtub}, \emptyset, 0]} >$$

$$\alpha_{13} \leftarrow \alpha_{13} + \text{[ba}, \text{id}_{bt}, \emptyset, \text{(hygiene)}, \text{(bathtub}, 100), \text{(isolated}, 100), \text{(q}_{level}, 0), \text{w}_{bt}, \text{h}_{bt}, \text{a}_{bt}]} >$$

$$n_{bathrooms} = 0$$

$$n_{floors} = 1 \Rightarrow$$

$$\alpha_{14} \leftarrow \alpha_{14} + \text{sleeping}(\text{id}_{bt}, \text{a}_{bt})$$

$$\alpha_{15} \leftarrow \alpha_{15} + \text{available}, (0, -\text{a}_{bt}, 0, -\text{a}_{bt}), (0, 0, 0, 0), (0, -\text{a}_{bt}, 0, -\text{a}_{bt}), 0 >$$

$$\alpha_{16} \leftarrow \alpha_{16} + \text{used}, (0, \text{a}_{bt}, 0, \text{a}_{bt}), (0, 0, 0, 0), (0, \text{a}_{bt}, 0, \text{a}_{bt}), -a_i / a_u + (a_i / a_u + \text{a}_{bt}) >$$

$$\alpha_{17} \leftarrow \alpha_{17} + \text{[id}_{bt}, \text{id}_{r1}, \text{on}, 100]} >$$

$$n_{floors} > 1 \Rightarrow$$

$$\alpha_{14} \leftarrow \alpha_{14} + f2(\text{id}_{bt}, \text{a}_{bt})$$

$$\alpha_{15} \leftarrow \alpha_{15} + \text{available}, (0, 0, 0, 0), (0, -\text{a}_{bt}, 0, -\text{a}_{bt}), (0, -\text{a}_{bt}, 0, -\text{a}_{bt}), 0 >$$

$$\alpha_{16} \leftarrow \alpha_{16} + \text{used}, (0, 0, 0, 0), (0, \text{a}_{bt}, 0, \text{a}_{bt}), (0, \text{a}_{bt}, 0, \text{a}_{bt}), -a_i / a_u + (a_i / a_u + \text{a}_{bt}) >$$

$$\alpha_{17} \leftarrow \alpha_{17} + \text{[id}_{bt}, \text{id}_{l2}, \text{on}, 100]} >$$

$$\alpha_{24} \leftarrow \alpha_{24} + \text{a}_{bt} \cdot \text{cm2}$$

$$\beta_1 \leftarrow \beta_1, w_{bt} = w(\text{ba}, \text{bt}, q_{level})$$

$$\beta_2 \leftarrow \beta_2, h_{bt} = h(\text{ba}, \text{bt}, q_{level})$$

$$\beta_3 \leftarrow \beta_3, a_{bt} = a(\text{ba}, \text{bt}, q_{level})$$

$$\beta_6 \leftarrow \beta_6, \text{cm2} = \text{cost\_m2}(q_{level}, \text{covered})$$

#### g76: Adding a lavatory bathroom as a second bathroom to the current list of spaces

$$\alpha_4 \leftarrow \alpha_4, n_{users} = \alpha_4$$

$$\alpha_6 \leftarrow \alpha_6, q_{level} = \alpha_6$$

$$\alpha_8 \leftarrow \alpha_8, n_{floors} = \alpha_8$$

$$\alpha_{11} \leftarrow \alpha_{11} + \text{[lavatory}, \emptyset, 0]} >$$

$$\alpha_{13} \leftarrow \alpha_{13} + \text{[ba}, \text{id}_v, \emptyset, \text{(hygiene)}, \text{(lavatory}, 100), \text{(isolated}, 100), \text{(q}_{level}, 0), \text{w}_v, \text{h}_v, \text{a}_v]} >$$

$$n_{bathrooms} = 1 \wedge q_{level} = \min \wedge [(n_{users} \in \{2, 3, 4, 5, 6\} \wedge n_{floors} = 2) \vee n_{users} = 7]$$

$$\text{id}_v = \max(\text{id}) + 1$$

$$\alpha_{14} \leftarrow \alpha_{14} + \text{sleeping}(\text{id}_v, \text{a}_v)$$

$$\alpha_{15} \leftarrow \alpha_{15} + \text{available}, (0, -\text{a}_v, 0, -\text{a}_v), (0, 0, 0, 0), (0, -\text{a}_v, 0, -\text{a}_v), 0 >$$

$$\alpha_{16} \leftarrow \alpha_{16} + \langle \text{used}, (0, a_v, 0, a_v), (0, 0, 0, 0), (0, a_v, 0, a_v), -a_i / a_u + (a_i / a_u + a_v) \rangle$$

$$\alpha_{17} \leftarrow \alpha_{17} + \langle [\text{id}_v, \text{id}_n, \text{on}, 100] \rangle$$

$$\alpha_{24} \leftarrow \alpha_{24} + a_v \cdot \text{cm2}$$

$$\beta_1 \leftarrow \beta_1, w_v = w(\text{ba}, \text{lv}, \text{qlevel})$$

$$\beta_2 \leftarrow \beta_2, h_v = h(\text{ba}, \text{lv}, \text{qlevel})$$

$$\beta_3 \leftarrow \beta_3, a_v = a(\text{ba}, \text{lv}, \text{qlevel})$$

$$\beta_6 \leftarrow \beta_6, \text{cm2} = \text{cost\_m2}(\text{qlevel}, \text{covered})$$

Note: the building regulations say that on the same floor of the living there should be a lavatory and that on the same floor of the kitchen there should also be a lavatory, as the living and kitchen are on the first floor, then the lavatory should be on the first floor. However, it would be possible to write the rule so that it checked the floor of the living and kitchen first and then assigned the lavatory to the same floor, unless there existed another bathroom on this floor.

**g77: Adding shower bathroom as a second bathroom to the current list of spaces**

$$\alpha_4 \leftarrow \alpha_4, n_{\text{users}} = \alpha_4$$

$$\alpha_6 \leftarrow \alpha_6, \text{qlevel} = \alpha_6$$

$$\alpha_{11} \leftarrow \alpha_{11} + \langle [\text{shower}, \emptyset, 0] \rangle$$

$$\alpha_{13} \leftarrow \alpha_{13} + \langle [\text{ba}, \text{id}_{\text{sh}}, \emptyset, (\text{hygiene}), (\text{shower}, 100), (\text{isolated}, 100), (\text{qlevel}, 0), w_{\text{sh}}, h_{\text{sh}}, a_{\text{sh}}] \rangle$$

$$n_{\text{bathrooms}} = 1 \wedge \text{qlevel} = \min \wedge n_{\text{users}} \in \{8, 9\}$$

$$\text{id}_{\text{sh}} = \max(\text{id}) + 1$$

$$\alpha_{14} \leftarrow \alpha_{14} + \text{sleeping}(\text{id}_{\text{sh}}, a_{\text{sh}})$$

$$\alpha_{15} \leftarrow \alpha_{15} + \langle \text{available}, (0, -a_{\text{sh}}, 0, -a_{\text{sh}}), (0, 0, 0, 0), (0, -a_{\text{sh}}, 0, -a_{\text{sh}}), 0 \rangle$$

$$\alpha_{16} \leftarrow \alpha_{16} + \langle \text{used}, (0, a_{\text{sh}}, 0, a_{\text{sh}}), (0, 0, 0, 0), (0, a_{\text{sh}}, 0, a_{\text{sh}}), -a_i / a_u + (a_i / a_u + a_{\text{sh}}) \rangle$$

$$\alpha_{17} \leftarrow \alpha_{17} + \langle [\text{id}_{\text{sh}}, \text{id}_n, \text{on}, 100] \rangle$$

$$\alpha_{24} \leftarrow \alpha_{24} + a_{\text{sh}} \cdot \text{cm2}$$

$$\beta_1 \leftarrow \beta_1, w_{\text{sh}} = w(\text{ba}, \text{sh}, \text{qlevel})$$

$$\beta_2 \leftarrow \beta_2, h_{\text{sh}} = h(\text{ba}, \text{sh}, \text{qlevel})$$

$$\beta_3 \leftarrow \beta_3, a_{\text{sh}} = a(\text{ba}, \text{sh}, \text{qlevel})$$

$$\beta_6 \leftarrow \beta_6, \text{cm2} = \text{cost\_m2}(\text{qlevel}, \text{covered})$$

Same note as above.

**g78: Adding a lavatory bathroom as a second bathroom to the current list of spaces**

$$\alpha_4 \leftarrow \alpha_4, n_{\text{users}} = \alpha_4$$

$$\alpha_6 \leftarrow \alpha_6, \text{qlevel} = \alpha_6$$

$$\alpha_8 \leftarrow \alpha_8, n_{\text{floors}} = \alpha_8$$

$$\alpha_{11} \leftarrow \alpha_{11} + \langle [\text{lavatory}, \emptyset, 0] \rangle$$

$$\alpha_{13} \leftarrow \alpha_{13} + \langle [\text{ba}, \text{id}_v, \emptyset, (\text{hygiene}), (\text{lavatory}, 80), (\text{isolated}, 100), (\text{qlevel}, 0), w_v, h_v, a_v] \rangle$$

$$n_{\text{bathrooms}} = 1 \wedge \text{qlevel} = \text{med} \wedge [(n_{\text{users}} \in \{2, 3, 4, 5\} \wedge n_{\text{floors}} = 2) \vee n_{\text{users}} = 6]$$

$$\text{id}_v = \max(\text{id}) + 1$$

$$\alpha_{14} \leftarrow \alpha_{14} + \text{sleeping}(\text{id}_v, a_v)$$

$$\alpha_{15} \leftarrow \alpha_{15} + \langle \text{available}, (0, -a_v, 0, -a_v), (0, 0, 0, 0), (0, -a_v, 0, -a_v), 0 \rangle$$

$$\alpha_{16} \leftarrow \alpha_{16} + \langle \text{used}, (0, a_v, 0, a_v), (0, 0, 0, 0), (0, a_v, 0, a_v), -a_i / a_u + (a_i / a_u + a_v) \rangle$$

$$\alpha_{17} \leftarrow \alpha_{17} + \langle [\text{id}_v, \text{id}_n, \text{on}, 100] \rangle$$

$$\alpha_{24} \leftarrow \alpha_{24} + a_v \cdot \text{cm2}$$

$$\beta_1 \leftarrow \beta_1, w_v = w(\text{ba}, \text{lv}, \text{qlevel})$$

$$\beta_2 \leftarrow \beta_2, h_v = h(\text{ba}, \text{lv}, \text{qlevel})$$

$$\beta_3 \leftarrow \beta_3, a_{q, \text{lv}} = a(\text{ba}, \text{lv}, \text{qlevel})$$

$$\beta_6 \leftarrow \beta_6, \text{cm2} = \text{cost\_m2}(\text{qlevel}, \text{covered})$$

Same note as above.

**g79: Adding a shower bathroom as a second bathroom to the current list of spaces**

$$\alpha_4 \leftarrow \alpha_4, n_{\text{users}} = \alpha_4$$

$$\alpha_6 \leftarrow \alpha_6, \text{qlevel} = \alpha_6$$

$$\alpha_{11} \leftarrow \alpha_{11} + \langle [\text{shower}, \emptyset, 0] \rangle$$

$$\alpha_{13} \leftarrow \alpha_{13} + \langle [\text{ba}, \text{id}_{\text{sh}}, \emptyset, (\text{hygiene}), (\text{shower}, 80), (\text{isolated}, 100), (\text{qlevel}, 0), w_{\text{sh}}, h_{\text{sh}}, a_{\text{sh}}] \rangle$$

$$n_{\text{bathrooms}} = 1 \wedge \text{qlevel} = \text{med} \wedge n_{\text{users}} \in \{7, 8, 9\}$$

$$\text{id}_{\text{sh}} = \max(\text{id}) + 1$$

$$\alpha_{14} \leftarrow \alpha_{14} + \text{sleeping}(\text{id}_{\text{sh}}, a_{\text{sh}})$$

$\alpha_{15} \leftarrow \alpha_{15} + \text{available}, (0, -a_{sh}, 0, -a_{sh}), (0, 0, 0, 0), (0, -a_{sh}, 0, -a_{sh}), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + \text{used}, (0, a_{sh}, 0, a_{sh}), (0, 0, 0, 0), (0, a_{sh}, 0, a_{sh}), -a_i / a_u + (a_i / a_u + a_{sh}) >$   
 $\alpha_{17} \leftarrow \alpha_{17} + \text{[id}_{sh}, \text{id}_{f1}, \text{on}, 100]} >$   
 $\alpha_{24} \leftarrow \alpha_{24} + a_{sh} \cdot \text{cm2}$   
 $\beta_1 \leftarrow \beta_1, w_{sh} = w(\text{ba}, \text{sh}, q_{level})$   
 $\beta_2 \leftarrow \beta_2, h_{sh} = h(\text{ba}, \text{sh}, q_{level})$   
 $\beta_3 \leftarrow \beta_3, a_{sh} = a(\text{ba}, \text{sh}, q_{level})$   
 $\beta_6 \leftarrow \beta_6, \text{cm2} = \text{cost\_m2}(q_{level}, \text{covered})$   
 Same note as above.

**g80: Adding a lavatory bathroom as a third bathroom to the current list of spaces**

$\alpha_4 \leftarrow \alpha_4, n_{users} = \alpha_4$   
 $\alpha_6 \leftarrow \alpha_6, q_{level} = \alpha_6$   
 $\alpha_{11} \leftarrow \alpha_{11} + \text{[lavatory}, \emptyset, 0]} >$   
 $\alpha_{13} \leftarrow \alpha_{13} + \text{[ba}, \text{id}_{lv}, \emptyset, (\text{hygiene}), (\text{lavatory}, 100), (\text{isolated}, 100), (q_{level}, 0), w_{lv}, h_{lv}, a_{lv}] >$   
 $n_{bathrooms} = 2 \wedge n_{users} \in \{8, 9\}$   
 $\text{id}_{lv} = \max(\text{id}) + 1$   
 $\alpha_{14} \leftarrow \alpha_{14} + \text{sleeping}(\text{id}_{lv}, a_{lv})$   
 $\alpha_{15} \leftarrow \alpha_{15} + \text{available}, (0, -a_{lv}, 0, -a_{lv}), (0, 0, 0, 0), (0, -a_{lv}, 0, -a_{lv}), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + \text{used}, (0, a_{lv}, 0, a_{lv}), (0, 0, 0, 0), (0, a_{lv}, 0, a_{lv}), -a_i / a_u + (a_i / a_u + a_{lv}) >$   
 $\alpha_{17} \leftarrow \alpha_{17} + \text{[id}_{lv}, \text{id}_{f1}, \text{on}, 100]} >$   
 $\alpha_{24} \leftarrow \alpha_{24} + a_{lv} \cdot \text{cm2}$   
 $\beta_1 \leftarrow \beta_1, w_{lv} = w(\text{ba}, \text{lv}, q_{level})$   
 $\beta_2 \leftarrow \beta_2, h_{lv} = h(\text{ba}, \text{lv}, q_{level})$   
 $\beta_3 \leftarrow \beta_3, a_{lv} = a(\text{ba}, \text{lv}, q_{level})$   
 $\beta_6 \leftarrow \beta_6, \text{cm2} = \text{cost\_m2}(q_{level}, \text{covered})$   
 Note: If there is no bedroom on the first floor, the shower bathroom can be on the second floor

**g81: Adding a lavatory bathroom as a second bathroom to the current list of spaces**

$\alpha_4 \leftarrow \alpha_4, n_{users} = \alpha_4$   
 $\alpha_6 \leftarrow \alpha_6, q_{level} = \alpha_6$   
 $\alpha_8 \leftarrow \alpha_8, n_{floors} = \alpha_8$   
 $\alpha_{11} \leftarrow \alpha_{11} + \text{[lavatory}, \emptyset, 0]} >$   
 $\alpha_{13} \leftarrow \alpha_{13} + \text{[ba}, \text{id}_{lv}, \emptyset, (\text{hygiene}), (\text{lavatory}, 0), (\text{isolated}, 100), (q_{level}, 0), w_{lv}, h_{lv}, a_{lv}] >$   
 $n_{bathrooms} = 1 \wedge q_{level} = \max \wedge [(n_{users} \in \{2, 3, 4\} \wedge n_{floors} = 2) \vee n_{users} = 4]$   
 $\text{id}_{lv} = \max(\text{id}) + 1$   
 $\alpha_{14} \leftarrow \alpha_{14} + \text{sleeping}(\text{id}_{lv}, a_{lv})$   
 $\alpha_{15} \leftarrow \alpha_{15} + \text{available}, (0, -a_{lv}, 0, -a_{lv}), (0, 0, 0, 0), (0, -a_{lv}, 0, -a_{lv}), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + \text{used}, (0, a_{lv}, 0, a_{lv}), (0, 0, 0, 0), (0, a_{lv}, 0, a_{lv}), -a_i / a_u + (a_i / a_u + a_{lv}) >$   
 $\alpha_{17} \leftarrow \alpha_{17} + \text{[id}_{lv}, \text{id}_{f1}, \text{on}, 100]} >$   
 $\alpha_{24} \leftarrow \alpha_{24} + a_{lv} \cdot \text{cm2}$   
 $\beta_1 \leftarrow \beta_1, w_{lv} = w(\text{ba}, \text{lv}, q_{level})$   
 $\beta_2 \leftarrow \beta_2, h_{lv} = h(\text{ba}, \text{lv}, q_{level})$   
 $\beta_3 \leftarrow \beta_3, a_{lv} = a(\text{ba}, \text{lv}, q_{level})$   
 $\beta_6 \leftarrow \beta_6, \text{cm2} = \text{cost\_m2}(q_{level}, \text{covered})$

**g82: Adding a shower bathroom as a second bathroom to the current list of spaces**

$\alpha_4 \leftarrow \alpha_4, n_{users} = \alpha_4$   
 $\alpha_6 \leftarrow \alpha_6, q_{level} = \alpha_6$   
 $\alpha_{11} \leftarrow \alpha_{11} + \text{[shower}, \emptyset, 0]} >$   
 $\alpha_{13} \leftarrow \alpha_{13} + \text{[ba}, \text{id}_{sh}, \emptyset, (\text{hygiene}), (\text{shower}, 80), (\text{isolated}, 100), (q_{level}, 0), w_{sh}, h_{sh}, a_{sh}] >$   
 $n_{bathrooms} = 1 \wedge q_{level} = \max \wedge n_{users} \in \{6, 7, 8, 9\}$   
 $\text{id}_{sh} = \max(\text{id}) + 1$   
 $\alpha_{14} \leftarrow \alpha_{14} + \text{sleeping}(\text{id}_{sh}, a_{sh})$   
 $\alpha_{15} \leftarrow \alpha_{15} + \text{available}, (0, -a_{sh}, 0, -a_{sh}), (0, 0, 0, 0), (0, -a_{sh}, 0, -a_{sh}), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + \text{used}, (0, a_{sh}, 0, a_{sh}), (0, 0, 0, 0), (0, a_{sh}, 0, a_{sh}), -a_i / a_u + (a_i / a_u + a_{sh}) >$   
 $\alpha_{17} \leftarrow \alpha_{17} + \text{[id}_{sh}, \text{id}_{f1}, \text{on}, 100]} >$   
 $\alpha_{24} \leftarrow \alpha_{24} + a_{sh} \cdot \text{cm2}$   
 $\beta_1 \leftarrow \beta_1, w_{sh} = w(\text{ba}, \text{sh}, q_{level})$

$\beta_2 \leftarrow \beta_2, h_{sh} = h(ba, sh, q_{level})$   
 $\beta_3 \leftarrow \beta_3, a_{sh} = a(ba, sh, q_{level})$   
 $\beta_6 \leftarrow \beta_6, cm2 = cost\_m2(q_{level}, covered)$

**g83: Adding a lavatory bathroom as a third bathroom to the current list of spaces (programmer)**

$\alpha_4 \leftarrow \alpha_4, n_{users} = \alpha_4$   
 $\alpha_6 \leftarrow \alpha_6, q_{level} = \alpha_6$   
 $\alpha_{11} \leftarrow \alpha_{11} + < [lavatory, \emptyset, 0] >$   
 $\alpha_{13} \leftarrow \alpha_{13} + < [ba, id_v, \emptyset, (hygiene), (lavatory, 80), (isolated, 100), (q_{level}, 0), w_v, h_v, a_v] >$   
 $n_{bathrooms} = 2 \wedge q_{level} = \max \wedge n_{users} \in \{6, 7, 8, 9\}$   
 $id_v = \max(id) + 1$   
 $\alpha_{14} \leftarrow \alpha_{14} + \text{sleeping}(id_v, a_v)$   
 $\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, -a_v, 0, -a_v), (0, 0, 0, 0), (0, -a_v, 0, -a_v), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (0, a_v, 0, a_v), (0, 0, 0, 0), (0, a_v, 0, a_v), -a_i / a_u + (a_i / a_u + a_v) >$   
 $\alpha_{17} \leftarrow \alpha_{17} + < [id_v, id_{f1}, on, 100] >$   
 $\alpha_{24} \leftarrow \alpha_{24} + a_v \cdot cm2$   
 $\beta_1 \leftarrow \beta_1, w_v = w(ba, lv, q_{level})$   
 $\beta_2 \leftarrow \beta_2, h_v = h(ba, lv, q_{level})$   
 $\beta_3 \leftarrow \beta_3, a_v = a(ba, lv, q_{level})$   
 $\beta_6 \leftarrow \beta_6, cm2 = cost\_m2(q_{level}, covered)$

Note: If there is no bedroom on the first floor, the shower bathroom can be on the second floor

**Dining: formal / informal dining spaces (programmer)**

**g84: Adding an informal dining space to the minimum list of spaces**

$\alpha_{10} \leftarrow \alpha_{10} + < [ts, 2, ((informal, included, 1), (formal, included, 1))] >$

**g85: Adding an included informal dining space to the current list of spaces**

$\alpha_4 \leftarrow \alpha_4, n_{dwellers} = \alpha_4$   
 $\alpha_5 \leftarrow \alpha_5, n_{bedrooms} = n(\text{bedrooms})$   
 $\alpha_6 \leftarrow \alpha_6, q_{level} = \alpha_6$   
 $\alpha_{11} \leftarrow \alpha_{11} + < [informal\_dining, (included, delimited, isolated), 0] >$   
 $\alpha_{12} \leftarrow \alpha_{12} + < [informal\_dining, (included, delimited, isolated), 0] >$   
 $\alpha_{13} \leftarrow \alpha_{13} + < [id, id_{id}, \emptyset, (informal\ dining), (n_{dwellers}, 100), (included, 100), (q_{level}, 0), w_{id}, h_{id}, a_{id}] >$   
 $n_{dining\ informal} = 0 \wedge q_{level} \in \{\min, med\} \wedge n_{dwellers} \geq 3 \wedge n_{bedrooms} \geq 2$   
 $\alpha_{14} \leftarrow \alpha_{14} + \text{service}(id_{id}, a_{id})$   
 $\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, -a_{id}, 0, -a_{id}), (0, 0, 0, 0), (0, -a_{id}, 0, -a_{id}), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (a_{id}, a_{id}, 0, a_{id}), (0, 0, 0, 0), (0, a_{id}, 0, a_{id}), -a_i / a_u + (a_i + a_{id} / a_u + a_{id}) >$   
 $\alpha_{17} \leftarrow \alpha_{17} + < [id_{id}, id_{f1}, on, 100] >$   
 $\alpha_{24} \leftarrow \alpha_{24} + a_{id} \cdot cm2$   
 $\beta_1 \leftarrow \beta_1, w_{id} = w(id, n_{dwellers}, included, q_{level})$   
 $\beta_2 \leftarrow \beta_2, h_{id} = h(id, n_{dwellers}, included, q_{level})$   
 $\beta_3 \leftarrow \beta_3, a_{id} = a(id, n_{dwellers}, included, q_{level})$   
 $\beta_6 \leftarrow \beta_6, cm2 = cost\_m2(q_{level}, covered)$

**g86: Adding a formal dining space to the current list of spaces**

$\alpha_4 \leftarrow \alpha_4, n_{dwellers} = \alpha_4$   
 $\alpha_5 \leftarrow \alpha_5, n_{bedrooms} = n(\text{bedrooms})$   
 $\alpha_6 \leftarrow \alpha_6, q_{level} = \alpha_6$   
 $\alpha_{11} \leftarrow \alpha_{11} + < [formal\_dining, (included, delimited, isolated), 0] >$   
 $\alpha_{13} \leftarrow \alpha_{13} + < [fd, id_{fd}, \emptyset, (formal\ dining), (n_{dwellers}, 100), (included, 100), (q_{level}, 0), w_{fd}, h_{fd}, a_{fd}] >$   
 $n_{dining\ formal} = 0 \wedge q_{level} \in \{\min, med\}, \forall n_{dwellers}, n_{bedrooms}$   
 $\alpha_{14} \leftarrow \alpha_{14} + \text{living}(id_{fd}, a_{fd})$   
 $\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, -a_{fd}, 0, -a_{fd}), (0, 0, 0, 0), (0, -a_{fd}, 0, -a_{fd}), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (a_{fd}, a_{fd}, 0, a_{fd}), (0, 0, 0, 0), (0, a_{fd}, 0, a_{fd}), -a_i / a_u + (a_i + a_{fd} / a_u + a_{fd}) >$   
 $\alpha_{17} \leftarrow \alpha_{17} + < [id_{fd}, id_{f1}, on, 100] >$   
 $\alpha_{24} \leftarrow \alpha_{24} + a_{fd} \cdot cm2$   
 $\beta_1 \leftarrow \beta_1, w_{fd} = w(fd, n_{dwellers}, included, q_{level})$



$\beta_2 \leftarrow \beta_2, h_{fd} = h(\text{fd}, n_{\text{dwellers}}, \text{included}, q_{\text{level}})$   
 $\beta_3 \leftarrow \beta_3, a_{fd} = a(\text{fd}, n_{\text{dwellers}}, \text{included}, q_{\text{level}})$   
 $\beta_6 \leftarrow \beta_6, \text{cm2} = \text{cost\_m2}(q_{\text{level}}, \text{covered})$

**g87: Adding a delimited informal dining space to the current list of spaces**

$\alpha_4 \leftarrow \alpha_4, n_{\text{dwellers}} = \alpha_4$   
 $\alpha_5 \leftarrow \alpha_5, n_{\text{bedrooms}} = n(\text{bedrooms})$   
 $\alpha_6 \leftarrow \alpha_6, q_{\text{level}} = \alpha_6$   
 $\alpha_{11} \leftarrow \alpha_{11} + \langle [\text{informal\_dining}, (\text{delimited}, \text{included}, \text{isolated}), 0] \rangle$   
 $\alpha_{12} \leftarrow \alpha_{12} - \langle [\text{informal\_dining}, (\text{included}, \text{delimited}, \text{isolated}), 0] \rangle$   
 $\alpha_{13} \leftarrow \alpha_{13} + \langle [\text{id}, \text{id}_{fd}, \emptyset, (\text{informal dining}), (n_{\text{dwellers}}, 100), (\text{delimited}, 80), (q_{\text{level}}, 0), w_{fd}, h_{fd}, a_{fd}] \rangle$   
 $n_{\text{dining informal}} = 0 \wedge q_{\text{level}} = \text{max} \wedge n_{\text{users}} \geq 2 \wedge n_{\text{bedrooms}} \geq 1$   
 $\alpha_{14} \leftarrow \alpha_{14} + \text{service}(\text{id}_{fd}, a_{fd})$   
 $\alpha_{15} \leftarrow \alpha_{15} + \langle \text{available}, (0, -a_{fd}, 0, -a_{fd}), (0, 0, 0, 0), (0, -a_{fd}, 0, -a_{fd}), 0 \rangle$   
 $\alpha_{16} \leftarrow \alpha_{16} + \langle \text{used}, (a_{fd}, a_{fd}, 0, a_{fd}), (0, 0, 0, 0), (0, a_{fd}, 0, a_{fd}), -a_i / a_u + (a_i + a_{fd} / a_u + a_{fd}) \rangle$   
 $\alpha_{17} \leftarrow \alpha_{17} + \langle [\text{id}_{fd}, \text{id}_{fr}, \text{on}, 100] \rangle$   
 $\alpha_{24} \leftarrow \alpha_{24} + a_{fd} \cdot \text{cm2}$   
 $\beta_1 \leftarrow \beta_1, w_{fd} = w(\text{id}, n_{\text{dwellers}}, \text{delimited}, q_{\text{level}})$   
 $\beta_2 \leftarrow \beta_2, h_{fd} = h(\text{id}, n_{\text{dwellers}}, \text{delimited}, q_{\text{level}})$   
 $\beta_3 \leftarrow \beta_3, a_{fd} = a(\text{id}, n_{\text{dwellers}}, \text{delimited}, q_{\text{level}})$   
 $\beta_6 \leftarrow \beta_6, \text{cm2} = \text{cost\_m2}(q_{\text{level}}, \text{covered})$

**g88: Adding a delimited formal dining space to the current list of spaces**

$\alpha_4 \leftarrow \alpha_4, n_{\text{dwellers}} = \alpha_4$   
 $\alpha_5 \leftarrow \alpha_5, n_{\text{bedrooms}} = n(\text{bedrooms})$   
 $\alpha_6 \leftarrow \alpha_6, q_{\text{level}} = \alpha_6$   
 $\alpha_{11} \leftarrow \alpha_{11} + \langle [\text{formal\_dining}, (\text{delimited}, \text{included}, \text{isolated}), 0] \rangle$   
 $\alpha_{13} \leftarrow \alpha_{13} + \langle [\text{fd}, \text{id}_{fd}, \emptyset, (\text{formal dining}), (n_{\text{dwellers}}, 100), (\text{delimited}, 80), (q_{\text{level}}, 0), w_{fd}, h_{fd}, a_{fd}] \rangle$   
 $n_{\text{dining formal}} = 0 \wedge q_{\text{level}} = \text{max}, \forall n_{\text{dwellers}}, n_{\text{bedrooms}}$   
 $\alpha_{14} \leftarrow \alpha_{14} + \text{living}(\text{id}_{fd}, a_{fd})$   
 $\alpha_{15} \leftarrow \alpha_{15} + \langle \text{available}, (0, -a_{fd}, 0, -a_{fd}), (0, 0, 0, 0), (0, -a_{fd}, 0, -a_{fd}), 0 \rangle$   
 $\alpha_{16} \leftarrow \alpha_{16} + \langle \text{used}, (a_{fd}, a_{fd}, 0, a_{fd}), (0, 0, 0, 0), (0, a_{fd}, 0, a_{fd}), -a_i / a_u + (a_i + a_{fd} / a_u + a_{fd}) \rangle$   
 $\alpha_{17} \leftarrow \alpha_{17} + \langle [\text{id}_{fd}, \text{id}_{fr}, \text{on}, 100] \rangle$   
 $\alpha_{24} \leftarrow \alpha_{24} + a_{fd} \cdot \text{cm2}$   
 $\beta_1 \leftarrow \beta_1, w_{fd} = w(\text{fd}, n_{\text{dwellers}}, \text{delimited}, q_{\text{level}})$   
 $\beta_2 \leftarrow \beta_2, h_{fd} = h(\text{fd}, n_{\text{dwellers}}, \text{delimited}, q_{\text{level}})$   
 $\beta_3 \leftarrow \beta_3, a_{fd} = a(\text{fd}, n_{\text{dwellers}}, \text{delimited}, q_{\text{level}})$   
 $\beta_6 \leftarrow \beta_6, \text{cm2} = \text{cost\_m2}(q_{\text{level}}, \text{covered})$

**Studio: adding work/study/play spaces (programmer)**

**g89: Adding a single youth study space to the current list of spaces to a housetype**

$\alpha_3 \leftarrow \alpha_3, \alpha_3 = \text{type}$   
 $\alpha_6 \leftarrow \alpha_6, q_{\text{level}} = \alpha_6$   
 $\alpha_{10} \leftarrow \alpha_{10} + \langle [\text{sd}, 1, ((\text{youth}, \text{single included}, 1))] \rangle$   
 $\alpha_{11} \leftarrow \alpha_{11} + \langle [\text{single youth study}, (\text{included}, \text{delimited}, \text{isolated}), 0] \rangle$   
 $\alpha_{13} \leftarrow \alpha_{13}, \exists [\text{be}, \text{id}_{be}, \emptyset, (\text{sleeping}), (\text{single } ?w_i), (\text{isolated}, ?w_a), (?q_{\text{level}}, ?w_q), ?w, ?h, ?a] \wedge$   
 $\quad \neg \exists [\text{id}_{be}, ?id, \text{merged}] \in \alpha_{17} \mid \text{function} (?id) = \text{youth study} \Rightarrow$   
 $\quad + \langle [\text{sd}, \text{id}_{ys}, \emptyset, (\text{youth study}), (\text{single } ?w_i), (\text{included}, ?w_a), (q_{\text{level}}, ?w_q), w_{ys}, h_{ys}, a_{ys}] \rangle$   
 $\quad \text{zone} = \text{zone} (?id)$   
 $\text{zone} = \text{sleeping} \Rightarrow$   
 $\quad \alpha_{14} \leftarrow \alpha_{14} + \text{sleeping}(\text{id}_{ys}, a_{ys})$   
 $\quad \alpha_{15} \leftarrow \alpha_{15} + \langle \text{available}, (0, -a_{ys}, 0, -a_{ys}), (0, 0, 0, 0), (0, -a_{ys}, 0, -a_{ys}), 0 \rangle$   
 $\quad \alpha_{16} \leftarrow \alpha_{16} + \langle \text{used}, (a_{ys}, a_{ys}, 0, a_{ys}), (0, 0, 0, 0), (0, a_{ys}, 0, a_{ys}), -a_i / a_u + (a_i + a_{ys} / a_u + a_{ys}) \rangle$   
 $\quad \alpha_{17} \leftarrow \alpha_{17} + \langle [\text{id}_{ys}, \text{id}_{fr}, \text{on}, 100], [\text{id}_{ys}, \text{id}_{be}, \text{merged}, 80] \rangle$   
 $\text{zone} = \text{f2} \Rightarrow$   
 $\quad \alpha_{14} \leftarrow \alpha_{14} + \text{f2}(\text{id}_{ys}, a_{ys})$

$$\alpha_{15} \leftarrow \alpha_{15} + \langle \text{available}, (0, 0, 0, 0), (0, -a_{ys}, 0, -a_{ys}), (0, -a_{ys}, 0, -a_{ys}), 0 \rangle$$

$$\alpha_{16} \leftarrow \alpha_{16} + \langle \text{used}, (0, 0, 0, 0), (a_{ys}, a_{ys}, 0, a_{ys}), (0, a_{ys}, 0, a_{ys}), -a_i / a_u + (a_i + a_{ys} / a_u + a_{ys}) \rangle$$

$$\alpha_{17} \leftarrow \alpha_{17} + \langle [id_{ys}, id_{l2}, \text{on}, 100], [id_{ys}, id_{be}, \text{merged}, 80] \rangle$$

$$\alpha_{24} \leftarrow \alpha_{24} + a_{ys} \cdot \text{cm2}$$

$$\beta_1 \leftarrow \beta_1, w_{ys} = w(\text{ys}, \text{si}, \text{included}, q_{\text{level}})$$

$$\beta_2 \leftarrow \beta_2, h_{ys} = h(\text{ys}, \text{si}, \text{included}, q_{\text{level}})$$

$$\beta_3 \leftarrow \beta_3, a_{ys} = a(\text{ys}, \text{si}, \text{included}, q_{\text{level}})$$

$$\beta_6 \leftarrow \beta_6, \text{cm2} = \text{cost\_m2}(q_{\text{level}}, \text{covered})$$

**g90: Adding a double youth study space to the current list of spaces to a housetype**

$$\alpha_3 \leftarrow \alpha_3, \alpha_3 = \text{type}$$

$$\alpha_6 \leftarrow \alpha_6, q_{\text{level}} = \alpha_6$$

$$\alpha_{10} \leftarrow \alpha_{10} + \langle [\text{sd}, 1, ((\text{youth}, \text{double included}, 1))] \rangle$$

$$\alpha_{11} \leftarrow \alpha_{11} + \langle [\text{double\_youth\_study}, (\text{included}, \text{delimited}, \text{isolated}), 0] \rangle, \text{Ndouble youth study} < \text{Ndouble bedrooms}$$

$$\alpha_{13} \leftarrow \alpha_{13}, \exists [\text{be}, id_{be}, \emptyset, (\text{sleeping}), (\text{double ?w}), (\text{isolated}, ?w_a), (?q_{\text{level}}, ?w_q), ?w, ?h, ?a] \wedge$$

$$\quad \sim \exists [id_{be}, ?id, \text{merged}] \in \alpha_{17} \mid \text{function} (?id) = \text{youth study} \Rightarrow$$

$$\quad + \langle [\text{sd}, id_{ys}, \emptyset, (\text{youth study}), (\text{double ?w}), (\text{included}, ?w_a), (q_{\text{level}}, ?w_q), w_{ys}, h_{ys}, a_{ys}] \rangle$$

$$\quad \text{zone} = \text{zone} (?id)$$

zone = sleeping  $\Rightarrow$

$$\alpha_{14} \leftarrow \alpha_{14} + \text{sleeping}(id_{ys}, a_{ys})$$

$$\alpha_{15} \leftarrow \alpha_{15} + \langle \text{available}, (0, -a_{ys}, 0, -a_{ys}), (0, 0, 0, 0), (0, -a_{ys}, 0, -a_{ys}), 0 \rangle$$

$$\alpha_{16} \leftarrow \alpha_{16} + \langle \text{used}, (a_{ys}, a_{ys}, 0, a_{ys}), (0, 0, 0, 0), (0, a_{ys}, 0, a_{ys}), -a_i / a_u + (a_i + a_{ys} / a_u + a_{ys}) \rangle$$

$$\alpha_{17} \leftarrow \alpha_{17} + \langle [id_{ys}, id_{h1}, \text{on}, 100], [id_{ys}, id_{be}, \text{merged}, 80] \rangle$$

zone = f2  $\Rightarrow$

$$\alpha_{14} \leftarrow \alpha_{14} + \text{sleeping}(id_{ys}, a_{ys})$$

$$\alpha_{15} \leftarrow \alpha_{15} + \langle \text{available}, (0, 0, 0, 0), (0, -a_{ys}, 0, -a_{ys}), (0, -a_{ys}, 0, -a_{ys}), 0 \rangle$$

$$\alpha_{16} \leftarrow \alpha_{16} + \langle \text{used}, (0, 0, 0, 0), (a_{ys}, a_{ys}, 0, a_{ys}), (0, a_{ys}, 0, a_{ys}), -a_i / a_u + (a_i + a_{ys} / a_u + a_{ys}) \rangle$$

$$\alpha_{17} \leftarrow \alpha_{17} + \langle [id_{ys}, id_{l2}, \text{on}, 100], [id_{ys}, id_{be}, \text{merged}, 80] \rangle$$

$$\alpha_{24} \leftarrow \alpha_{24} + a_{ys} \cdot \text{cm2}$$

$$\beta_1 \leftarrow \beta_1, w_{ys} = w(\text{ys}, \text{do}, \text{included}, q_{\text{level}})$$

$$\beta_2 \leftarrow \beta_2, h_{ys} = h(\text{ys}, \text{do}, \text{included}, q_{\text{level}})$$

$$\beta_3 \leftarrow \beta_3, a_{ys} = a(\text{ys}, \text{do}, \text{included}, q_{\text{level}})$$

$$\beta_6 \leftarrow \beta_6, \text{cm2} = \text{cost\_m2}(q_{\text{level}}, \text{covered})$$

**g91: Adding an adult workspace to the current list of spaces to a housetype**

$$\alpha_3 \leftarrow \alpha_3, \alpha_3 = \text{type}$$

$$\alpha_6 \leftarrow \alpha_6, q_{\text{level}} = \alpha_6$$

$$\alpha_{10} \leftarrow \alpha_{10} + \langle [\text{sd}, 1, ((\text{adult}, \text{included}, 1))] \rangle$$

$$\alpha_{11} \leftarrow \alpha_{11} + \langle [\text{adult\_work}, (\text{included}, \text{delimited}, \text{isolated}), 0] \rangle$$

$$\alpha_{13} \leftarrow \alpha_{13}, \exists [\text{be}, id_{be}, \emptyset, (\text{sleeping}), (\text{couple ?w}), (\text{isolated}, ?w_a), (?q_{\text{level}}, ?w_q), ?w, ?h, ?a] \wedge$$

$$\quad \sim \exists [id_{be}, ?id, \text{merged}] \in \alpha_{17} \mid \text{function} (?id) = \text{adult work} \Rightarrow$$

$$\quad + \langle [\text{sd}, id_{aw}, \emptyset, (\text{adult work}), (\text{couple ?w}), (\text{included}, ?w_a), (q_{\text{level}}, ?w_q), w_{aw}, h_{aw}, a_{aw}] \rangle$$

$$\quad \text{zone} = \text{zone} (?id)$$

zone = sleeping  $\Rightarrow$

$$\alpha_{14} \leftarrow \alpha_{14} + \text{sleeping}(id_{aw}, a_{aw})$$

$$\alpha_{15} \leftarrow \alpha_{15} + \langle \text{available}, (0, -a_{aw}, 0, -a_{aw}), (0, 0, 0, 0), (0, -a_{aw}, 0, -a_{aw}), 0 \rangle$$

$$\alpha_{16} \leftarrow \alpha_{16} + \langle \text{used}, (a_{aw}, a_{aw}, 0, a_{aw}), (0, 0, 0, 0), (0, a_{aw}, 0, a_{aw}), -a_i / a_u + (a_i + a_{aw} / a_u + a_{aw}) \rangle$$

$$\alpha_{17} \leftarrow \alpha_{17} + \langle [id_{aw}, id_{h1}, \text{on}, 100], [id_{aw}, id_{be}, \text{merged}, 80] \rangle$$

zone = f2  $\Rightarrow$

$$\alpha_{14} \leftarrow \alpha_{14} + \text{sleeping}(id_{aw}, a_{aw})$$

$$\alpha_{15} \leftarrow \alpha_{15} + \langle \text{available}, (0, 0, 0, 0), (0, -a_{aw}, 0, -a_{aw}), (0, -a_{aw}, 0, -a_{aw}), 0 \rangle$$

$$\alpha_{16} \leftarrow \alpha_{16} + \langle \text{used}, (0, 0, 0, 0), (a_{aw}, a_{aw}, 0, a_{aw}), (0, a_{aw}, 0, a_{aw}), -a_i / a_u + (a_i + a_{aw} / a_u + a_{aw}) \rangle$$

$$\alpha_{17} \leftarrow \alpha_{17} + \langle [id_{aw}, id_{l2}, \text{on}, 100], [id_{aw}, id_{be}, \text{merged}, 80] \rangle$$

$$\alpha_{24} \leftarrow \alpha_{24} + a_{aw} \cdot \text{cm2}$$

$$\beta_1 \leftarrow \beta_1, w_{aw} = w(\text{aw}, \text{included}, q_{\text{level}})$$

$$\beta_2 \leftarrow \beta_2, h_{aw} = h(\text{aw}, \text{included}, q_{\text{level}})$$

$$\beta_3 \leftarrow \beta_3, a_{aw} = a(\text{aw}, \text{included}, q_{\text{level}})$$

$$\beta_6 \leftarrow \beta_6, \text{cm2} = \text{cost\_m2}(q_{\text{level}}, \text{covered})$$

**g92: Adding a single children play space to the current list of spaces to a customized house**

$\alpha_3 \leftarrow \alpha_3$  ,  $\alpha_3 = \text{custom}$   
 $\alpha_4 \leftarrow \alpha_4$ ,  $\text{age}(\text{name}) < 14$ ,  $\text{name} \notin \text{user}(\text{studios})$   
 $\alpha_6 \leftarrow \alpha_6$ ,  $q_{\text{level}} = \alpha_6$   
 $\alpha_{10} \leftarrow \alpha_{10} + < [\text{sd}, 1, ((\text{children}, \text{single included}, 1))] >$   
 $\alpha_{11} \leftarrow \alpha_{11} + < [\text{single\_children\_play}, (\text{included}, \text{delimited}, \text{isolated}), 0] >$   
 $\alpha_{13} \leftarrow \alpha_{13}$ ,  $\exists [\text{be}, \text{id}_{\text{be}}, (?name), (\text{sleeping}), (\text{single } ?w_i), (\text{isolated}, ?w_a), (?q_{\text{level}}, ?w_q), ?w, ?h, ?a] \wedge$   
 $\quad \neg \exists [\text{id}_{\text{be}}, ?id, \text{merged}] \in \alpha_{17} \mid \text{function} (?id) = \text{children play} \Rightarrow$   
 $\quad + < [\text{sd}, \text{id}_{\text{cp}}, (?name), (\text{children play}), (\text{single } ?w_i), (\text{included}, ?w_a), (q_{\text{level}}, ?w_q), w_{\text{cp}}, h_{\text{cp}}, a_{\text{cp}}] >$   
 $\quad \text{zone} = \text{zone} (?id)$   
 $\text{zone} = \text{sleeping} \Rightarrow$   
 $\quad \alpha_{14} \leftarrow \alpha_{14} + \text{sleeping}(\text{id}_{\text{cp}}, a_{\text{cp}})$   
 $\quad \alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, -a_{\text{cp}}, 0, -a_{\text{cp}}), (0, 0, 0, 0), (0, -a_{\text{cp}}, 0, -a_{\text{cp}}), 0 >$   
 $\quad \alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (a_{\text{cp}}, a_{\text{cp}}, 0, a_{\text{cp}}), (0, 0, 0, 0), (0, a_{\text{cp}}, 0, a_{\text{cp}}), -a_i / a_u + (a_i + a_{\text{cp}} / a_u + a_{\text{cp}}) >$   
 $\quad \alpha_{17} \leftarrow \alpha_{17} + < [\text{id}_{\text{cp}}, \text{id}_{\text{f1}}, \text{on}, 100], [\text{id}_{\text{cp}}, \text{id}_{\text{be}}, \text{merged}, 80] >$   
 $\text{zone} = \text{f2} \Rightarrow$   
 $\quad \alpha_{14} \leftarrow \alpha_{14} + \text{f2}(\text{id}_{\text{cp}}, a_{\text{cp}})$   
 $\quad \alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, 0, 0, 0), (0, -a_{\text{cp}}, 0, -a_{\text{cp}}), (0, -a_{\text{cp}}, 0, -a_{\text{cp}}), 0 >$   
 $\quad \alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (0, 0, 0, 0), (a_{\text{cp}}, a_{\text{cp}}, 0, a_{\text{cp}}), (0, a_{\text{cp}}, 0, a_{\text{cp}}), -a_i / a_u + (a_i + a_{\text{cp}} / a_u + a_{\text{cp}}) >$   
 $\quad \alpha_{17} \leftarrow \alpha_{17} + < [\text{id}_{\text{cp}}, \text{id}_{\text{f2}}, \text{on}, 100], [\text{id}_{\text{cp}}, \text{id}_{\text{be}}, \text{merged}, 80] >$   
 $\alpha_{24} \leftarrow \alpha_{24} + a_{\text{cp}} \cdot \text{cm2}$   
 $\beta_1 \leftarrow \beta_1$ ,  $w_{\text{cp}} = w(\text{cp}, \text{single}, \text{included}, q_{\text{level}})$   
 $\beta_2 \leftarrow \beta_2$ ,  $h_{\text{cp}} = h(\text{cp}, \text{single}, \text{included}, q_{\text{level}})$   
 $\beta_3 \leftarrow \beta_3$ ,  $a_{\text{cp}} = a(\text{cp}, \text{single}, \text{included}, q_{\text{level}})$   
 $\beta_6 \leftarrow \beta_6$ ,  $\text{cm2} = \text{cost\_m2}(q_{\text{level}}, \text{covered})$

**g93: Adding a single youth study space to the current list of spaces to a customized house**

$\alpha_3 \leftarrow \alpha_3$  ,  $\alpha_3 = \text{custom}$   
 $\alpha_4 \leftarrow \alpha_4$ ,  $14 \leq \text{age}(\text{name}) < 18$ ,  $\text{name} \notin \text{user}(\text{studios})$   
 $\alpha_6 \leftarrow \alpha_6$ ,  $q_{\text{level}} = \alpha_6$   
 $\alpha_{10} \leftarrow \alpha_{10} + < [\text{sd}, 1, ((\text{youth}, \text{single included}, 1))] >$   
 $\alpha_{11} \leftarrow \alpha_{11} + < [\text{single\_youth\_study}, (\text{included}, \text{delimited}, \text{isolated}), 0] >$   
 $\alpha_{13} \leftarrow \alpha_{13}$ ,  $\exists [\text{be}, \text{id}_{\text{be}}, (?name), (\text{sleeping}), (\text{single } ?w_i), (\text{isolated}, ?w_a), (?q_{\text{level}}, ?w_q), ?w, ?h, ?a] \wedge$   
 $\quad \neg \exists [\text{id}_{\text{be}}, ?id, \text{merged}] \in \alpha_{17} \mid \text{function} (?id) = \text{youth study} \Rightarrow$   
 $\quad + < [\text{sd}, \text{id}_{\text{ys}}, (?name), (\text{youth study}), (\text{single } ?w_i), (\text{included}, ?w_a), (q_{\text{level}}, ?w_q), w_{\text{ys}}, h_{\text{ys}}, a_{\text{ys}}] >$   
 $\quad \text{zone} = \text{zone} (?id)$   
 $\text{zone} = \text{sleeping} \Rightarrow$   
 $\quad \alpha_{14} \leftarrow \alpha_{14} + \text{sleeping}(\text{id}_{\text{ys}}, a_{\text{ys}})$   
 $\quad \alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, -a_{\text{ys}}, 0, -a_{\text{ys}}), (0, 0, 0, 0), (0, -a_{\text{ys}}, 0, -a_{\text{ys}}), 0 >$   
 $\quad \alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (a_{\text{ys}}, a_{\text{ys}}, 0, a_{\text{ys}}), (0, 0, 0, 0), (0, a_{\text{ys}}, 0, a_{\text{ys}}), -a_i / a_u + (a_i + a_{\text{ys}} / a_u + a_{\text{ys}}) >$   
 $\quad \alpha_{17} \leftarrow \alpha_{17} + < [\text{id}_{\text{ys}}, \text{id}_{\text{f1}}, \text{on}, 100], [\text{id}_{\text{ys}}, \text{id}_{\text{be}}, \text{merged}, 80] >$   
 $\text{zone} = \text{f2} \Rightarrow$   
 $\quad \alpha_{14} \leftarrow \alpha_{14} + \text{f2}(\text{id}_{\text{ys}}, a_{\text{ys}})$   
 $\quad \alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, 0, 0, 0), (0, -a_{\text{ys}}, 0, -a_{\text{ys}}), (0, -a_{\text{ys}}, 0, -a_{\text{ys}}), 0 >$   
 $\quad \alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (0, 0, 0, 0), (a_{\text{ys}}, a_{\text{ys}}, 0, a_{\text{ys}}), (0, a_{\text{ys}}, 0, a_{\text{ys}}), -a_i / a_u + (a_i + a_{\text{ys}} / a_u + a_{\text{ys}}) >$   
 $\quad \alpha_{17} \leftarrow \alpha_{17} + < [\text{id}_{\text{ys}}, \text{id}_{\text{f2}}, \text{on}, 100], [\text{id}_{\text{ys}}, \text{id}_{\text{be}}, \text{merged}, 80] >$   
 $\alpha_{24} \leftarrow \alpha_{24} + a_{\text{ys}} \cdot \text{cm2}$   
 $\beta_1 \leftarrow \beta_1$ ,  $w_{\text{ys}} = w(\text{ys}, \text{single}, \text{included}, q_{\text{level}})$   
 $\beta_2 \leftarrow \beta_2$ ,  $h_{\text{ys}} = h(\text{ys}, \text{single}, \text{included}, q_{\text{level}})$   
 $\beta_3 \leftarrow \beta_3$ ,  $a_{\text{ys}} = a(\text{ys}, \text{single}, \text{included}, q_{\text{level}})$   
 $\beta_6 \leftarrow \beta_6$ ,  $\text{cm2} = \text{cost\_m2}(q_{\text{level}}, \text{covered})$

**g94: Adding a double children play space to the current list of spaces to a customized house**

$\alpha_3 \leftarrow \alpha_3$  ,  $\alpha_3 = \text{custom}$   
 $\alpha_4 \leftarrow \alpha_4$ ,  $\text{age}(\max(\text{name}_1, \text{name}_2)) \leq 14$ ,  $\text{name} \notin \text{user}(\text{studios})$   
 $\alpha_6 \leftarrow \alpha_6$ ,  $q_{\text{level}} = \alpha_6$   
 $\alpha_{10} \leftarrow \alpha_{10} + < [\text{sd}, 1, ((\text{children}, \text{double included}, 1))] >$   
 $\alpha_{11} \leftarrow \alpha_{11} + < [\text{double\_children\_play}, (\text{included}, \text{delimited}, \text{isolated}), 0] >$   
 $\alpha_{13} \leftarrow \alpha_{13}$ ,  $\exists [\text{be}, \text{id}_{\text{be}}, (?name_1, ?name_2), (\text{sleeping}), (\text{single } ?w_i), (\text{isolated}, ?w_a), (?q_{\text{level}}, ?w_q), ?w, ?h, ?a] \wedge$

$$\sim \exists [\text{id}_{be}, ?\text{id}, \text{merged}] \in \alpha_{17} \mid \text{function} (? \text{id}) = \text{children play} \Rightarrow$$

$$+ < [\text{sd}, \text{id}_{aw}, (? \text{name}_1, ? \text{name}_2), (\text{children play}), (\text{single} ? \text{w}_1), (\text{included}, ? \text{w}_a), (\text{qlevel}, ? \text{w}_q), \text{waw}, \text{h}_{aw}, \text{a}_{aw}] >$$

$$\text{zone} = \text{zone} (? \text{id})$$

zone = sleeping  $\Rightarrow$

$$\alpha_{14} \leftarrow \alpha_{14} + \text{sleeping} (\text{id}_{cp}, \text{a}_{cp})$$

$$\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, -\text{a}_{cp}, 0, -\text{a}_{cp}), (0, 0, 0, 0), (0, -\text{a}_{cp}, 0, -\text{a}_{cp}), 0 >$$

$$\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (\text{a}_{cp}, \text{a}_{cp}, 0, \text{a}_{cp}), (0, 0, 0, 0), (0, \text{a}_{cp}, 0, \text{a}_{cp}), -\text{a}_i / \text{a}_u + (\text{a}_i + \text{a}_{cp} / \text{a}_u + \text{a}_{cp}) >$$

$$\alpha_{17} \leftarrow \alpha_{17} + < [\text{id}_{cp}, \text{id}_{f1}, \text{on}, 100], [\text{id}_{cp}, \text{id}_{be}, \text{merged}, 80] >$$

zone = f2  $\Rightarrow$

$$\alpha_{14} \leftarrow \alpha_{14} + \text{f2} (\text{id}_{cp}, \text{a}_{cp})$$

$$\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, 0, 0, 0), (0, -\text{a}_{cp}, 0, -\text{a}_{cp}), (0, -\text{a}_{cp}, 0, -\text{a}_{cp}), 0 >$$

$$\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (0, 0, 0, 0), (\text{a}_{cp}, \text{a}_{cp}, 0, \text{a}_{cp}), (0, \text{a}_{cp}, 0, \text{a}_{cp}), -\text{a}_i / \text{a}_u + (\text{a}_i + \text{a}_{cp} / \text{a}_u + \text{a}_{cp}) >$$

$$\alpha_{17} \leftarrow \alpha_{17} + < [\text{id}_{cp}, \text{id}_{f2}, \text{on}, 100], [\text{id}_{cp}, \text{id}_{be}, \text{merged}, 80] >$$

$$\alpha_{24} \leftarrow \alpha_{24} + \text{a}_{cp} \cdot \text{cm2}$$

$$\beta_1 \leftarrow \beta_1, \text{w}_{cp} = \text{w} (\text{cp}, \text{double}, \text{included}, \text{qlevel})$$

$$\beta_2 \leftarrow \beta_2, \text{h}_{cp} = \text{h} (\text{cp}, \text{double}, \text{included}, \text{qlevel})$$

$$\beta_3 \leftarrow \beta_3, \text{a}_{cp} = \text{a} (\text{cp}, \text{double}, \text{included}, \text{qlevel})$$

$$\beta_6 \leftarrow \beta_6, \text{cm2} = \text{cost\_m2} (\text{qlevel}, \text{covered})$$

**g95: Adding a double youht study space to the current list of spaces to a customized house**

$$\alpha_3 \leftarrow \alpha_3, \alpha_3 = \text{custom}$$

$$\alpha_4 \leftarrow \alpha_4, 14 \leq \text{age}(\max(\text{name}_1, \text{name}_2)) < 18, \text{name} \notin \text{user}(\text{studios})$$

$$\alpha_6 \leftarrow \alpha_6$$

$$\alpha_{10} \leftarrow \alpha_{10} + < [\text{sd}, 1, ((\text{youth}, \text{double}, \text{included}, 1))] >$$

$$\alpha_{11} \leftarrow \alpha_{11} + < [\text{double\_youth\_study}, (\text{included}, \text{delimited}, \text{isolated}), 0] >$$

$$\alpha_{13} \leftarrow \alpha_{13}, \exists [\text{be}, \text{id}_{be}, (? \text{name}_1, ? \text{name}_2), (\text{sleeping}), (\text{double} ? \text{w}_1), (\text{isolated}, ? \text{w}_a), (? \text{qlevel}, ? \text{w}_q), ? \text{w}, ? \text{h}, ? \text{a}] \wedge$$

$$\sim \exists [\text{id}_{be}, ? \text{id}, \text{merged}] \in \alpha_{17} \mid \text{function} (? \text{id}) = \text{youth study} \Rightarrow$$

$$+ < [\text{sd}, \text{id}_{ys}, (? \text{name}_1, ? \text{name}_2), (\text{youth study}), (\text{double} ? \text{w}_1), (\text{included}, ? \text{w}_a), (\text{qlevel}, ? \text{w}_q), \text{w}_{ys}, \text{h}_{ys}, \text{a}_{ys}] >$$

$$\text{zone} = \text{zone} (? \text{id})$$

zone = sleeping  $\Rightarrow$

$$\alpha_{14} \leftarrow \alpha_{14} + \text{sleeping} (\text{id}_{ys}, \text{a}_{ys})$$

$$\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, -\text{a}_{ys}, 0, -\text{a}_{ys}), (0, 0, 0, 0), (0, -\text{a}_{ys}, 0, -\text{a}_{ys}), 0 >$$

$$\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (\text{a}_{ys}, \text{a}_{ys}, 0, \text{a}_{ys}), (0, 0, 0, 0), (0, \text{a}_{ys}, 0, \text{a}_{ys}), -\text{a}_i / \text{a}_u + (\text{a}_i + \text{a}_{ys} / \text{a}_u + \text{a}_{ys}) >$$

$$\alpha_{17} \leftarrow \alpha_{17} + < [\text{id}_{ys}, \text{id}_{f1}, \text{on}, 100], [\text{id}_{ys}, \text{id}_{be}, \text{merged}, 80] >$$

zone = f2  $\Rightarrow$

$$\alpha_{14} \leftarrow \alpha_{14} + \text{f2} (\text{id}_{ys}, \text{a}_{ys})$$

$$\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, 0, 0, 0), (0, -\text{a}_{ys}, 0, -\text{a}_{ys}), (0, -\text{a}_{ys}, 0, -\text{a}_{ys}), 0 >$$

$$\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (0, 0, 0, 0), (\text{a}_{ys}, \text{a}_{ys}, 0, \text{a}_{ys}), (0, \text{a}_{ys}, 0, \text{a}_{ys}), -\text{a}_i / \text{a}_u + (\text{a}_i + \text{a}_{ys} / \text{a}_u + \text{a}_{ys}) >$$

$$\alpha_{17} \leftarrow \alpha_{17} + < [\text{id}_{ys}, \text{id}_{f2}, \text{on}, 100], [\text{id}_{ys}, \text{id}_{be}, \text{merged}, 80] >$$

$$\alpha_{24} \leftarrow \alpha_{24} + \text{a}_{ys} \cdot \text{cm2}$$

$$\beta_1 \leftarrow \beta_1, \text{w}_{ys} = \text{w} (\text{ys}, \text{double}, \text{included}, \text{qlevel})$$

$$\beta_2 \leftarrow \beta_2, \text{h}_{ys} = \text{h} (\text{ys}, \text{double}, \text{included}, \text{qlevel})$$

$$\beta_3 \leftarrow \beta_3, \text{a}_{ys} = \text{a} (\text{ys}, \text{double}, \text{included}, \text{qlevel})$$

$$\beta_6 \leftarrow \beta_6, \text{cm2} = \text{cost\_m2} (\text{qlevel}, \text{covered})$$

**g96: Adding a double adult workspace to the current list of spaces to a customized house**

$$\alpha_3 \leftarrow \alpha_3, \alpha_3 = \text{custom}$$

$$\alpha_4 \leftarrow \alpha_4, \text{age}(\max(\text{name}_1, \text{name}_2)) \geq 18, \text{name} \notin \text{user}(\text{studios})$$

$$\alpha_6 \leftarrow \alpha_6$$

$$\alpha_{10} \leftarrow \alpha_{10} + < [\text{sd}, 1, ((\text{adult}, \text{included}, 1))] >$$

$$\alpha_{11} \leftarrow \alpha_{11} + < [\text{adult\_work}, (\text{included}, \text{delimited}, \text{isolated}), 0] >$$

$$\alpha_{13} \leftarrow \alpha_{13}, \exists [\text{be}, \text{id}_{be}, (? \text{name}_1, ? \text{name}_2), (\text{sleeping}) (\text{couple} ? \text{w}_1), (\text{isolated}, ? \text{w}_a), (? \text{qlevel}, ? \text{w}_q), ? \text{w}, ? \text{h}, ? \text{a}] \wedge$$

$$\sim \exists [\text{id}_{be}, ? \text{id}, \text{merged}] \in \alpha_{17} \mid \text{function} (? \text{id}) = \text{adult work} \Rightarrow$$

$$+ < [\text{sd}, \text{id}_{aw}, (? \text{name}_1, ? \text{name}_2), (\text{adult work}), (\text{double}, 0), (\text{included}, ? \text{w}_a), (\text{qlevel}, ? \text{w}_q), \text{waw}, \text{h}_{aw}, \text{a}_{aw}] >$$

$$\text{zone} = \text{zone} (? \text{id})$$

zone = sleeping  $\Rightarrow$

$$\alpha_{14} \leftarrow \alpha_{14} + \text{sleeping} (\text{id}_{aw}, \text{a}_{aw})$$

$$\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, -\text{a}_{aw}, 0, -\text{a}_{aw}), (0, 0, 0, 0), (0, -\text{a}_{aw}, 0, -\text{a}_{aw}), 0 >$$

$$\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (\text{a}_{aw}, \text{a}_{aw}, 0, \text{a}_{aw}), (0, 0, 0, 0), (0, \text{a}_{aw}, 0, \text{a}_{aw}), -\text{a}_i / \text{a}_u + (\text{a}_i + \text{a}_{aw} / \text{a}_u + \text{a}_{aw}) >$$

$$\alpha_{17} \leftarrow \alpha_{17} + \langle [\text{id}_{aw}, \text{id}_{f1}, \text{on}, 100], [\text{id}_{aw}, \text{id}_{be}, \text{merged}, 80] \rangle$$

zone = f2  $\Rightarrow$

$$\alpha_{14} \leftarrow \alpha_{14} + \text{sleeping}(\text{id}_{aw}, a_{aw})$$

$$\alpha_{15} \leftarrow \alpha_{15} + \langle \text{available}, (0, 0, 0, 0), (0, -a_{aw}, 0, -a_{aw}), (0, -a_{aw}, 0, -a_{aw}), 0 \rangle$$

$$\alpha_{16} \leftarrow \alpha_{16} + \langle \text{used}, (0, 0, 0, 0), (a_{aw}, a_{aw}, 0, a_{aw}), (0, a_{aw}, 0, a_{aw}), -a_i / a_u + (a_i + a_{aw} / a_u + a_{aw}) \rangle$$

$$\alpha_{17} \leftarrow \alpha_{17} + \langle [\text{id}_{aw}, \text{id}_{f2}, \text{on}, 100], [\text{id}_{aw}, \text{id}_{be}, \text{merged}, 80] \rangle$$

$$\alpha_{24} \leftarrow \alpha_{24} + a_{aw} \cdot \text{cm2}$$

$$\beta_1 \leftarrow \beta_1, w_{aw} = w(\text{aw}, \text{included}, q_{\text{level}})$$

$$\beta_2 \leftarrow \beta_2, h_{aw} = h(\text{aw}, \text{included}, q_{\text{level}})$$

$$\beta_3 \leftarrow \beta_3, a_{aw} = a(\text{aw}, \text{included}, q_{\text{level}})$$

$$\beta_6 \leftarrow \beta_6, \text{cm2} = \text{cost\_m2}(q_{\text{level}}, \text{covered})$$

## Laundry (programmer)

### g97: Adding an included laundry space

$$\alpha_4 \leftarrow \alpha_4, n_{\text{users}} = \alpha_4$$

$$\alpha_6 \leftarrow \alpha_6, q_{\text{level}} = \alpha_6$$

$$\alpha_{10} \leftarrow \alpha_{10} + \langle [\text{la}, 1, \text{included}] \rangle$$

$$\alpha_{11} \leftarrow \alpha_{11} + \langle [\text{laundry}, (\text{included}, \text{delimited}, \text{isolated}), 0] \rangle$$

$$\alpha_{13} \leftarrow \alpha_{13}, \exists [k_i, ?\text{id}_{k_i}, \emptyset, (\text{cooking}) (\text{isolated } ?w_i), (?q_{\text{level}}, ?w_q), ?w, ?h, ?a]$$

$$+ \langle [\text{la}, \text{id}_{\text{la}}, \emptyset, (\text{wash and dry cloths}), (\text{included}, 100), (q_{\text{level}}, 0), w_{\text{la}}, h_{\text{la}}, a_{\text{la}}] \rangle$$

$$n_{\text{la}} = 0 \wedge q_{\text{level}} = \min$$

$$\alpha_{14} \leftarrow \alpha_{14} + \text{service}(\text{id}_{\text{la}}, a_{\text{la}})$$

$$\alpha_{15} \leftarrow \alpha_{15} + \langle \text{available}, (0, -a_{\text{la}}, 0, -a_{\text{la}}), (0, 0, 0, 0), (0, -a_{\text{la}}, 0, -a_{\text{la}}), 0 \rangle$$

$$\alpha_{16} \leftarrow \alpha_{16} + \langle \text{used}, (a_{\text{la}}, a_{\text{la}}, 0, a_{\text{la}}), (0, 0, 0, 0), (0, a_{\text{la}}, 0, a_{\text{la}}), -a_i / a_u + (a_i + a_{\text{la}} / a_u + a_{\text{la}}) \rangle$$

$$\alpha_{17} \leftarrow \alpha_{17} + \langle [\text{id}_{\text{la}}, \text{id}_{f1}, \text{on}, 100] \rangle$$

$$\alpha_{24} \leftarrow \alpha_{24} + a_{\text{la}} \cdot \text{cm2}$$

$$\beta_1 \leftarrow \beta_1, w_{\text{la}} = w(\text{la}, n_{\text{users}}, \text{included}, q_{\text{level}})$$

$$\beta_2 \leftarrow \beta_2, h_{\text{la}} = h(\text{la}, n_{\text{users}}, \text{included}, q_{\text{level}})$$

$$\beta_3 \leftarrow \beta_3, a_{\text{la}} = a(\text{la}, n_{\text{users}}, \text{included}, q_{\text{level}})$$

$$\beta_6 \leftarrow \beta_6, \text{cm2} = \text{cost\_m2}(q_{\text{level}}, \text{covered})$$

### g98: Adding a delimited laundry space

$$\alpha_4 \leftarrow \alpha_4, n_{\text{users}} = \alpha_4$$

$$\alpha_6 \leftarrow \alpha_6, q_{\text{level}} = \alpha_6$$

$$\alpha_{11} \leftarrow \alpha_{11} + \langle [\text{laundry}, (\text{included}, \text{delimited}, \text{isolated}), 0] \rangle$$

$$\alpha_{13} \leftarrow \alpha_{13}, \exists [k_i, ?\text{id}_{k_i}, \emptyset, (\text{cooking}) (\text{isolated } ?w_i), (?q_{\text{level}}, ?w_q), ?w, ?h, ?a]$$

$$+ \langle [\text{la}, \text{id}_{\text{la}}, \emptyset, (\text{wash and dry cloths}), (\text{delimited}, 80), (q_{\text{level}}, 0), w_{\text{la}}, h_{\text{la}}, a_{\text{la}}] \rangle$$

$$n_{\text{la}} = 0 \wedge q_{\text{level}} \in \{\text{med}, \text{max}\}$$

$$\alpha_{14} \leftarrow \alpha_{14} + \text{service}(\text{id}_{\text{la}}, a_{\text{la}})$$

$$\alpha_{15} \leftarrow \alpha_{15} + \langle \text{available}, (0, -a_{\text{la}}, 0, -a_{\text{la}}), (0, 0, 0, 0), (0, -a_{\text{la}}, 0, -a_{\text{la}}), 0 \rangle$$

$$\alpha_{16} \leftarrow \alpha_{16} + \langle \text{used}, (0, a_{\text{la}}, 0, a_{\text{la}}), (0, 0, 0, 0), (0, a_{\text{la}}, 0, a_{\text{la}}), -a_i / a_u + (a_i / a_u + a_{\text{la}}) \rangle$$

$$\alpha_{17} \leftarrow \alpha_{17} + \langle [\text{id}_{\text{la}}, \text{id}_{f1}, \text{on}, 100] \rangle$$

$$\alpha_{24} \leftarrow \alpha_{24} + a_{\text{la}} \cdot \text{cm2}$$

$$\beta_1 \leftarrow \beta_1, w_{\text{la}} = w(\text{la}, n_{\text{users}}, \text{included}, q_{\text{level}})$$

$$\beta_2 \leftarrow \beta_2, h_{\text{la}} = h(\text{la}, n_{\text{users}}, \text{included}, q_{\text{level}})$$

$$\beta_3 \leftarrow \beta_3, a_{\text{la}} = a(\text{la}, n_{\text{users}}, \text{included}, q_{\text{level}})$$

$$\beta_6 \leftarrow \beta_6, \text{cm2} = \text{cost\_m2}(q_{\text{level}}, \text{covered})$$

## Storage (programmer)

### g99: Adding an included pantry space

$$\alpha_4 \leftarrow \alpha_4, n_{\text{users}} = \alpha_4$$

$$\alpha_6 \leftarrow \alpha_6, q_{\text{level}} = \alpha_6$$

$$\alpha_{10} \leftarrow \alpha_{10} + \langle [\text{cl}, 1, (\text{pantry}, 1, \text{included})] \rangle$$

$$\alpha_{11} \leftarrow \alpha_{11} + \langle [\text{pantry}, (\text{included}, \text{isolated}), 0] \rangle$$

$$\alpha_{13} \leftarrow \alpha_{13} + \langle [pa, id_{pa}, \emptyset, (\text{food storage}), (\text{included}, 100), (q_{level}, 0), w_{pa}, h_{pa}, a_{pa}] \rangle$$

$$n_{pa} = 0$$

$$\alpha_{14} \leftarrow \alpha_{14} + \text{service}(id_{pa}, a_{pa})$$

$$\alpha_{15} \leftarrow \alpha_{15} + \langle \text{available}, (0, -a_{pa}, 0, -a_{pa}), (0, 0, 0, 0), (0, -a_{pa}, 0, -a_{pa}), 0 \rangle$$

$$\alpha_{16} \leftarrow \alpha_{16} + \langle \text{used}, (a_{pa}, a_{pa}, 0, a_{pa}), (0, 0, 0, 0), (a_{pa}, a_{pa}, 0, a_{pa}), -a_i / a_u + (a_i + a_{pa} / a_u + a_{pa}) \rangle$$

$$\alpha_{17} \leftarrow \alpha_{17} + \langle [id_{pa}, id_{f1}, \text{on}, 100] \rangle$$

$$\alpha_{24} \leftarrow \alpha_{24} + a_{pa} \cdot cm2$$

$$\beta_1 \leftarrow \beta_1, w_{pa} = w(pa, n_{users}, \text{included}, q_{level})$$

$$\beta_2 \leftarrow \beta_2, h_{pa} = h(pa, n_{users}, \text{included}, q_{level})$$

$$\beta_3 \leftarrow \beta_3, a_{pa} = a(pa, n_{users}, \text{included}, q_{level})$$

$$\beta_6 \leftarrow \beta_6, cm2 = \text{cost\_m2}(q_{level}, \text{covered})$$

#### g100: Adding a house clothing storage space (programmer)

$$\alpha_4 \leftarrow \alpha_4, n_{users} = \alpha_4$$

$$\alpha_6 \leftarrow \alpha_6, q_{level} = \alpha_6$$

$$\alpha_{10} \leftarrow \alpha_{10} + [cl, 1, (\text{clothing}, 1, \text{included})]$$

$$\alpha_{11} \leftarrow \alpha_{11} + \langle [\text{house\_clothing\_storage}, (\text{included}, \text{isolated}), 0] \rangle$$

$$\alpha_{13} \leftarrow \alpha_{13} + \langle [cl, id_{hc}, \emptyset, (\text{house clothing storage}), (\text{included}, 100), (q_{level}, 0), w_{hc}, h_{hc}, a_{hc}] \rangle$$

$$n_{\text{house clothing}} = 0$$

$$\alpha_{14} \leftarrow \alpha_{14} + \text{service}(id_{hc}, a_{hc})$$

$$\alpha_{15} \leftarrow \alpha_{15} + \langle \text{available}, (0, -a_{hc}, 0, -a_{hc}), (0, 0, 0, 0), (0, -a_{hc}, 0, -a_{hc}), 0 \rangle$$

$$\alpha_{16} \leftarrow \alpha_{16} + \langle \text{used}, (0, a_{hc}, 0, a_{hc}), (0, 0, 0, 0), (0, a_{hc}, 0, a_{hc}), -a_i / a_u + (a_i / a_u + a_{hc}) \rangle$$

$$\alpha_{17} \leftarrow \alpha_{17} + \langle [id_{hc}, id_{f1}, \text{on}, 100] \rangle$$

$$\alpha_{24} \leftarrow \alpha_{24} + a_{hc} \cdot cm2$$

$$\beta_1 \leftarrow \beta_1, w_{hc} = w(hc, n_{users}, \text{included}, q_{level})$$

$$\beta_2 \leftarrow \beta_2, h_{hc} = h(hc, n_{users}, \text{included}, q_{level})$$

$$\beta_3 \leftarrow \beta_3, a_{hc} = a(hc, n_{users}, \text{included}, q_{level})$$

$$\beta_6 \leftarrow \beta_6, cm2 = \text{cost\_m2}(q_{level}, \text{covered})$$

#### g101: Adding a global storage space (programmer)

$$\alpha_4 \leftarrow \alpha_4, n_{users} = \alpha_4$$

$$\alpha_6 \leftarrow \alpha_6, q_{level} = \alpha_6$$

$$\alpha_{10} \leftarrow \alpha_{10} + \langle [cl, 1, (\text{global}, 1, \text{included})] \rangle$$

$$\alpha_{11} \leftarrow \alpha_{11} + \langle [\text{global\_storage}, (\text{included}, \text{isolated}), 0] \rangle$$

$$\alpha_{13} \leftarrow \alpha_{13} + \langle [cl, id_{cl}, \emptyset, (\text{global storage}), (\text{included}, 100), (q_{level}, 0), w_{cl}, h_{cl}, a_{cl}] \rangle$$

$$n_{\text{global storage}} = 0$$

$$\alpha_{14} \leftarrow \alpha_{14} + \text{service}(id_{cl}, a_{cl})$$

$$\alpha_{15} \leftarrow \alpha_{15} + \langle \text{available}, (0, -a_{cl}, 0, -a_{cl}), (0, 0, 0, 0), (0, -a_{cl}, 0, -a_{cl}), 0 \rangle$$

$$\alpha_{16} \leftarrow \alpha_{16} + \langle \text{used}, (0, a_{cl}, 0, a_{cl}), (0, 0, 0, 0), (0, a_{cl}, 0, a_{cl}), -a_i / a_u + (a_i / a_u + a_{cl}) \rangle$$

$$\alpha_{17} \leftarrow \alpha_{17} + \langle [id_{cl}, id_{f1}, \text{on}, 100] \rangle$$

$$\alpha_{24} \leftarrow \alpha_{24} + a_{cl} \cdot cm2$$

$$\beta_1 \leftarrow \beta_1, w_{cl} = w(cl, n_{users}, \text{included}, q_{level})$$

$$\beta_2 \leftarrow \beta_2, h_{cl} = h(cl, n_{users}, \text{included}, q_{level})$$

$$\beta_3 \leftarrow \beta_3, a_{cl} = a(cl, n_{users}, \text{included}, q_{level})$$

$$\beta_6 \leftarrow \beta_6, cm2 = \text{cost\_m2}(q_{level}, \text{covered})$$

### Exterior spaces (programmer)

#### g102: Adding a balcony (programmer)

$$\alpha_4 \leftarrow \alpha_4, n_{dwellers} = \alpha_6$$

$$\alpha_6 \leftarrow \alpha_6, q_{level} = \alpha_6$$

$$\alpha_9 \leftarrow \alpha_9, \alpha_9 = \text{true}$$

$$\alpha_{11} \leftarrow \alpha_{11} + \langle [bl, (\text{isolated}), 0] \rangle$$

$$\alpha_{13} \leftarrow \alpha_{13}$$

$$+ \langle [bl, id_{bl}, \emptyset, (\text{being outside}), (n_{dwellers}, 100), (\text{isolated}, 100), (q_{level}, w_q), w_{bl}, h_{bl}, a_{bl}] \rangle$$

$$n_{\text{balcony}} = 0$$

$$\alpha_{14} \leftarrow \alpha_{14} + f2(id_{bl}, -a_0)$$

$$\alpha_{15} \leftarrow \alpha_{15} + \langle \text{available}, (0, 0, 0, 0), (0, 0, -a_{bl}, -a_{bl}), (0, 0, -a_{bl}, -a_{bl}), 0 \rangle$$

$\alpha_{16} \leftarrow \alpha_{16} + \langle \text{used}, (0, 0, 0, 0), (0, 0, a_{bl}, a_{bl}), (0, 0, a_{bl}, a_{bl}), -a_i / a_u + (a_i / a_u + a_{bl}) \rangle$   
 $\alpha_{17} \leftarrow \alpha_{17} + \langle [id_{bl}, id_{bl}, on, 100] \rangle$   
 $\alpha_{24} \leftarrow \alpha_{24} + a_{ba} \cdot cm2$   
 $\beta_1 \leftarrow \beta_1, w_{bl} = w(bl, n_{dwellers}, isolated, q_{level})$   
 $\beta_2 \leftarrow \beta_2, h_{bl} = h(bl, n_{dwellers}, isolated, q_{level})$   
 $\beta_3 \leftarrow \beta_3, a_{bl} = a(bl, n_{dwellers}, isolated, q_{level})$   
 $\beta_6 \leftarrow \beta_6, cm2 = cost\_m2(q_{level}, covered)$

## Optional Dwelling capacity

### g103: Initialize list of optional spaces (programmer)

$\alpha_{12} \leftarrow \alpha_{12} + \langle$   
     [*bathroom* (*bathtub, shower, lavatory*), 0],  
     [*informal\_dining*, (*included, delimited, isolated*), 0],  
     [*adult\_studio*, (*included, delimited, isolated*), 0]  
     [*clothing\_storage*, (*included, delimited, isolated*), 0],  
     [*global\_storage*, (*included, delimited, isolated*), 0],  
     [*balcony*, (*isolated*), 0], [*terrace*, (*isolated*), 0]  $\rangle$

## Bathrooms (user)

### g104: Up/downgrading a bathroom (user)

$\alpha_{10} \leftarrow \alpha_{10}, n_{obligatory} = n(\text{capacity})$   
 $\alpha_{13} \leftarrow \alpha_{13}$   
 $- \langle [ba, ?id_{ba}, \emptyset, (hygiene), (?capacity, w_c), (isolated, 100), (?q_{level}, w_q), ?w_o, ?h_o, ?a_o] \rangle$   
 $+ \langle [ba, ?id_{ba}, \emptyset, (hygiene), (capacity, w_c), (isolated, 100), (q_{level}, w_q), w_n, h_n, a_n] \rangle$   
 $n_{current} = n(\text{capacity})$   
 $n_{current} \geq n_{obligatory}$   
 $capacity \in \{bathtub, shower, lavatory\}$   
 $q_{level} \in \{min, med, max\}$   
 $w_c, w_q \in \{0, 5, 10, \dots, 100\}$   
 $n_{current} = n_{obligatory} \Rightarrow w_c = 100,$   
 $n_{current} > n_{obligatory} \Rightarrow w_c < 80$

$fn = f1 \Rightarrow$

$\alpha_{14} \leftarrow \alpha_{14} + \text{sleeping}(id_{ba}, -?a_o + a_n)$   
 $\alpha_{15} \leftarrow \alpha_{15} + \langle \text{available}, (0, ?a_o - a_n, 0, ?a_o - a_n), (0, 0, 0, 0), (0, ?a_o - a_n, 0, ?a_o - a_n), 0 \rangle$   
 $\alpha_{16} \leftarrow \alpha_{16} + \langle \text{used}, (0, -?a_o + a_n, 0, -?a_o + a_n),$   
     (0, 0, 0, 0),  
     (0, -?a\_o + a\_n, 0, -?a\_o + a\_n),  
      $-a_i / a_u + (a_i / a_u - ?a_o + a_n) \rangle$

$fn = f2 \Rightarrow$

$\alpha_{14} \leftarrow \alpha_{14} + f2(id_{ba}, -?a_o + a_n)$   
 $\alpha_{15} \leftarrow \alpha_{15} + \langle \text{available}, (0, 0, 0, 0), (0, ?a_o - a_n, 0, ?a_o - a_n), (0, ?a_o - a_n, 0, ?a_o - a_n), 0 \rangle$   
 $\alpha_{16} \leftarrow \alpha_{16} + \langle \text{used}, (0, 0, 0, 0),$   
     (0, -?a\_o + a\_n, 0, -?a\_o + a\_n),  
     (0, -?a\_o + a\_n, 0, -?a\_o + a\_n),  
      $-a_i / a_u + (a_i / a_u - ?a_o + a_n) \rangle$

$\alpha_{17} \leftarrow \alpha_{17}, \exists \langle [id_{ba}, ?id_{fn}, on, ?w_i] \rangle$

$\alpha_{22} \leftarrow \alpha_{22} + (?a_o - a_n) \cdot cm2$   
 $\beta_1 \leftarrow \beta_1, w_n = w(ba, capacity, isolated, q_{level})$   
 $\beta_2 \leftarrow \beta_2, h_n = h(ba, capacity, isolated, q_{level})$   
 $\beta_3 \leftarrow \beta_3, a_n = a(ba, capacity, isolated, q_{level})$   
 $\beta_6 \leftarrow \beta_6, cm2 = cost\_m2(q_{level}, covered)$

### g105: Adding a bathroom, if the number of bathrooms does not exceed the maximum allowed (user)

$\alpha_5 \leftarrow \alpha_5, n_{bedrooms} = \alpha_5$   
 $\alpha_{13} \leftarrow \alpha_{13} + \langle [ba, id_{ba}, \emptyset, (hygiene), (capacity, w_c), (isolated, 100), (q_{level}, w_q), w_n, h_n, a_n] \rangle$   
 $id_{ba} = id(\text{max}) + 1$

$n_{\text{bathrooms}} < n_{\text{bedrooms}} + 2$   
 $\text{new\_type} \in \{\text{bathtub, shower, lavatory}\}$   
 $\text{new\_qlevel} \in \{\text{min, med, max}\}$   
 $w_c, w_q \in \{0, 5, 10, \dots, 80\}$   
 $\text{capacity} = \text{lavatory} \wedge \text{first floor available area} \geq a_n \Rightarrow$   
 $\alpha_{14} \leftarrow \alpha_{14} + \text{sleeping}(\text{id}_{ba}, + a_n)$   
 $\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, - a_n, 0, - a_n), (0, 0, 0, 0), (0, - a_n, 0, - a_n), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (0, a_n, 0, a_n), (0, 0, 0, 0), (0, a_n, 0, a_n), - a_i / a_u + (a_i / a_u + a_n) >$   
 $\alpha_{17} \leftarrow \alpha_{17}, \exists < [\text{id}_{ba}, \text{id}_{r1}, \text{on}, 80] >$   
 $\text{capacity} \neq \text{lavatory} \vee (\text{capacity} = \text{lavatory} \wedge \text{first floor available area} < a_n) \Rightarrow$   
 $\alpha_{14} \leftarrow \alpha_{14} + f2(\text{id}_{ba}, + a_n)$   
 $\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, 0, 0, 0), (0, - a_n, 0, - a_n), (0, - a_n, 0, - a_n), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (0, 0, 0, 0), (0, a_n, 0, a_n), (0, a_n, 0, a_n), - a_i / a_u + (a_i / a_u + a_n) >$   
 $\alpha_{17} \leftarrow \alpha_{17}, \exists < [\text{id}_{ba}, \text{id}_{r2}, \text{on}, 80] >$   
 $\alpha_{22} \leftarrow \alpha_{22} + a_{nt} \cdot \text{cm}2$   
 $\beta_1 \leftarrow \beta_1, w_n = w(\text{ba}, \text{capacity}, \text{isolated}, \text{qlevel})$   
 $\beta_2 \leftarrow \beta_2, h_n = h(\text{ba}, \text{capacity}, \text{isolated}, \text{qlevel})$   
 $\beta_3 \leftarrow \beta_3, a_n = a(\text{ba}, \text{capacity}, \text{isolated}, \text{qlevel})$   
 $\beta_6 \leftarrow \beta_6, \text{cm}2 = \text{cost\_m}2(\text{qlevel}, \text{covered})$

**g106: Deleting a bathroom, if there is more than the obligatory number of bathrooms (user)**

$\alpha_{10} \leftarrow \alpha_{10}, n_{\text{obligatory}} = n(\text{new\_type})$   
 $\alpha_{13} \leftarrow \alpha_{13}$   
 $- < [\text{ba}, \text{id}_{ba}, (?users), \emptyset, (\text{hygiene}), (?capacity, ?w_c), (\text{isolated}, 100), (?qlevel, ?w_q), ?w_o, ?h_o, ?a_o] >$   
 $n_{\text{current}} = n(\text{capacity})$   
 $n_{\text{current}} > n_{\text{obligatory}}$   
 $\alpha_{14} \leftarrow \alpha_{14} \exists [\text{zone}, ?rooms, ?area], \text{id}_{ba} \in ?rooms$   
 $- \text{space}(\text{zone}, \text{id}_{ba}, - ?a_o)$   
 $\text{zone} \neq f2 \Rightarrow$   
 $\alpha_{14} \leftarrow \alpha_{14} - \text{zone}(\text{id}_{ba}, - ?a_o)$   
 $\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, ?a_o, 0, ?a_o), (0, 0, 0, 0), (0, ?a_o, 0, ?a_o), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (0, -?a_o, 0, -?a_o), (0, 0, 0, 0), (0, -?a_o, 0, -?a_o), - a_i / a_u + (a_i / a_u - ?a_o) >$   
 $\text{zone} = f2 \Rightarrow$   
 $\alpha_{14} \leftarrow \alpha_{14} - \text{zone}(\text{id}_{ba}, - ?a_o)$   
 $\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, 0, 0, 0), (0, ?a_o, 0, ?a_o), (0, ?a_o, 0, ?a_o), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (0, 0, 0, 0), (0, -?a_o, 0, -?a_o), (0, -?a_o, 0, -?a_o), - a_i / a_u + (a_i / a_u - ?a_o) >$   
 $\alpha_{17} \leftarrow \alpha_{17} - < [\text{id}_{ba}, ?id, ?r, ?w] >, \forall ?id, ?r, ?w$   
 $\alpha_{22} \leftarrow \alpha_{22} - ?a_o \cdot \text{cm}2$   
 $\beta_6 \leftarrow \beta_6, \text{cm}2 = \text{cost\_m}2(\text{qlevel}, \text{covered})$

**g107: Deleting the bathroom from the optional list when its number equals the maximum allowed**

$\alpha_5 \leftarrow \alpha_5, n_{\text{bedrooms}} = \alpha_5$   
 $\alpha_{12} \leftarrow \alpha_{12} - < [\text{bathroom}(\text{bathtub}, \text{shower}, \text{lavatory}), 0] >$   
 $\alpha_{13} \leftarrow \alpha_{13}$   
 $n_{\text{bathrooms}} = n_{\text{bedrooms}} + 2$

**g108: Adding the bathroom to the optional list when its number is below the maximum allowed**

$\alpha_5 \leftarrow \alpha_5, n_{\text{bedrooms}} = \alpha_5$   
 $\alpha_{12} \leftarrow \alpha_{12} - < [\text{bathroom}(\text{bathtub}, \text{shower}, \text{lavatory}), 0] >$   
 $\alpha_{13} \leftarrow \alpha_{13}$   
 $n_{\text{bathrooms}} = n_{\text{bedrooms}} + 2$

**Dining (user)**

**g109: Up/downgrading a formal dining space (user)**

$\alpha_4 \leftarrow \alpha_4, n_{\text{users}} = \alpha_4$   
 $\alpha_{10} \leftarrow \alpha_{10}, \text{obligatory\_articulation} = \text{articulation}(\text{formal dining})$



$\alpha_{13} \leftarrow \alpha_{13}$   
 $- < [ts, id_{fd}, \emptyset, (\text{formal dining}), (?capacity, w_c), (?articulation, ?w_a), (?q_{level}, ?w_q), ?w_o, ?h_o, ?a_o] >$   
 $+ < [ts, id_{fd}, \emptyset, (\text{formal dining}), (capacity, w_c), (articulation, w_a), (q_{level}, w_q), w_n, h_n, a_n] >$   
 $capacity_{id} = capacity (\text{informal dining}), n_{users} \leq capacity + capacity_{id} \leq 2 \cdot n_{users}$   
 $articulation \in \{\text{included, delimited, isolated}\}, articulation \geq obligatory\_articulation,$   
 $w_c, w_a, w_q \in \{0, 5, 10, \dots, 100\}$   
 $capacity > n_{users} \Rightarrow w_c < 80$   
 $articulation > obligatory\_articulation \Rightarrow w_a < 80$   
 $q_{level} > \min \Rightarrow w_q < 80$   
 $\alpha_{14} \leftarrow \alpha_{14} \exists [zone, ?rooms, ?area], id_{fd} \in ?rooms$   
 $- space (zone, ?id_{op}, -?a_o)$   
 $articulation = \text{included} \Rightarrow + \text{living} (id_{fd}, a_n)$   
 $articulation \in \{\text{delimited, isolated}\} \Rightarrow + \text{service} (id_{fd}, a_{fd})$   
 $\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, ?a_o - a_n, 0, ?a_o - a_n), (0, 0, 0, 0), (0, ?a_o - a_n, 0, ?a_o - a_n), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (-?a_o + a_n, -?a_o + a_n, 0, -?a_o + a_n),$   
 $(0, 0, 0, 0),$   
 $(0, -?a_o + a_n, 0, -?a_o + a_n),$   
 $- a_i / a_u + (a_i - ?a_o + a_n / a_u - ?a_o + a_n) >$   
 $\alpha_{22} \leftarrow \alpha_{22} + (a_n - ?a_o) \cdot cm2$   
 $\beta_1 \leftarrow \beta_1, w_n = w (fd, capacity, articulation, q_{level})$   
 $\beta_2 \leftarrow \beta_2, h_n = h (fd, capacity, articulation, q_{level})$   
 $\beta_3 \leftarrow \beta_3, a_n = a (fd, capacity, articulation, q_{level})$   
 $\beta_6 \leftarrow \beta_6, cm2 = cost\_m2 (q_{level}, covered)$

Note: It is not necessary to check the floor of the dining space because it is always on the first floor. In fact, the dining space needs to be close to the kitchen, which in turn, needs to be on the same floor as the bedroom that is required to be on the entrance floor, which in the case of the Malagueira houses is on first floor.

#### g110: Up/downgrading an informal dining space (user)

$\alpha_4 \leftarrow \alpha_4, n_{users} = \alpha_4$   
 $\alpha_{10} \leftarrow \alpha_{10}, obligatory\_type = \text{type} (\text{informal dining})$   
 $\alpha_{13} \leftarrow \alpha_{13}$   
 $- < [ts, id_{fd}, \emptyset, (\text{informal dining}), (?capacity, w_c), (?articulation, ?w_a), (?q_{level}, ?w_q), ?w_o, ?h_o, ?a_o] >$   
 $+ < [ts, id_{fd}, \emptyset, (\text{informal dining}), (capacity, w_c), (articulation, w_a), (q_{level}, w_q), w_n, h_n, a_n] >$   
 $capacity_{id} = capacity (\text{formal dining}), n_{users} \leq capacity + capacity_{id} \leq 2 \cdot n_{users}$   
 $articulation \in \{\text{included, delimited, isolated}\}, articulation \geq obligatory\_articulation,$   
 $w_c, w_a, w_q \in \{0, 5, 10, \dots, 100\}$   
 $capacity > n_{users} \Rightarrow w_c < 80$   
 $articulation > obligatory\_articulation \Rightarrow w_a < 80$   
 $q_{level} > \min \Rightarrow w_q < 80$   
 $\alpha_{14} \leftarrow \alpha_{14}, + \text{service} (id_{fd}, -?a_o + a_n)$   
 $\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, ?a_o - a_n, 0, ?a_o - a_n), (0, 0, 0, 0), (0, ?a_o - a_n, 0, ?a_o - a_n), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (-?a_o + a_n, -?a_o + a_n, 0, -?a_o + a_n),$   
 $(0, 0, 0, 0),$   
 $(0, -?a_o + a_n, 0, -?a_o + a_n),$   
 $- a_i / a_u + (a_i - ?a_o + a_n / a_u - ?a_o + a_n) >$   
 $\alpha_{22} \leftarrow \alpha_{22} + (a_n - ?a_o) \cdot cm2$   
 $\beta_1 \leftarrow \beta_1, w_{fd} = w (id, capacity, articulation, q_{level})$   
 $\beta_2 \leftarrow \beta_2, h_{fd} = h (id, capacity, articulation, q_{level})$   
 $\beta_3 \leftarrow \beta_3, a_{fd} = a (id, capacity, articulation, q_{level})$   
 $\beta_6 \leftarrow \beta_6, cm2 = cost\_m2 (q_{level}, covered)$

#### g111: Adding an informal dining space, is there is none (user)

$\alpha_{12} \leftarrow \alpha_{12} - < [\text{informal\_dining}, (\text{included, delimited, isolated}), 0] >$   
 $\alpha_{13} \leftarrow \alpha_{13}$   
 $+ < [ts, id_{fd}, \emptyset, (\text{informal dining}), (capacity, w_c), (articulation, w_a), (q_{level}, w_q), w_n, h_n, a_n] >$   
 $id_{fd} = id (\text{max}) + 1$   
 $n_{\text{informal dining}} = 0$   
 $capacity_{id} = capacity (\text{formal dining}), n_{users} \leq capacity + capacity_{id} \leq 2 \cdot n_{users}$

articulation  $\in$  {included, delimited, isolated}, articulation  $\geq$  obligatory\_articulation,  
 $w_c, w_a, w_q \in \{0, 5, 10, \dots, 100\}$   
capacity  $> n_{users} \Rightarrow w_c < 80$   
articulation  $>$  obligatory\_articulation  $\Rightarrow w_a < 80$   
 $q_{level} > \min \Rightarrow w_q < 80$

$\alpha_{14} \leftarrow \alpha_{14} + \text{service}(\text{id}_{id}, a_n)$   
 $\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, -a_n, 0, -a_n), (0, 0, 0, 0), (0, -a_n, 0, -a_n), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (a_n, a_n, 0, a_n), (0, 0, 0, 0), (0, a_n, 0, a_n), -a_i / a_u + (a_i + a_n / a_u + a_n) >$   
 $\alpha_{17} \leftarrow \alpha_{17} + < [\text{id}_{id}, \text{id}_{fi}, \text{on}, 100] >$   
 $\alpha_{22} \leftarrow \alpha_{22} + a_n \cdot \text{cm2}$   
 $\beta_1 \leftarrow \beta_1, w_{id} = w(\text{id}, \text{capacity}, \text{articulation}, q_{level})$   
 $\beta_2 \leftarrow \beta_2, h_{id} = h(\text{id}, \text{capacity}, \text{articulation}, q_{level})$   
 $\beta_3 \leftarrow \beta_3, a_{id} = a(\text{id}, \text{capacity}, \text{articulation}, q_{level})$   
 $\beta_6 \leftarrow \beta_6, \text{cm2} = \text{cost\_m2}(q_{level}, \text{covered})$

#### g112: Deleting an informal dining space, if exists an isolated formal dining space (user)

$\alpha_{12} \leftarrow \alpha_{12} + < [\text{informal\_dining}, (\text{included}, \text{delimited}, \text{isolated}), 0] >$   
 $\alpha_{13} \leftarrow \alpha_{13}$   
-  $< [\text{id}, ?\text{id}_{id}, (?users), (\text{informal dining}), (?capacity, ?w_c), (?articulation, ?w_a), (?q_{level}, ?w_q), ?w_o, ?h_o, ?a_o] >$   
articulation (formal dining) = isolated  
 $\alpha_{14} \leftarrow \alpha_{14} - \text{service}(\text{id}_{id}, -?a_o)$   
 $\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, ?a_o, 0, ?a_o), (0, 0, 0, 0), (0, ?a_o, 0, ?a_o), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (-?a_o, -?a_o, 0, -?a_o), (0, 0, 0, 0), (0, -?a_o, 0, -?a_o), -a_i / a_u + (a_i - ?a_o / a_u - ?a_o) >$   
 $\alpha_{17} \leftarrow \alpha_{17} - < [\text{id}_{id}, ?\text{id}, ?r, ?w] >, \forall ?\text{id}, ?r, ?w$   
 $\alpha_{22} \leftarrow \alpha_{22} + ?a_o \cdot \text{cm2}$   
 $\beta_6 \leftarrow \beta_6, \text{cm2} = \text{cost\_m2}(?q_{level}, \text{covered})$

### Studio spaces (user)

#### g113: Up/downgrading a child play space (user)

$\alpha_{10} \leftarrow \alpha_{10}, n_{obligatory} = n(\text{child play}), \text{capacity}_{obligatory} = \sum_{i=1}^{n_{obligatory}} \text{capacity}(\text{child play})$   
 $\alpha_{13} \leftarrow \alpha_{13}$   
-  $< [\text{sd}, \text{id}_{cp}, (?users), (\text{child play}), (?capacity, ?w_c), (?articulation, ?w_a), (?q_{level}, ?w_q), ?w_o, ?h_o, ?a_o] >$   
+  $< [\text{sd}, \text{id}_{cp}, (?users), (\text{child play}), (\text{capacity}, w_c), (\text{articulation}, w_a), (q_{level}, 0), w_n, h_n, a_n] >$   
 $n_{current} = n(\text{child play}), \text{capacity}_{current} = \sum_{i=1}^{n_{current}} \text{capacity}(\text{child play})$   
 $\text{capacity}_{obligatory} \leq \text{capacity} \leq \text{capacity}_{total} + 2$   
articulation  $\in$  {included, delimited, isolated}  
 $q_{level} \in \{\min, \text{med}, \text{max}\}$   
 $w_c, w_a, w_q \in \{0, 5, 10, \dots, 100\}$   
capacity  $\geq \text{capacity}_{obligatory} \Rightarrow w_c < 80$   
articulation  $>$  included  $\Rightarrow w_a < 80$   
 $q_{level} > \min \Rightarrow w_q < 80$   
 $\alpha_{14} \leftarrow \alpha_{14} \exists [\text{zone}, ?rooms, ?area], \text{id}_{cp} \in ?rooms$   
- space (zone,  $\text{id}_{cp}$ , - ?a\_o)  
zone  $\neq$  f2  $\Rightarrow$   
 $\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, ?a_o - a_n, 0, ?a_o - a_n), (0, 0, 0, 0), (0, ?a_o - a_n, 0, ?a_o - a_n), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (-?a_o + a_n, -?a_o + a_n, 0, -?a_o + a_n),$   
(0, 0, 0, 0),  
(-?a\_o + a\_n, -?a\_o + a\_n, 0, -?a\_o + a\_n),  
-  $a_i / a_u + (a_i - ?a_o + a_n / a_u - ?a_o + a_n) >$   
zone = f2  $\Rightarrow$   
 $\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, 0, 0, 0), (0, ?a_o - a_n, 0, ?a_o - a_n), (0, ?a_o - a_n, 0, ?a_o - a_n), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (0, 0, 0, 0),$   
(-?a\_o + a\_n, -?a\_o + a\_n, 0, -?a\_o + a\_n),  
(-?a\_o + a\_n, -?a\_o + a\_n, 0, -?a\_o + a\_n),  
-  $a_i / a_u + (a_i - ?a_o + a_n / a_u - ?a_o + a_n) >$   
 $\alpha_{22} \leftarrow \alpha_{22} + (a_n - a_o) \cdot \text{cm2}$   
 $\beta_1 \leftarrow \beta_1, w_n = w(\text{cp}, \text{capacity}, \text{articulation}, q_{level})$

$\beta_2 \leftarrow \beta_2, h_n = h(\text{cp, capacity, articulation, } q_{\text{level}})$   
 $\beta_3 \leftarrow \beta_3, a_n = a(\text{cp, capacity, articulation, } q_{\text{level}})$   
 $\beta_6 \leftarrow \beta_6, \text{cm2} = \text{cost\_m2}(q_{\text{level}}, \text{covered})$

**g114: Up/downgrading a youth study space (user)**

$\alpha_{10} \leftarrow \alpha_{10}, n_{\text{obligatory}} = n(\text{youth study}), \text{capacity}_{\text{obligatory}} = \sum_{i=1}^{n_{\text{obligatory}}} \text{capacity}(\text{youth study}_i)$

$\alpha_{13} \leftarrow \alpha_{13}$

$- < [\text{sd, id}_{\text{ys}}, (?users), (\text{youth study}), (?capacity, ?w_c), (?articulation, ?w_a), (?q_{\text{level}}, ?w_q), ?w_o, ?h_o, ?a_o] >$   
 $+ < [\text{sd, id}_{\text{ys}}, (?users), (\text{youth study}), (\text{capacity}, w_c), (\text{articulation}, w_a), (q_{\text{level}}, 0), w_n, h_n, a_n] >$

$n_{\text{current}} = n(\text{youth study}), \text{capacity}_{\text{current}} = \sum_{i=1}^{n_{\text{current}}} \text{capacity}(\text{youth study}_i)$

$\text{capacity}_{\text{total}} \leq \text{capacity} \leq \text{capacity}_{\text{total}} + 2$

$\text{articulation} \in \{\text{included, delimited, isolated}\}$

$q_{\text{level}} \in \{\text{min, med, max}\}$

$w_c, w_a, w_q \in \{0, 5, 10, \dots, 100\}$

$\text{capacity} \geq \text{capacity}_{\text{obligatory}} \Rightarrow w_c < 80$

$\text{articulation} > \text{included} \Rightarrow w_a < 80$

$q_{\text{level}} > \text{min} \Rightarrow w_q < 80$

$\alpha_{14} \leftarrow \alpha_{14} \exists [\text{zone, ?rooms, ?area}], \text{id}_{\text{cp}} \in ?rooms$

- space (zone, id<sub>ys</sub>, - ?a<sub>o</sub>)

zone ≠ f2 ⇒

$\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, ?a_o - a_n, 0, ?a_o - a_n), (0, 0, 0, 0), (0, ?a_o - a_n, 0, ?a_o - a_n), 0 >$

$\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (-?a_o + a_n, -?a_o + a_n, 0, -?a_o + a_n),$

$(0, 0, 0, 0),$

$(-?a_o + a_n, -?a_o + a_n, 0, -?a_o + a_n),$

$- a_i / a_u + (a_i - ?a_o + a_n / a_u - ?a_o + a_n) >$

zone = f2 ⇒

$\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, 0, 0, 0), (0, ?a_o - a_n, 0, ?a_o - a_n), (0, ?a_o - a_n, 0, ?a_o - a_n), 0 >$

$\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (0, 0, 0, 0),$

$(-?a_o + a_n, -?a_o + a_n, 0, -?a_o + a_n),$

$(-?a_o + a_n, -?a_o + a_n, 0, -?a_o + a_n),$

$- a_i / a_u + (a_i - ?a_o + a_n / a_u - ?a_o + a_n) >$

$\alpha_{22} \leftarrow \alpha_{22} + (a_n - a_o) \cdot \text{cm2}$

$\beta_1 \leftarrow \beta_1, w_n = w(\text{ys, capacity, articulation, } q_{\text{level}})$

$\beta_2 \leftarrow \beta_2, h_n = h(\text{ys, capacity, articulation, } q_{\text{level}})$

$\beta_3 \leftarrow \beta_3, a_n = a(\text{ys, capacity, articulation, } q_{\text{level}})$

$\beta_6 \leftarrow \beta_6, \text{cm2} = \text{cost\_m2}(q_{\text{level}}, \text{covered})$

**g115: Up/downgrading an adult workspace (user)**

$\alpha_4 \leftarrow \alpha_4, n_{\text{adult dwellers}} = \alpha_4$

$\alpha_{10} \leftarrow \alpha_{10}, n_{\text{obligatory}} = n(\text{adult work}), \text{capacity}_{\text{obligatory}} = \sum_{i=1}^{n_{\text{obligatory}}} \text{capacity}(\text{adult work}_i)$

$\alpha_{13} \leftarrow \alpha_{13}$

$- < [\text{sd, id}_{\text{aw}}, (?users), (\text{adult work}), (?capacity, ?w_c), (?articulation, ?w_a), (?q_{\text{level}}, ?w_q), ?w_o, ?h_o, ?a_o] >$   
 $+ < [\text{sd, id}_{\text{aw}}, (?users), (\text{adult work}), (\text{capacity}, w_c), (\text{articulation}, w_a), (q_{\text{level}}, 0), w_n, h_n, a_n] >$

$n_{\text{current}} = n(\text{adult work}), \text{capacity}_{\text{current}} = \sum_{i=1}^{n_{\text{current}}} \text{capacity}(\text{adult work}_i)$

$\text{capacity}_{\text{obligatory}} \leq \text{capacity}_{\text{current}} \leq n_{\text{adult dwellers}} + 2$

$\text{capacity} \in \{1, 2\}$

$\text{articulation} \in \{\text{included, delimited, isolated}\}$

$q_{\text{level}} \in \{\text{min, med, max}\}$

$w_c, w_a, w_q \in \{0, 5, 10, \dots, 100\}$

$\text{capacity}_{\text{current}} \geq \text{capacity}_{\text{obligatory}} \Rightarrow w_c < 80$

$\text{articulation} > \text{included} \Rightarrow w_a < 80$

$q_{\text{level}} > \text{min} \Rightarrow w_q < 80$

$\alpha_{14} \leftarrow \alpha_{14} \exists [\text{zone, ?rooms, ?area}], \text{id}_{\text{cp}} \in ?rooms$

- space (zone, id<sub>aw</sub>, - ?a<sub>o</sub>)

$\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, ?a_o - a_n, 0, ?a_o - a_n), (0, 0, 0, 0), (0, ?a_o - a_n, 0, ?a_o - a_n), 0 >$

$\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (-?a_o + a_n, -?a_o + a_n, 0, -?a_o + a_n),$

$(0, 0, 0, 0),$

$(-?a_o + a_n, -?a_o + a_n, 0, -?a_o + a_n),$

$- a_i / a_u + (a_i - ?a_o + a_n / a_u - ?a_o + a_n) >$

$\alpha_{22} \leftarrow \alpha_{22} + (a_n - a_o) \cdot cm2$   
 $\beta_1 \leftarrow \beta_1, w_n = w$  (aw, capacity, articulation, ql<sub>level</sub>)  
 $\beta_2 \leftarrow \beta_2, h_n = h$  (aw, capacity, articulation, ql<sub>level</sub>)  
 $\beta_3 \leftarrow \beta_3, a_n = a$  (aw, capacity, articulation, ql<sub>level</sub>)  
 $\beta_6 \leftarrow \beta_6, cm2 = cost\_m2$  (ql<sub>level</sub>, covered)

**g116: Adding an adult workspace (user)**

$\alpha_4 \leftarrow \alpha_4, n_{adult\ dwellers} = \alpha_4$   
 $\alpha_{10} \leftarrow \alpha_{10}, n_{obligatory} = n$  (adult work),  $capacity_{obligatory} = \sum_{i=1}^{n_{obligatory}} capacity$  (adult work i)  
 $\alpha_{13} \leftarrow \alpha_{13}$   
 $+ < [sd, id_{aw}, (?users), (adult\ work), (capacity, w_c), (articulation, w_a), (q_{level}, 0), w_n, h_n, a_n] >$   
 $n_{current} = n$  (adult work),  $capacity_{current} = \sum_{i=1}^{n_{current}} capacity$  (adult work i)  
 $capacity_{obligatory} \leq capacity_{current} \leq n_{adult\ dwellers} + 2$   
 $capacity \in \{1, 2\}$   
 $articulation \in \{included, delimited, isolated\}$   
 $q_{level} \in \{min, med, max\}$   
 $w_c, w_a, w_q \in \{0, 5, 10, \dots, 100\}$   
 $capacity_{current} \geq capacity_{obligatory} \Rightarrow w_c < 80$   
 $articulation > included \Rightarrow w_a < 80$   
 $q_{level} > min \Rightarrow w_q < 80$   
 $first\ floor\ available\ area \geq a_n \Rightarrow$   
 $\alpha_{14} \leftarrow \alpha_{14} + sleeping$  (id<sub>aw</sub>, + a<sub>n</sub>)  
 $\alpha_{15} \leftarrow \alpha_{15} + < available, (-a_n, -a_n, 0, -a_n), (0, 0, 0, 0), (-a_n, -a_n, 0, -a_n), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + < used, (a_n, a_n, 0, a_n), (0, 0, 0, 0), (a_n, a_n, 0, a_n), -a_i / a_u + (a_i + a_n / a_u + a_n) >$   
 $\alpha_{17} \leftarrow \alpha_{17}, \exists < [id_{aw}, id_{f1}, on, 80] >$   
 $first\ floor\ available\ area < a_n \Rightarrow$   
 $\alpha_{14} \leftarrow \alpha_{14} + f2$  (id<sub>aw</sub>, + a<sub>n</sub>)  
 $\alpha_{15} \leftarrow \alpha_{15} + < available, (0, 0, 0, 0), (-a_n, -a_n, 0, -a_n), (-a_n, -a_n, 0, -a_n), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + < used, (0, 0, 0, 0), (a_n, a_n, 0, a_n), (a_n, a_n, 0, a_n), -a_i / a_u + (a_i + a_n / a_u + a_n) >$   
 $\alpha_{17} \leftarrow \alpha_{17}, \exists < [id_{aw}, id_{f2}, on, 80] >$   
 $\alpha_{22} \leftarrow \alpha_{22} + a_n \cdot cm2$   
 $\beta_1 \leftarrow \beta_1, w_n = w$  (aw, capacity, articulation, ql<sub>level</sub>)  
 $\beta_2 \leftarrow \beta_2, h_n = h$  (aw, capacity, articulation, ql<sub>level</sub>)  
 $\beta_3 \leftarrow \beta_3, a_n = a$  (aw, capacity, articulation, ql<sub>level</sub>)  
 $\beta_6 \leftarrow \beta_6, cm2 = cost\_m2$  (ql<sub>level</sub>, covered)

**g117: Deleting an adult workspace**

$\alpha_{10} \leftarrow \alpha_{10}, n_{obligatory} = n$  (adult work),  $capacity_{total} = \sum_{i=1}^{n_{obligatory}} capacity$  (adult work i)  
 $\alpha_{13} \leftarrow \alpha_{13}$   
 $- < [sd, id_{aw}, (?users), (adult\ work), (?capacity, ?w_c), (?articulation, ?w_a), (?q_{level}, ?w_q), ?w_o, ?h_o, ?a_o] >$   
 $n_{current} = n$  (adult work),  $capacity_{current} = \sum_{i=1}^{n_{current}} capacity$  (adult work i)  
 $capacity_{total} \leq capacity_{current}$   
 $\alpha_{14} \leftarrow \alpha_{14} \exists [zone, ?rooms, ?area], id_{aw} \in ?rooms$   
 $- space$  (zone, id<sub>aw</sub>, - ?a<sub>o</sub>)  
 $zone \neq 2 \Rightarrow$   
 $\alpha_{14} \leftarrow \alpha_{14} + sleeping$  (id<sub>aw</sub>, - ?a<sub>o</sub>)  
 $\alpha_{15} \leftarrow \alpha_{15} + < available, (0, ?a_o, 0, ?a_o), (0, 0, 0, 0), (0, ?a_o, 0, ?a_o), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + < used, (-?a_o, -?a_o, 0, ?a_o), (0, 0, 0, 0), (-?a_o, -?a_o, 0, -?a_o), -a_i / a_u + (a_i - ?a_o / a_u - ?a_o) >$   
 $zone = 2 \Rightarrow$   
 $\alpha_{14} \leftarrow \alpha_{14} + f2$  (id<sub>aw</sub>, - ?a<sub>o</sub>)  
 $\alpha_{15} \leftarrow \alpha_{15} + < available, (0, 0, 0, 0), (0, a_n, 0, a_n), (0, a_n, 0, a_n), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + < used, (0, 0, 0, 0), (-a_n, -a_n, 0, -a_n), (-a_n, -a_n, 0, -a_n), -a_i / a_u + (a_i - a_n / a_u - a_n) >$   
 $\alpha_{17} \leftarrow \alpha_{17} - < [id_{aw}, ?id, ?r, ?w] >, \forall ?id, ?r, ?w$   
 $\alpha_{22} \leftarrow \alpha_{22} - ?a_n \cdot cm2$   
 $\beta_6 \leftarrow \beta_6, cm2 = cost\_m2$  (ql<sub>level</sub>, covered)

**g118: Deleting the adult workspace from the optional list when its number equals the maximum (programmer)**

$\alpha_4 \leftarrow \alpha_4, n_{adult\ dwellers} = \alpha_4$   
 $\alpha_{12} \leftarrow \alpha_{12} - < [adult\_work, (included, delimited, isolated), 0] >$

$$\alpha_{13} \leftarrow \alpha_{13}, n_{\text{current}} = n \text{ (adult work)}, \text{capacity}_{\text{current}} = \sum_{i=1}^{n_{\text{current}}} \text{capacity (adult work}_i\text{)}$$

$$\text{capacity}_{\text{current}} = n_{\text{adult dwellers}} + 2$$

**g119: Adding the adult workspace to the optional list when its number is below the maximum (programmer)**

$$\alpha_4 \leftarrow \alpha_4, n_{\text{adult dwellers}} = \alpha_4$$

$$\alpha_{10} \leftarrow \alpha_{10}, n_{\text{obligatory}} = n \text{ (adult work)}, \text{capacity}_{\text{obligatory}} = \sum_{i=1}^{n_{\text{obligatory}}} \text{capacity (adult work}_i\text{)}$$

$$\alpha_{12} \leftarrow \alpha_{12} + \langle [\text{adult\_work}, (\text{included}, \text{delimited}, \text{isolated}), 0] \rangle$$

$$\alpha_{13} \leftarrow \alpha_{13}, n_{\text{current}} = n \text{ (adult work)}, \text{capacity}_{\text{current}} = \sum_{i=1}^{n_{\text{current}}} \text{capacity (adult work}_i\text{)}$$

$$\text{capacity}_{\text{current}} < n_{\text{adult dwellers}} + 2$$

## Laundry (user)

**g120: up/downgrading the laundry (user)**

## Storage (user)

**g121: up/downgrading the pantry (user)**

$$\alpha_{11} \leftarrow \alpha_{11} - \langle [\text{pantry}, (\text{included}, \text{isolated}), 0] \rangle$$

$$+ \langle [\text{pantry}, (\text{isolated}, \text{included}), 0] \rangle$$

$$\alpha_{13} \leftarrow \alpha_{13}$$

$$- \langle [\text{pa}, \text{id}_{\text{pa}}, (?users), (\text{food storage}), (?capacity ?w_c), (?articulation, ?w_a), (?q_{\text{level}}, ?w_q), ?w_o, ?h_o, ?a_o] \rangle$$

$$+ \langle [\text{pa}, \text{id}_{\text{pa}}, (?users), (\text{food storage}), (?capacity ?w_c), (\text{articulation}, w_a), (q_{\text{level}}, w_q), w_n, h_n, a_n] \rangle$$

$$\text{articulation} \in \{\text{included}, \text{isolated}\}$$

$$q_{\text{level}} \in \{\text{min}, \text{med}, \text{max}\}$$

$$w_a, w_q \in \{0, 5, 10, \dots, 100\}$$

$$\text{articulation} > \text{included} \Rightarrow w_a < 80$$

$$q_{\text{level}} > \text{min} \Rightarrow w_q < 80$$

$$\alpha_{14} \leftarrow \alpha_{14} + \text{service} (\text{zone}, \text{id}_{\text{aw}}, a_n - ?a_o)$$

$$\alpha_{15} \leftarrow \alpha_{15} + \langle \text{available}, (0, ?a_o - a_n, 0, ?a_o - a_n), (0, 0, 0, 0), (0, ?a_o - a_n, 0, ?a_o - a_n), 0 \rangle$$

$$?articulation \neq \text{articulation} \wedge ?articulation = \text{included} \Rightarrow$$

$$\alpha_{16} \leftarrow \alpha_{16} + \langle \text{used}, (-?a_o, -?a_o + a_n, 0, -?a_o + a_n),$$

$$(0, 0, 0, 0),$$

$$(-?a_o, -?a_o + a_n, 0, -?a_o + a_n),$$

$$- a_i / a_u + (a_i - ?a_o / a_u - ?a_o + a_n) \rangle$$

$$?articulation \neq \text{articulation} \wedge ?articulation = \text{isolated} \Rightarrow$$

$$\alpha_{16} \leftarrow \alpha_{16} + \langle \text{used}, (a_n, -?a_o + a_n, 0, -?a_o + a_n),$$

$$(0, 0, 0, 0),$$

$$(a_n, -?a_o + a_n, 0, -?a_o + a_n),$$

$$- a_i / a_u + (a_i + a_n / a_u - ?a_o + a_n) \rangle$$

$$\alpha_{22} \leftarrow \alpha_{22} + (?a_o - a_n) \cdot \text{cm2}$$

$$\beta_1 \leftarrow \beta_1, w_n = w (\text{pa}, ?capacity, \text{articulation}, q_{\text{level}})$$

$$\beta_2 \leftarrow \beta_2, h_n = h (\text{pa}, ?capacity, \text{articulation}, q_{\text{level}})$$

$$\beta_3 \leftarrow \beta_3, a_n = a (\text{pa}, ?capacity, \text{articulation}, q_{\text{level}})$$

$$\beta_6 \leftarrow \beta_6, \text{cm2} = \text{cost\_m2} (q_{\text{level}}, \text{covered})$$

**g122: Up/downgrading the house clothing storage space (user)**

$$\alpha_4 \leftarrow \alpha_4, n_{\text{users}} = \alpha_4$$

$$\alpha_{10} \leftarrow \alpha_{10}, n_{\text{obligatory}} = n \text{ (house clothing storage)}, \text{capacity}_{\text{total}} = \sum_{i=1}^{n_{\text{obligatory}}} \text{capacity (house clothing storage)}$$

$$\alpha_{13} \leftarrow \alpha_{13} - \langle [\text{hc}, \text{id}_{\text{hc}}, (?users), (\text{house clothing storage}), (?capacity ?w_c), (?articulation, ?w_a), (?q_{\text{level}}, ?w_q), ?w_o, ?h_o,$$

$$?a_o] \rangle$$

$$+ \langle [\text{hc}, \text{id}_{\text{hc}}, (?users), (\text{house clothing storage}), (?capacity ?w_c), (\text{articulation}, w_a), (q_{\text{level}}, w_q), w_n, h_n, a_n] \rangle$$

$$n_{\text{current}} = n \text{ (global storage)}, \text{capacity}_{\text{current}} = \sum_{i=1}^{n_{\text{current}}} \text{capacity (global storage)}$$

$$\text{capacity}_{\text{obligatory}} < \text{capacity}_{\text{current}} < n_{\text{users}} + 2$$

$$\text{articulation} \in \{\text{included}, \text{isolated}\}$$

$$q_{\text{level}} \in \{\text{min}, \text{med}, \text{max}\}$$

$$w_c, w_a, w_q \in \{0, 5, 10, \dots, 100\}$$

$$\begin{aligned} & \text{capacity}_{\text{current}} > \text{capacity}_{\text{obligatory}} \Rightarrow W_c < 80 \\ & \text{articulation} > \text{included} \Rightarrow W_a < 80 \\ & q_{\text{level}} > \text{min} \Rightarrow W_q < 80 \\ \alpha_{14} \leftarrow & \alpha_{14} \exists [\text{zone}, ?\text{rooms}, ?\text{area}], \text{id}_{\text{hc}} \in ?\text{rooms} \\ & + \text{space}(\text{zone}, \text{id}_{\text{hc}}, a_n - ?a_o) \\ \alpha_{15} \leftarrow & \alpha_{15} + < \text{available}, (0, ?a_o - a_n, 0, ?a_o - a_n), (0, 0, 0, 0), (0, ?a_o - a_n, 0, ?a_o - a_n), 0 > \\ ?\text{articulation} \neq & \text{articulation} \wedge ?\text{articulation} = \text{included} \Rightarrow \\ \alpha_{16} \leftarrow & \alpha_{16} + < \text{used}, (-?a_o, -?a_o + a_n, 0, -?a_o + a_n), \\ & (0, 0, 0, 0), \\ & (-?a_o, -?a_o + a_n, 0, -?a_o + a_n), \\ & - a_i / a_u + (a_i - ?a_o / a_u - ?a_o + a_n) > \\ ?\text{articulation} \neq & \text{articulation} \wedge ?\text{articulation} = \text{isolated} \Rightarrow \\ \alpha_{16} \leftarrow & \alpha_{16} + < \text{used}, (a_n, -?a_o + a_n, 0, -?a_o + a_n), \\ & (0, 0, 0, 0), \\ & (a_n, -?a_o + a_n, 0, -?a_o + a_n), \\ & - a_i / a_u + (a_i + a_n / a_u - ?a_o + a_n) > \\ \alpha_{22} \leftarrow & \alpha_{22} + (?a_o - a_n) \cdot \text{cm2} \\ \beta_1 \leftarrow & \beta_1, W_n = w(\text{hc}, ?\text{capacity}, \text{articulation}, q_{\text{level}}) \\ \beta_2 \leftarrow & \beta_2, h_n = h(\text{hc}, ?\text{capacity}, \text{articulation}, q_{\text{level}}) \\ \beta_3 \leftarrow & \beta_3, a_n = a(\text{hc}, ?\text{capacity}, \text{articulation}, q_{\text{level}}) \\ \beta_6 \leftarrow & \beta_6, \text{cm2} = \text{cost\_m2}(q_{\text{level}}, \text{covered}) \end{aligned}$$

### g123: Up/downgrading the global storage space (user)

$$\begin{aligned} \alpha_4 \leftarrow & \alpha_4, \text{Nusers} = \alpha_4 \\ \alpha_{10} \leftarrow & \alpha_{10}, \text{Nobligatory} = n(\text{global storage}), \text{capacity}_{\text{total}} = \sum_{i=1}^{\text{Nobligatory}} \text{capacity}(\text{global storage}_i) \\ \alpha_{13} \leftarrow & \alpha_{13} - < [\text{gs}, \text{id}_{\text{gs}}, (?\text{users}), (\text{global storage}), (?\text{capacity}, ?w_c), (?\text{articulation}, ?w_a), (?q_{\text{level}}, ?w_q), ?w_o, ?h_o, ?a_o] > \\ & + < [\text{gs}, \text{id}_{\text{gs}}, (?\text{users}), (\text{global storage}), (\text{capacity}, w_c), (\text{articulation}, w_a), (q_{\text{level}}, w_q), w_n, h_n, a_n] > \\ & \text{Ncurrent} = n(\text{global storage}), \text{capacity}_{\text{current}} = \sum_{i=1}^{\text{Ncurrent}} \text{capacity}(\text{global storage}_i) \\ & \text{capacity}_{\text{obligatory}} < \text{capacity}_{\text{current}} < \text{Nusers} + 2 \\ & \text{articulation} \in \{\text{included}, \text{isolated}\} \\ & q_{\text{level}} \in \{\text{min}, \text{med}, \text{max}\} \\ & W_c, W_a, W_q \in \{0, 5, 10, \dots, 100\} \\ & \text{capacity}_{\text{current}} > \text{capacity}_{\text{obligatory}} \Rightarrow W_c < 80 \\ & \text{articulation} > \text{included} \Rightarrow W_a < 80 \\ & q_{\text{level}} > \text{min} \Rightarrow W_q < 80 \\ \alpha_{14} \leftarrow & \alpha_{14} \exists [\text{zone}, ?\text{rooms}, ?\text{area}], \text{id}_{\text{gs}} \in ?\text{rooms} \\ & + \text{space}(\text{zone}, \text{id}_{\text{gs}}, a_n - ?a_o) \\ \alpha_{15} \leftarrow & \alpha_{15} + < \text{available}, (0, ?a_o - a_n, 0, ?a_o - a_n), (0, 0, 0, 0), (0, ?a_o - a_n, 0, ?a_o - a_n), 0 > \\ ?\text{articulation} \neq & \text{articulation} \wedge ?\text{articulation} = \text{included} \Rightarrow \\ \alpha_{16} \leftarrow & \alpha_{16} + < \text{used}, (-?a_o, -?a_o + a_n, 0, -?a_o + a_n), \\ & (0, 0, 0, 0), \\ & (-?a_o, -?a_o + a_n, 0, -?a_o + a_n), \\ & - a_i / a_u + (a_i - ?a_o / a_u - ?a_o + a_n) > \\ ?\text{articulation} \neq & \text{articulation} \wedge ?\text{articulation} = \text{isolated} \Rightarrow \\ \alpha_{16} \leftarrow & \alpha_{16} + < \text{used}, (a_n, -?a_o + a_n, 0, -?a_o + a_n), \\ & (0, 0, 0, 0), \\ & (a_n, -?a_o + a_n, 0, -?a_o + a_n), \\ & - a_i / a_u + (a_i + a_n / a_u - ?a_o + a_n) > \\ \alpha_{22} \leftarrow & \alpha_{22} + (?a_o - a_n) \cdot \text{cm2} \\ \beta_1 \leftarrow & \beta_1, W_n = w(\text{gs}, \text{capacity}, \text{articulation}, q_{\text{level}}) \\ \beta_2 \leftarrow & \beta_2, h_n = h(\text{gs}, \text{capacity}, \text{articulation}, q_{\text{level}}) \\ \beta_3 \leftarrow & \beta_3, a_n = a(\text{gs}, \text{capacity}, \text{articulation}, q_{\text{level}}) \\ \beta_6 \leftarrow & \beta_6, \text{cm2} = \text{cost\_m2}(q_{\text{level}}, \text{covered}) \end{aligned}$$

### g124: Adding a house clothing storage space (user)

$$\begin{aligned} \alpha_{10} \leftarrow & \alpha_{10}, \text{Nobligatory} = n(\text{house clothing storage}), \text{capacity}_{\text{total}} = \sum_{i=1}^{\text{Nobligatory}} \text{capacity}(\text{house clothing storage}_i) \\ \alpha_{13} \leftarrow & \alpha_{13} + < [\text{hc}, \text{id}_{\text{hc}}, (?\text{users}), (\text{house clothing storage}), (\text{capacity}, w_c), (\text{articulation}, w_a), (q_{\text{level}}, w_q), w_n, h_n, a_n] > \\ & \text{Ncurrent} = n(\text{house clothing storage}), \text{capacity}_{\text{current}} = \sum_{i=1}^{\text{Ncurrent}} \text{capacity}(\text{house clothing storage}_i) \\ & \text{capacity}_{\text{current}} < \text{Ndwellers} + 2 \end{aligned}$$

articulation  $\in$  {included, isolated}  
 $q_{level} \in$  {min, med, max}  
 $w_c, w_a, w_q \in$  {0, 5, 10, ..., 100}  
 capacity<sub>current</sub> > capacity<sub>obligatory</sub>  $\Rightarrow w_c < 80$   
 articulation > included  $\Rightarrow w_a < 80$   
 $q_{level} > \text{min} \Rightarrow w_q < 80$   
 first floor available area  $\geq a_n \Rightarrow$   
 $\alpha_{14} \leftarrow \alpha_{14} + \text{sleeping} (id_{hc}, + a_n)$   
 $\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, -a_n, 0, -a_n), (0, 0, 0, 0), (0, -a_n, 0, -a_n), 0 >$   
 articulation = included  $\Rightarrow$   
 $\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (a_n, a_n, 0, a_n), (0, 0, 0, 0), (a_n, a_n, 0, a_n), -a_i / a_u + (a_i + a_n / a_u + a_n) >$   
 articulation = isolated  $\Rightarrow$   
 $\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (0, a_n, 0, a_n), (0, 0, 0, 0), (0, a_n, 0, a_n), -a_i / a_u + (a_i / a_u + a_n) >$   
 $\alpha_{17} \leftarrow \alpha_{17}, + < [id_{hc}, id_{r1}, 80] >$   
 first floor available area <  $a_n \Rightarrow$   
 $\alpha_{14} \leftarrow \alpha_{14} + f2 (id_{hc}, + a_n)$   
 $\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, 0, 0, 0), (0, -a_n, 0, -a_n), (0, -a_n, 0, -a_n), 0 >$   
 articulation = included  $\Rightarrow$   
 $\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (0, 0, 0, 0), (a_n, a_n, 0, a_n), (a_n, a_n, 0, a_n), -a_i / a_u + (a_i + a_n / a_u + a_n) >$   
 articulation = isolated  $\Rightarrow$   
 $\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (0, 0, 0, 0), (0, a_n, 0, a_n), (0, a_n, 0, a_n), -a_i / a_u + (a_i / a_u + a_n) >$   
 $\alpha_{17} \leftarrow \alpha_{17} + < [id_{hc}, id_{r2}, 80] >$   
 $\alpha_{22} \leftarrow \alpha_{22} + a_n \cdot cm2$   
 $\beta_1 \leftarrow \beta_1, w_n = w (hc, \text{capacity}, \text{articulation}, q_{level},)$   
 $\beta_2 \leftarrow \beta_2, h_n = h (hc, \text{capacity}, \text{articulation}, q_{level})$   
 $\beta_3 \leftarrow \beta_3, a_n = a (hc, \text{capacity}, \text{articulation}, q_{level})$   
 $\beta_6 \leftarrow \beta_6, cm2 = \text{cost\_m2} (q_{level}, \text{covered})$

**g125: Adding a global storage space (user)**

$\alpha_{10} \leftarrow \alpha_{10}, n_{obligatory} = n (\text{global storage}), \text{capacity}_{total} = \sum_{i=1}^{n_{obligatory}} \text{capacity} (\text{global storage}_i)$   
 $\alpha_{13} \leftarrow \alpha_{13} + < [gs, id_{gs}, (?users), (\text{house clothing storage}), (\text{capacity } w_c), (\text{articulation}, w_a), (q_{level}, w_q), w_n, h_n, a_n] >$   
 $n_{current} = n (\text{global storage}), \text{capacity}_{current} = \sum_{i=1}^{n_{current}} \text{capacity} (\text{global storage}_i)$   
 capacity<sub>current</sub> <  $n_{dwellers} + 2$   
 articulation  $\in$  {included, isolated}  
 $q_{level} \in$  {min, med, max}  
 $w_c, w_a, w_q \in$  {0, 5, 10, ..., 100}  
 capacity<sub>current</sub> > capacity<sub>obligatory</sub>  $\Rightarrow w_c < 80$   
 articulation > included  $\Rightarrow w_a < 80$   
 $q_{level} > \text{min} \Rightarrow w_q < 80$   
 first floor available area  $\geq a_n \Rightarrow$   
 $\alpha_{14} \leftarrow \alpha_{14} + \text{sleeping} (id_{gs}, + a_n)$   
 $\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, -a_n, 0, -a_n), (0, 0, 0, 0), (0, -a_n, 0, -a_n), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (a_n, a_n, 0, a_n), (0, 0, 0, 0), (a_n, a_n, 0, a_n), -a_i / a_u + (a_i + a_n / a_u + a_n) >$   
 $\alpha_{17} \leftarrow \alpha_{17} + < [id_{gs}, id_{r1}, 80] >$   
 first floor available area <  $a_n \Rightarrow$   
 $\alpha_{14} \leftarrow \alpha_{14} + f2 (id_{gs}, + a_n)$   
 $\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, 0, 0, 0), (0, -a_n, 0, -a_n), (0, -a_n, 0, -a_n), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (0, 0, 0, 0), (a_n, a_n, 0, a_n), (a_n, a_n, 0, a_n), -a_i / a_u + (a_i + a_n / a_u + a_n) >$   
 $\alpha_{17} \leftarrow \alpha_{17}, + < [id_{gs}, id_{r2}, 80] >$   
 $\alpha_{22} \leftarrow \alpha_{22} + a_n \cdot cm2$   
 $\beta_1 \leftarrow \beta_1, w_n = w (gs, \text{capacity}, \text{articulation}, q_{level},)$   
 $\beta_2 \leftarrow \beta_2, h_n = h (gs, \text{capacity}, \text{articulation}, q_{level})$   
 $\beta_3 \leftarrow \beta_3, a_n = a (gs, \text{capacity}, \text{articulation}, q_{level})$   
 $\beta_6 \leftarrow \beta_6, cm2 = \text{cost\_m2} (q_{level}, \text{covered})$

**g126: Deleting a house clothing storage space (user)**

$\alpha_{10} \leftarrow \alpha_{10}, n_{obligatory} = n (\text{house clothing storage}), \text{capacity}_{total} = \sum_{i=1}^{n_{obligatory}} \text{capacity} (\text{house clothing storage}_i)$

?a<sub>0</sub> >

$$\alpha_{13} \leftarrow \alpha_{13} - \langle [hc, id_{hc}, (?users), (house\ clothing\ storage), (?capacity\ ?w_c), (?articulation, ?w_a), (?q_{level}, ?w_q), ?w_o, ?h_o, n_{current} = n (house\ clothing\ storage), capacity_{current} = \sum_{i=1}^{n_{current}} capacity (house\ clothing\ storage)_i \rangle$$

$$capacity_{current} > capacity_{obligatory}$$

$$\alpha_{14} \leftarrow \alpha_{14} \exists [zone, ?rooms, ?area], id_{hc} \in ?rooms$$

$$- space (zone, id_{hc}, - ?a_o)$$

zone ≠ 2 ⇒

$$\alpha_{14} \leftarrow \alpha_{14} + sleeping (id_{hc}, - a_o)$$

$$\alpha_{15} \leftarrow \alpha_{15} + \langle available, (0, a_o, 0, a_o), (0, 0, 0, 0), (0, a_o, 0, a_o), 0 \rangle$$

$$\alpha_{16} \leftarrow \alpha_{16} + \langle used, (-a_o, -a_o, 0, a_o), (0, 0, 0, 0), (-a_o, -a_o, 0, -a_o), -a_i / a_u + (a_i - a_o / a_u - a_o) \rangle$$

zone = 2 ⇒

$$\alpha_{14} \leftarrow \alpha_{14} + f2 (id_{hc}, - a_o)$$

$$\alpha_{15} \leftarrow \alpha_{15} + \langle available, (0, 0, 0, 0), (0, a_o, 0, a_o), (0, a_o, 0, a_o), 0 \rangle$$

$$\alpha_{16} \leftarrow \alpha_{16} + \langle used, (0, 0, 0, 0), (-a_o, -a_o, 0, -a_o), (-a_o, -a_o, 0, -a_o), -a_i / a_u + (a_i - a_o / a_u - a_o) \rangle$$

$$\alpha_{17} \leftarrow \alpha_{17} - \langle [id_{hc}, ?id, ?r, ?w] \rangle, \forall ?id, ?r, ?w$$

$$\alpha_{22} \leftarrow \alpha_{22} - ?a_o \cdot cm2$$

$$\beta_6 \leftarrow \beta_6, cm2 = cost\_m2 (q_{level}, covered)$$

**g127: Deleting a global storage space (user)**

$$\alpha_{10} \leftarrow \alpha_{10}, n_{obligatory} = n (global\ storage), capacity_{total} = \sum_{i=1}^{n_{obligatory}} capacity (global\ storage)_i$$

$$\alpha_{13} \leftarrow \alpha_{13} - \langle [gs, id_{hc}, (?users), (global\ storage), (?capacity\ ?w_c), (?articulation, ?w_a), (?q_{level}, ?w_q), ?w_o, ?h_o, ?a_o] \rangle$$

$$n_{current} = n (global\ storage), capacity_{current} = \sum_{i=1}^{n_{current}} capacity (global\ storage)_i$$

$$capacity_{current} > capacity_{obligatory}$$

$$\alpha_{14} \leftarrow \alpha_{14} \exists [zone, ?rooms, ?area], id_{gs} \in ?rooms$$

$$- space (zone, id_{gs}, - ?a_o)$$

zone ≠ 2 ⇒

$$\alpha_{14} \leftarrow \alpha_{14} - sleeping (id_{gs}, + a_o)$$

$$\alpha_{15} \leftarrow \alpha_{15} + \langle available, (0, a_o, 0, a_o), (0, 0, 0, 0), (0, a_o, 0, a_o), 0 \rangle$$

$$\alpha_{16} \leftarrow \alpha_{16} + \langle used, (-a_o, -a_o, 0, a_o), (0, 0, 0, 0), (-a_o, -a_o, 0, -a_o), -a_i / a_u + (a_i - a_n / a_u + - a_o) \rangle$$

zone = 2 ⇒

$$\alpha_{14} \leftarrow \alpha_{14} - f2 (id_{gs}, + a_o)$$

$$\alpha_{15} \leftarrow \alpha_{15} + \langle available, (0, 0, 0, 0), (0, a_o, 0, a_o), (0, a_o, 0, a_o), 0 \rangle$$

$$\alpha_{16} \leftarrow \alpha_{16} + \langle used, (0, 0, 0, 0), (-a_o, -a_o, 0, -a_o), (-a_o, -a_o, 0, -a_o), -a_i / a_u + (a_i - a_o / a_o - a_o) \rangle$$

$$\alpha_{17} \leftarrow \alpha_{17} - \langle [id_{gs}, ?id, ?r, ?w] \rangle, \forall ?id, ?r, ?w$$

$$\alpha_{22} \leftarrow \alpha_{22} - ?a_o \cdot cm2$$

$$\beta_6 \leftarrow \beta_6, cm2 = cost\_m2 (q_{level}, covered)$$

**g128: Deleting the house clothing storage space from the optional list spaces when the current capacity equals the maximum (programmer)**

$$\alpha_4 \leftarrow \alpha_4, n_{dwellers} = \alpha_4$$

$$\alpha_{10} \leftarrow \alpha_{10}, n_{obligatory} = n (house\ clothing\ storage), capacity_{total} = \sum_{i=1}^{n_{obligatory}} capacity (house\ clothing\ storage)_i$$

$$\alpha_{12} \leftarrow \alpha_{12} - \langle [house\_clothing\_storage, (included, isolated), 0] \rangle$$

$$\alpha_{13} \leftarrow \alpha_{13}, n_{current} = n (house\ clothing\ storage), capacity_{current} = \sum_{i=1}^{n_{current}} capacity (house\ clothing\ storage)_i$$

$$capacity_{current} = n_{dwellers} + 2$$

**g129: Adding the house clothing storage to the optional list when the current capacity is below the maximum (programmer)**

$$\alpha_4 \leftarrow \alpha_4, n_{adult\ dwellers} = \alpha_4$$

$$\alpha_{10} \leftarrow \alpha_{10}, n_{obligatory} = n (house\ clothing\ storage), capacity_{obligatory} = \sum_{i=1}^{n_{obligatory}} capacity (house\ clothing\ storage)_i$$

$$\alpha_{12} \leftarrow \alpha_{12} + \langle [house\_clothing\_storage, (included, isolated), 0] \rangle$$

$$\alpha_{13} \leftarrow \alpha_{13}, n_{current} = n (house\ clothing\ storage), capacity_{current} = \sum_{i=1}^{n_{current}} capacity (house\ clothing\ storage)_i$$

$$capacity_{current} < n_{dwellers} + 2$$

**g130: Deleting the global storage space from the optional list when the current capacity equals the maximum (programmer)**

$$\alpha_4 \leftarrow \alpha_4, n_{dwellers} = \alpha_4$$

$$\alpha_{10} \leftarrow \alpha_{10}, n_{obligatory} = n (global\ storage), capacity_{total} = \sum_{i=1}^{n_{obligatory}} capacity (global\ storage)_i$$

$$\alpha_{12} \leftarrow \alpha_{12} - \langle [global\_storage, (included, isolated), 0] \rangle$$

$$\alpha_{13} \leftarrow \alpha_{13}, n_{current} = n (global\ storage), capacity_{current} = \sum_{i=1}^{n_{current}} capacity (global\ storage)_i$$



$$\text{capacity}_{\text{current}} = \text{ndwellers} + 2$$

**g131: Adding the global storage space to the current list of optional spaces when the current capacity is below the maximum (programmer)**

$$\alpha_4 \leftarrow \alpha_4, \text{ndwellers} = \alpha_4$$

$$\alpha_{10} \leftarrow \alpha_{10}, \text{nobligatory} = n(\text{global storage}), \text{capacity}_{\text{obligatory}} = \sum_{i=1}^{\text{nobligatory}} \text{capacity}(\text{global storage}_i)$$

$$\alpha_{12} \leftarrow \alpha_{12} + \langle [\text{global\_storage}, (\text{included}, \text{isolated}), 0] \rangle$$

$$\alpha_{13} \leftarrow \alpha_{13}, \text{ncurrent} = n(\text{global storage}), \text{capacity}_{\text{current}} = \sum_{i=1}^{\text{ncurrent}} \text{capacity}(\text{global storage}_i)$$

## Exterior spaces

**g132: Up/downgrading a balcony (user)**

$$\alpha_6 \leftarrow \alpha_6, \text{ndwellers} = \alpha_4$$

$$\alpha_{13} \leftarrow \alpha_{13}$$

$$+ \langle [\text{bl}, \text{id}_{\text{bl}}, (?\text{users}), (\text{being outside}), (?capacity, ?w_c), (\text{isolated}, 100), (?q_{\text{level}}, ?w_q), ?w_o, ?h_o, ?a_o] \rangle$$

$$+ \langle [\text{bl}, \text{id}_{\text{bl}}, \emptyset, (\text{being outside}), (\text{capacity}, w_c), (\text{isolated}, 100), (q_{\text{level}}, w_q), w_n, h_n, a_n] \rangle$$

$$\text{ncurrent} = n(\text{being outside}) \wedge \langle [\text{id}_{\text{bl}}, \text{id}_{\text{l2}}, \text{on}, ?w_i] \rangle \in \alpha_{17}, \text{capacity}_{\text{current}} = \sum_{i=1}^{\text{ncurrent}} \text{capacity}(\text{being outside}_i)$$

$$\text{capacity}_{\text{current}} < \text{ndwellers} + 2$$

$$q_{\text{level}} \in \{\text{min}, \text{med}, \text{max}\}$$

$$w_c, w_a \in \{0, 5, 10, \dots, 100\}$$

$$\text{capacity}_{\text{current}} > \text{ndwellers} \Rightarrow w_c < 80$$

$$q_{\text{level}} > \text{min} \Rightarrow w_q < 80$$

$$\alpha_{14} \leftarrow \alpha_{14} + f_2(\text{id}_{\text{bl}}, -a_o + a_n)$$

$$\alpha_{15} \leftarrow \alpha_{15} + \langle \text{available}, (0, 0, 0, 0), (0, 0, a_o - a_n, -a_o), (0, 0, a_o - a_n, a_o - a_n), 0 \rangle$$

$$\alpha_{16} \leftarrow \alpha_{16} + \langle \text{used}, (0, 0, 0, 0), (0, 0, -a_o + a_n, -a_n + a_o), (0, 0, -a_n + a_o, -a_n + a_o), -a_i / a_u + (a_i / a_u - a_n + a_o) \rangle$$

$$\alpha_{17} \leftarrow \alpha_{17} + \langle [\text{id}_{\text{bl}}, \text{id}_{\text{l2}}, \text{on}, 100] \rangle$$

$$\alpha_{22} \leftarrow \alpha_{22} + (a_n - a_o) \cdot \text{cm}_2$$

$$\beta_1 \leftarrow \beta_1, w_n = w(\text{ba}, \text{capacity}, \text{isolated}, q_{\text{level}})$$

$$\beta_2 \leftarrow \beta_2, h_n = h(\text{ba}, \text{capacity}, \text{isolated}, q_{\text{level}})$$

$$\beta_3 \leftarrow \beta_3, a_n = a(\text{ba}, \text{capacity}, \text{isolated}, q_{\text{level}})$$

$$\beta_6 \leftarrow \beta_6, \text{cm}_2 = \text{cost\_m}_2(q_{\text{level}}, \text{uncovered})$$

**g133: Adding a balcony (user)**

$$\alpha_6 \leftarrow \alpha_6, \text{ndwellers} = \alpha_4$$

$$\alpha_9 \leftarrow \alpha_9, \alpha_9 = \text{true}$$

$$\alpha_{13} \leftarrow \alpha_{13}$$

$$+ \langle [\text{bl}, \text{id}_{\text{bl}}, \emptyset, (\text{being outside}), (\text{capacity } w_c), (\text{isolated}, 100), (q_{\text{level}}, w_q), w_n, h_n, a_n] \rangle$$

$$\text{ncurrent} = n(\text{being outside}) \wedge \langle [\text{id}_{\text{bl}}, \text{id}_{\text{l2}}, \text{on}, ?w_i] \rangle \in \alpha_{17}, \text{capacity}_{\text{current}} = \sum_{i=1}^{\text{ncurrent}} \text{capacity}(\text{being outside}_i)$$

$$\text{nbalconies} < 2$$

$$\text{capacity}_{\text{current}} < \text{ndwellers} + 2$$

$$q_{\text{level}} \in \{\text{min}, \text{med}, \text{max}\}$$

$$w_c, w_a \in \{0, 5, 10, \dots, 100\}$$

$$\text{capacity}_{\text{current}} > \text{ndwellers} \Rightarrow w_c < 80$$

$$q_{\text{level}} > \text{min} \Rightarrow w_q < 80$$

$$\alpha_{14} \leftarrow \alpha_{14} + f_2(\text{id}_{\text{bl}}, +a_n)$$

$$\alpha_{15} \leftarrow \alpha_{15} + \langle \text{available}, (0, 0, 0, 0), (0, 0, -a_n, -a_n), (0, 0, -a_n, -a_n), 0 \rangle$$

$$\alpha_{16} \leftarrow \alpha_{16} + \langle \text{used}, (0, 0, 0, 0), (0, 0, a_n, a_n), (0, 0, a_n, a_n), -a_i / a_u + (a_i / a_u + a_n) \rangle$$

$$\alpha_{17} \leftarrow \alpha_{17} + \langle [\text{id}_{\text{bl}}, \text{id}_{\text{l2}}, \text{on}, 100] \rangle$$

$$\alpha_{22} \leftarrow \alpha_{22} + a_n \cdot \text{cm}_2$$

$$\beta_1 \leftarrow \beta_1, w_n = w(\text{ba}, \text{capacity}, \text{isolated}, q_{\text{level}})$$

$$\beta_2 \leftarrow \beta_2, h_n = h(\text{ba}, \text{capacity}, \text{isolated}, q_{\text{level}})$$

$$\beta_3 \leftarrow \beta_3, a_n = a(\text{ba}, \text{capacity}, \text{isolated}, q_{\text{level}})$$

$$\beta_6 \leftarrow \beta_6, \text{cm}_2 = \text{cost\_m}_2(q_{\text{level}}, \text{uncovered})$$

**g134: Deleting a balcony (user)**

$$\alpha_6 \leftarrow \alpha_6, \text{ndwellers} = \alpha_4$$

$$\alpha_6 \leftarrow \alpha_6, q_{\text{level}} = \alpha_6$$

$$\alpha_9 \leftarrow \alpha_9$$

$\alpha_{13} \leftarrow \alpha_{13}$   
 $- < [bl, id_{bl}, \emptyset, (\text{being outside}), (?capacity, ?w_c), (\text{isolated}, ?w_a), (?q_{level}, ?w_q), ?w_o, ?h_o, ?a_o] >$   
 $n_{current} = n (\text{being outside}) \wedge < [id_{bl}, id_{l2}, on, ?w_l] > \in \alpha_{17}, capacity_{current} = \sum_{i=1}^{n_{current}} capacity (\text{being outside } i)$   
 $\alpha_9 = \text{false} \vee (n_{current} > 1 \wedge capacity_{current} > n_{dwellers})$   
 $\alpha_{14} \leftarrow \alpha_{14} + f2 (id_{bl}, -?a_o)$   
 $\alpha_{15} \leftarrow \alpha_{15} + < \text{available}, (0, 0, 0, 0), (0, 0, ?a_o, ?a_o), (0, 0, ?a_o, ?a_o), 0 >$   
 $\alpha_{16} \leftarrow \alpha_{16} + < \text{used}, (0, 0, 0, 0), (0, 0, -?a_o, -?a_o), (0, 0, -?a_o, -?a_o), -a_i / a_u + (a_i / a_u - ?a_o) >$   
 $\alpha_{17} \leftarrow \alpha_{17} - < [id_{bl}, id_{l2}, on, 100] >$   
 $\alpha_{22} \leftarrow \alpha_{22} - ?a_o \cdot cm2$   
 $\beta_6 \leftarrow \beta_6, cm2 = \text{cost\_m2} (q_{level}, \text{uncovered})$

**g135: Deleting the balcony from the optional list when the current number equals the maximum (programmer)**

$\alpha_{12} \leftarrow \alpha_{12} - < [\text{being outside}, (\text{isolated}), 0] >$   
 $\alpha_{13} \leftarrow \alpha_{13},$   
 $n_{current} = n (\text{being outside}) \wedge < [id_{bl}, id_{l2}, on, ?w_l] > \in \alpha_{17}$   
 $n_{current} = 2$

**g136: Adding the balcony to the optional list when the current number is below the maximum (programmer)**

$\alpha_{12} \leftarrow \alpha_{12} + < [\text{being outside}, (\text{isolated}), 0] >$   
 $\alpha_{13} \leftarrow \alpha_{13},$   
 $n_{current} = n (\text{being outside}) \wedge < [id_{bl}, id_{l2}, on, ?w_l] > \in \alpha_{17}$   
 $n_{current} < 2$

## TOPOLOGY

### Required

#### Setting the required topological relations among spaces (programmer)

- g137:**  $\alpha_6 \leftarrow \alpha_6, q_{level} = \alpha_6$   
 $\alpha_{13} \leftarrow \alpha_{13}, \text{kitchen}, \text{laundry} \in \alpha_{13}$   
 $\alpha_{17} \leftarrow \alpha_{17} + ((id_{kitchen}), (id_{laundry}), r, 100)$   
 if articulation(laundry) = isolated  
 $q_{level} = \text{min} \Rightarrow r = (\text{close}, \text{adjacent}, \text{door})$   
 $q_{level} = \text{med} \Rightarrow r = (\text{adjacent}, \text{close}, \text{door})$   
 $q_{level} = \text{max} \Rightarrow r = (\text{door}, \text{adjacent}, \text{close})$   
 if articulation(laundry) = delimited  $\Rightarrow r = (\text{passage})$   
 if articulation(laundry) = included  $\Rightarrow r = (\text{merged})$
- g138:**  $\alpha_{13} \leftarrow \alpha_{13}, \text{kitchen}, \text{informal dining} \in \alpha_{13}$   
 $\alpha_{17} \leftarrow \alpha_{17} + ((n_{kitchen}), (n_{informal dining}), r, 100)$   
 if articulation (informal dining) = isolated  
 $q_{level} = \text{min} \Rightarrow r = (\text{close}, \text{adjacent}, \text{window}, \text{door})$   
 $q_{level} = \text{med} \Rightarrow r = (\text{adjacent}, \text{close}, \text{window}, \text{door})$   
 $q_{level} = \text{max} \Rightarrow r = (\text{door}, \text{window}, \text{adjacent}, \text{close})$   
 if articulation (informal dining) = delimited  $\Rightarrow r = (\text{passage})$   
 if articulation (informal dining) = included  $\Rightarrow r = (\text{merged})$
- g139:**  $\alpha_{13} \leftarrow \alpha_{13}, \text{kitchen}, \text{formal dining} \in \alpha_{13}$   
 $\alpha_{17} \leftarrow \alpha_{17} + ((n_{kitchen}), (n_{formal dining}), r, 100)$   
 if articulation (formal dining) = isolated  
 $q_{level} = \text{min} \Rightarrow r = (\text{close}, \text{adjacent}, \text{window}, \text{door})$   
 $q_{level} = \text{med} \Rightarrow r = (\text{adjacent}, \text{close}, \text{window}, \text{door})$   
 $q_{level} = \text{max} \Rightarrow r = (\text{door}, \text{window}, \text{adjacent}, \text{close})$   
 if articulation (formal dining) = delimited  $\Rightarrow r = (\text{passage})$
- g140:**  $\alpha_{13} \leftarrow \alpha_{13}, \text{kitchen}, \text{pantry} \in \alpha_{13}$   
 $\alpha_{17} \leftarrow \alpha_{17} + ((n_{kitchen}), (n_{pantry}), r, 100),$   
 if articulation (pantry) = isolated  
 $q_{level} = \text{min} \Rightarrow r = (\text{close}, \text{adjacent}, \text{door})$

$Q_{level} = \text{med} \Rightarrow r = (\text{adjacent, close, door})$   
 $Q_{level} = \text{max} \Rightarrow r = (\text{door, adjacent, close})$   
 if articulation (pantry) = included  $\Rightarrow r = (\text{merged})$

**g141:**  $\alpha_{13} \leftarrow \alpha_{13}$ , kitchen, patio, street  $\in \alpha_{13}$   
 $\alpha_{17} \leftarrow \alpha_{17} + ((n_{kitchen}), (n_{patio}, n_{street}, n_{balcony}, n_{terrace}), r, 100)$ ,  
 $Q_{level} = \text{min} \Rightarrow r = (\text{adjacent, close, window, door})$   
 $Q_{level} = \text{med} \Rightarrow r = (\text{window, close, adjacent, door})$   
 $Q_{level} = \text{max} \Rightarrow r = (\text{door, window, adjacent, close})$

**g142:**  $\alpha_{13} \leftarrow \alpha_{13}$ , kitchen, living  $\in \alpha_{13}$   
 $\alpha_{17} \leftarrow \alpha_{17} + ((n_{kitchen}), (n_{living}), r, 100)$   
 $Q_{level} = \text{min} \Rightarrow r = (\text{close, adjacent, window, door})$   
 $Q_{level} = \text{med} \Rightarrow r = (\text{adjacent, close, window, door})$   
 $Q_{level} = \text{max} \Rightarrow r = (\text{door, window, adjacent, close})$

**g143:**  $\alpha_{13} \leftarrow \alpha_{13}$ , living, formal dining  $\in \alpha_{13}$   
 $\alpha_{17} \leftarrow \alpha_{17} + (n_{living}, n_{formal\ dining}, r, 100)$ ,  
 if type(formal dining) = isolated  
 $Q_{level} = \text{min} \Rightarrow r = (\text{close, adjacent, window, door})$   
 $Q_{level} = \text{med} \Rightarrow r = (\text{adjacent, close, window, door})$   
 $Q_{level} = \text{max} \Rightarrow r = (\text{door, window, adjacent, close})$   
 articulation(formal dining) = delimited  $\Rightarrow r = (\text{passage})$   
 articulation(formal dining) = included  $\Rightarrow r = (\text{merged})$

**g144:**  $\alpha_{13} \leftarrow \alpha_{13}$ , living, patio  $\in \alpha_{13}$   
 $\alpha_{17} \leftarrow \alpha_{17} + (n_{living}, n_i, r, 100)$ ,  $i \in \{\text{patio, terrace, street, balcony}\}$   
 $Q_{level} = \text{min} \Rightarrow r = (\text{adjacent, close, window, door})$   
 $Q_{level} = \text{med} \Rightarrow r = (\text{window, close, adjacent, door})$   
 $Q_{level} = \text{max} \Rightarrow r = (\text{door, window, adjacent, close})$

**g145:**  $\alpha_{13} \leftarrow \alpha_{13}$ , living, laundry  $\in \alpha_{13}$   
 $\alpha_{17} \leftarrow \alpha_{17} + (n_{living}, n_{laundry}, r, 100)$ ,  
 $Q_{level} = \text{min} \Rightarrow r = (\text{close, adjacent, window, door})$   
 $Q_{level} = \text{med} \Rightarrow r = (\text{adjacent, close, window, door})$   
 $Q_{level} = \text{max} \Rightarrow r = (\text{door, window, adjacent, close})$

**g146:**  $\alpha_{13} \leftarrow \alpha_{13}$ , global storage  $\in \alpha_{13}$   
 $\alpha_{17} \leftarrow \alpha_{17} + (n_{global\ storage}, n_{circulation}, r, 100)$ ,  
 $r = (\text{merged, door, adjacent, close})$   
 $Q_{level} = \text{min} \Rightarrow r = (\text{close, adjacent, merged, door})$   
 $Q_{level} = \text{med} \Rightarrow r = (\text{adjacent, merged, close, door})$   
 $Q_{level} = \text{max} \Rightarrow r = (\text{merged, door, adjacent, close})$

---

### Setting the recommended topological relations among spaces and floors (programmer)

**g147:**  $\alpha_{13} \leftarrow \alpha_{13}$ , bedroom  $\in \alpha_{13}$   
 $\alpha_{17} \leftarrow \alpha_{17} + (\text{bedroom, first floor, on}, 100)$   
 $(\text{bedroom, first floor, on}, w_{fp}) \notin \alpha_{17}$ ,  $w_{fp} \in \{5, 10, 15, \dots, 100\}$

**g148:**  $\alpha_{13} \leftarrow \alpha_{13}$ , bedroom, living  $\in \alpha_{13}$   
 $\alpha_{17} \leftarrow \alpha_{17} + (\text{bedroom, living, (same floor)}, 100)$   
 $(\text{bedroom, living, same floor}, w_{fp}) \notin \alpha_{17}$ ,  $w_{fp} \in \{5, 10, 15, \dots, 100\}$

**g149:**  $\alpha_{13} \leftarrow \alpha_{13}$ , bedroom, kitchen  $\in \alpha_{13}$   
 $\alpha_{17} \leftarrow \alpha_{17} + (\text{bedroom, kitchen, (same floor)}, 100)$   
 $(\text{bedroom, kitchen, same floor}, w_{fp}) \notin \alpha_{17}$ ,  $w_{fp} \in \{5, 10, 15, \dots, 100\}$

**g150:**  $\alpha_{13} \leftarrow \alpha_{13}$ , bathtub, bedroom  $\in \alpha_{13}$   
 $\alpha_{17} \leftarrow \alpha_{17} +$  (bathtub, bedrooms, (same floor), 100)  
 $n$  (bathtub, bedroom, same floor,  $w_{tp}$ )  $< n_{bedrooms} - 1$ ,  $w_{tp} \in \{5, 10, 15, \dots, 100\}$

**g151:**  $\alpha_{13} \leftarrow \alpha_{13}$ , lavatory, living  $\in \alpha_{13}$   
 $\alpha_{17} \leftarrow \alpha_{17} +$  (lavatory, living, (same floor), 100)  
(lavatory, living, same floor,  $w_{tp}$ )  $\notin \alpha_{17}$ ,  $w_{tp} \in \{5, 10, 15, \dots, 100\}$

**g152:**  $\alpha_{13} \leftarrow \alpha_{13}$ , lavatory, kitchen  $\in \alpha_{13}$   
 $\alpha_{17} \leftarrow \alpha_{17} +$  (lavatory, kitchen, (same floor), 100)  
(lavatory, kitchen, same floor,  $w_{tp}$ )  $\notin \alpha_{17}$ ,  $w_{tp} \in \{5, 10, 15, \dots, 100\}$

## Recommended

### Setting the default topological relations among spaces (programmer)

**g153:**  $\alpha_{17} \leftarrow \alpha_{17} +$  ( $n_1, n_2$ ,  
(any, merged, passage, door, window, adjacent, close, away, same floor, different floors),  $w_i$ )  
( $n_1, n_2, ?relation, ?w_i$ )  $\notin \alpha_{17}$ ,  $\forall ?relation, ?w_i, n_1, n_2 \neq \text{floor}_i, i \in \{1, 2\}$

### Setting the default topological relations among spaces and floors (programmer)

**g154:**  $\alpha_{17} \leftarrow \alpha_{17} +$  ( $n_1, n_2$ , (on, not on),  $w_i$ )  
( $n_1, n_2, ?relation, ?w_i$ )  $\notin \alpha_{17}$ ,  $\forall ?relation, ?w_i, n_1 = \text{floor}_i \wedge n_2 \neq \text{floor}_i, i \in \{1, 2\}$

## Optional

### Specifying the topological relations among spaces (user)

**g155:**  $\alpha_{17} \leftarrow \alpha_{17}$  - ( $n_1, n_2, ?r, ?w_i$ )  
+ ( $n_1, n_2, r, w_i$ )

if articulation ( $n_1$ ) = included  $\vee$  articulation ( $n_1$ ) = included  $\Rightarrow (n_i, n_j, ?r, ?w_i), i \in \{1, 2\}, \forall j \in N$

$?r = (r_1, r_2, \dots, r_{10})$

$r = (n_i, n_{k+1}, \dots, n_{i-1}, n_{i+1}, r_m)$ ,

$l \neq 1 \Rightarrow k = 1$ ,

$l = 1 \Rightarrow k = 2$ ,

$l = 10 \Rightarrow m = 9$ ,

$l \neq 10 \Rightarrow m = 10$

$l \in \{1, \dots, 10\}$

$r_{1, \dots, 10} \in \{\text{any, different floors, same floor, away, close, adjacent, window, door, passage, merged}\}$

$w_i \in \{0, 5, 10, \dots, 100\}$

### Specifying the topological relations among spaces and floors (user)

**g156:**  $\alpha_{17} \leftarrow \alpha_{17}$  - ( $n_1, n_2, ?r, ?w_i$ )  
+ ( $n_1, n_2, r, w_i$ )

$\forall ?relation, ?w_i, n_1 = \text{floor}_i \wedge n_2 \neq \text{floor}_i, i \in \{1, 2\}$

$?r = (r_1, r_2), r_1, r_2 \in \{\text{on, not on}\}$

$r = (r_2, r_1)$ ,

$w_i \in \{0, 5, 10, \dots, 100\}$

## AESTHETICS

### Introducing proportions and the corresponding default weights (Programmer)

**g157:**  $\alpha_{22} \leftarrow \alpha_{22}$ ,  
+  $\langle [1:1, 10], [1:\sqrt{2}, 9], [1:2, 9], [1:3, 9], [1:4, 9], [2:3, 9], [2:5, 9], [3:4, 9], [3:5, 9], [4:5, 9], [5:6, 9] \rangle$

### Set viewpoints weights

**g158:**  $\alpha_{23} \leftarrow \alpha_{23} + \text{set\_qLevel\_weight}(vp_i, w_i)$ ,

$v_i \in \{\text{function, spatiality, capacity, articulation, spaciousness, topology, aesthetics}\}$ ,  
 $w_i \in \{5, 10, 15, \dots, 100\}$

## AESTHETICS

### Proportion

#### Introducing proportions and the corresponding default weights (Programmer)

**g164:**  $\alpha_{23} \leftarrow \alpha_{23},$   
 $+ < [1:1, 10], [1:\sqrt{2}, 9], [1:2, 9], [1:3, 9], [1:4, 9], [2:3, 9], [2:5, 9], [3:4, 9], [3:5, 9], [4:5, 9], [5:6, 9] >$

#### Setting the weights assigned to proportions (User)

**g165:**  $\alpha_{23} \leftarrow \alpha_{23}$  - [?proportion, ?weight]  
 $+ [\text{proportion, weight}],$   
 $\text{proportion} \in \{1:1, 1:\sqrt{2}, 1:2, 1:3, 1:4, 2:3, 2:5, 3:4, 3:5, 4:5, 5:6\}$   
 $\text{weight} \in \{5, 10, 15, \dots, 100\}$

## ACTIVATE DESIGNER

#### **g166: Activate designer**

$\alpha_0 \leftarrow \alpha_0$   
 $\alpha_7 \leftarrow \alpha_7$   
 $\alpha_9 \leftarrow \alpha_9$   
 $\alpha_{22} \leftarrow \alpha_{22}$   
 $\alpha_{23} \leftarrow \alpha_{23}$   
 $\delta_1 \leftarrow < \text{nil, nil, nil, nil} >$   
 $\delta_2 \leftarrow < \text{nil, nil, nil, nil} >$   
 $\delta_3 \leftarrow < \text{nil} >$   
 $\delta_4 \leftarrow \emptyset$   
 $\delta_5 \leftarrow \emptyset$   
 $\delta_6 \leftarrow < \text{nil} >$   
 $\delta_7 \leftarrow < \text{nil} >$   
 $\delta_8 \leftarrow < \text{nil} >$   
 $\delta_9 \leftarrow < \text{nil} >$   
 $\delta_{10} \leftarrow \emptyset$   
 $\delta_{11} \leftarrow \emptyset$   
 $\delta_{12} \leftarrow \emptyset$   
 $\delta_{13} \leftarrow \emptyset$   
 $\delta_{14} \leftarrow < [\text{patio}, \emptyset], [\text{service}, \emptyset], [\text{living}, \emptyset], [\text{sleeping}, \emptyset], [\text{f2}, \emptyset] >$   
 $\delta_{15} \leftarrow \text{Ag} = \text{lot\_area}(\alpha_0) = 96.00 \text{ m}^2 \wedge \alpha_7 = \text{frontyard} \wedge \alpha_9 = \text{true} \Rightarrow$   
 $< \text{available}, (72.50, 23.50, \text{Ag}), (68.50, 4.00, 72.50), (0, 141.00, 27.50, 168.50), 0.77 >$   
 $\text{Ag} = \text{lot\_area}(\alpha_0) = 96.00 \text{ m}^2 \wedge \alpha_7 = \text{frontyard} \wedge \alpha_9 = \text{false} \Rightarrow$   
 $< \text{available}, (68.50, 27.50, \text{Ag}), (68.50, 0.00, 72.50), (0, 141.00, 27.50, 164.50), 0.77 >$   
 $\text{Ag} = \text{lot\_area}(\alpha_0) = 96.00 \text{ m}^2 \wedge \alpha_1 = < \text{street, ?context}_i, \text{house, ?context}_r > \wedge \alpha_7 = \text{backyard} \Rightarrow$   
 $< \text{available}, (72.97, 23.03, \text{Ag}), (56.80, 16.17, 72.97), (0, 129.77, 39.20, 168.97), 0.77 >$   
 $\text{Ag} = \text{lot\_area}(\alpha_0) = 96.00 \text{ m}^2 \wedge \alpha_1 = < \text{street, ?context}_i, \text{house, ?context}_r > \wedge \alpha_7 = \text{backyard} \wedge \alpha_9 = \text{true} \Rightarrow$   
 $< \text{available}, (72.50, 23.50, \text{Ag}), (68.50, 4.00, 72.50), (0, 141.00, 27.50, 168.50), 0.77 >$   
 $\text{Ag} = \text{lot\_area}(\alpha_0) = 96.00 \text{ m}^2 \wedge \alpha_1 = < \text{street, ?context}_i, \text{house, ?context}_r > \wedge \alpha_7 = \text{backyard} \wedge \alpha_9 = \text{false} \Rightarrow$   
 $< \text{available}, (68.50, 27.50, \text{Ag}), (68.50, 0.00, 72.50), (0, 137.00, 27.50, 164.50), 0.77 >$   
 $\delta_{16} \leftarrow (\text{used}, (0, 0, 0, 0), (0, 0, 0, 0), (0, 0, 0, 0), 0)$   
 $\delta_{17} \leftarrow \emptyset$   
 $\delta_{18} \leftarrow 0$   
 $\delta_{19} \leftarrow \emptyset$   
 $\delta_{20} \leftarrow \emptyset$   
 $\delta_{21} \leftarrow \emptyset$   
 $\delta_{22} \leftarrow \alpha_{22}$   
 $\delta_{23} \leftarrow \alpha_{23}$   
 $\delta_{24} \leftarrow 0$   
 $\delta_{25} \leftarrow < \text{R0} >$

## 7.5 The revised Malagueira grammar

This section presents the Malagueira designing grammar, which includes a revised version of the shape grammar presented in Chapter 5, and a description grammar. The initial grammar was changed following the experimental results discussed in Chapter 6 to increase the possibility of generating customized houses. There were three major changes.

The first change was to enlarge the universe of design solutions. Recall that the initial grammar only allowed the generation of designs that corresponded to the strictest interpretation of Siza's rules. The goal of the revised grammar is to permit the generation of designs that correspond to the broadest interpretation of such rules. This was achieved by writing the space allocating rules as a general algorithm that permits the allocation of any spaces specified in the housing program, while maintaining stylistic coherence.

The second change was to make rule application as deterministic as possible. In the initial grammar, a rule specified the context in which it could be applied in its left-hand side and in conditionals on dimension and function. However, this was not enough to make rule application deterministic, as several rules could be applied at each step in the derivation. In the revised grammar, dimensional and functional conditionals were incorporated into the description grammar, and heuristics that select the rule that takes the evolving design closer to the program description were introduced.

The third change was to permit moving back and forth between the design of the lower and upper floors. In the initial grammar, the generation of floors was strictly sequential as it started with the lower floor, then continued with the second, and terminated with the terrace. The drawback in this approach was that the derivation of the upper floor was completely constrained by the lower floor, which could jeopardize the possibility of satisfying upper floor requirements that were considered by the client more important than lower floor ones. In the revised grammar, the derivations of all floors proceed in parallel, and spaces allocation is determined by the heuristics mentioned above, independent from the floor.

Experimental results also showed that the derivation of a goal-matching design is not a continuous, linear progress towards the goal, but includes dead-ends and backtracking in an attempt to 'optimize' the solution. On the other hand, it also showed that many dead-ends and backtracking could be avoided if the right heuristics were used.

Therefore, effort was placed on the development of appropriate heuristics, and the grammar was revised accordingly. Nevertheless, it is yet to be shown that the best solutions can be found by heuristic search alone without 'optimization'.

The details of the revised grammar are explained over the next sections. The explanation does not include the complete grammar, but only enough detail to illustrate the mechanisms referred to above.

### **7.5.1 Algebras and parallel grammars: viewpoints and features**

As mentioned above, the Malagueira designing grammar includes both a shape grammar and a description grammar. Each of these grammars, in turn, includes several sub-grammars, as diagrammed in Table 7.36. These sub-grammars correspond to

viewpoints in the shape grammar (e.g. first floor plan), and to features in the description grammar (e.g. morphology). The viewpoints are defined in the same algebras used for the initial Malagueira grammar and they include the same viewpoints, plus two additional ones. These additional viewpoints—one for the lower floor and another for the upper one—are sketch viewpoints used to generate alternatives for assessment, before derivation proceeds with the best alternative. The description includes a program description, a fixed description, and a design description. The design description includes the same features as the program description, but they are identified with the Greek letter  $\delta$  instead of  $\alpha$ . Thus, for each program feature  $\alpha_n$  there is a design feature  $\delta_n$ . However, some features in the description are only manipulated by the one of the grammars. For instance, the minimum obligatory, initial obligatory, and the current optional spaces are only manipulated by the programming grammar.

**Table 7.36 – The features manipulated in the designing grammar**

Parts	Subparts	Identifier	Viewpoints and features	Algebra	More detail
Shape part (shape description)	3D	W	Envelope (walls)	$U_{33}$	See Figure 5.1
		R	Spaces (rooms)	$W_{33}$	
	2D	F1	1 <sup>st</sup> floor plan	$U_{12}V_{02}$	
		F1 sketch			
		F2	2 <sup>nd</sup> floor plan		
		F2 sketch			
		F3	Terrace		
		E	Elevation		
S	Stair section				
Description part (symbolic description)	Program description	$\alpha_1, \dots, \alpha_{25}$	Lot, ..., history	$U_{03}$	See Table 7.2
	Fixed description	$\beta_1, \dots, \beta_6$	Width, ..., cost		
	Design description	$\delta_1, \dots, \delta_{25}$	Lot, ..., history		

As in the initial grammar, the 3D viewpoints are used only for visualization purposes, and therefore, they are not included in the explanation below. The derivation is controlled both by the 2D viewpoints, and the description features. The derivation of the various



viewpoints and features can occur simultaneously or alternately, depending on the state of the design as diagrammed in Figure 7.1.)

Although the derivation of the design is not formally divided into steps, rules were grouped into the same analytical steps used for explaining the initial grammar, which are ‘start’, ‘locate functional zones’, ‘define circulation scheme’, ‘divide zones into rooms’, ‘introduce details’, and ‘introduce openings’. The rules of steps ‘introduce details’, and ‘introduce openings’ are not included in the explanation because the focus is on the derivation of the basic layout to illustrate the heuristic search mechanism.

Grammar	Define floors						
F <sub>1</sub>	S	Z	C	R	D	O	T
Sketch F <sub>1</sub>				↑↓	↑↓	↑↓	↑↓
F <sub>2</sub>				↑↓	↑↓	↑↓	↑↓
Sketch F <sub>2</sub>				↑↓	↑↓	↑↓	↑↓
F <sub>3</sub>				↑↓	↑↓	↑↓	↑↓
E				↑↓	↑↓	↑↓	↑↓
Description				↑↓	↑↓	↑↓	↑↓

Legend: F<sub>1</sub> – first floor; F<sub>2</sub> – 2nd floor; F<sub>3</sub> – terrace; E – elevation; S – Start; Z – Locate functional zones; C – Define circulation scheme; R – Divide zones into rooms; D – Introduce details; O – Introduce openings; T – Terminate.

Figure 7.17 – Use of parallel grammars in the derivation of a Malagueira house. Dark shaded areas identify main viewpoints, and light shaded areas identify sketch viewpoints. Letter symbols identify steps of the derivation; vertical line symbols mean that the derivation occurs simultaneously on the marked main viewpoints, with intermediate recourse to sketch viewpoints, if necessary; double-arrow symbols mean that the derivation might alternate from one or more viewpoints.

### 7.5.2 Rule types

‘Layout rules’ can be classified according to the type of operation they perform in the derivation of the design. (Table 7.37) There are composition, evaluation rules, transformation, and consistency rules. Composition rules encompass assigning,

dissecting, and concatenating rules, and their goal is to allocate spaces. Evaluation rules include generate, assess and select alternative rules, and they are part of the heuristic mechanism that controls the derivation. Transformation rules include extending and permuting rules, and they aim at improving the solution. Consistency rules include vertical and horizontal wall aligning rules, and they ensure that stylistic and structural coherence is maintained after transformation rules are applied. Transformation and consistency rules are only required if optimization is sought, which is not the case of the proposed revised grammar.

**Table 7.37 - Classification of layout rules according to the type of operation they perform**

Composition	Assigning
	Dissecting
	Concatenating
Evaluation	Generating alternatives
	Assessing alternatives
	Selecting best alternative
Transformation	Expanding
	Permuting
Consistency	Vertical wall aligning
	Horizontal wall aligning

### 7.5.3 Rules

The explanation of the rules used in the generation of the layout included in the revised grammar is provided below. The set of rules is presented in Table 7.38, at the end of Section 7.5.

#### 7.5.3.1 Step 0: Introduce initial shape

The single rule in this step (Rule R0) introduces the initial shape representing the lot. It is activated when the user hits the 'send' button on the interface and the programmer creates a design description, with the Rule R0 included its history ( $\alpha_{25}$ ). This rule retrieves the information on the lot ( $\alpha_0$ ), the urban context ( $\alpha_1$ ), and the solar orientation

( $\alpha_2$ ), contained in the program description, and introduces the initial shape in the correct context.

### **7.5.3.2 Step 1: Start**

Rules 1 through 4 apply at this step. Rule R1 introduces the floors and the pavements into the appropriate viewpoints. First, it checks the context features ( $\alpha_0$ ,  $\alpha_1$ , and  $\alpha_2$ ) and the number of floors ( $\alpha_8$ ) in the program description, and copies them onto the design description ( $\delta_0$ ,  $\delta_1$ ,  $\delta_2$ , and  $\delta_8$ ). Then, it introduces descriptions representing the floors to the list of current spaces ( $\delta_{13}$ ) and descriptions representing the pavements to the list of pavements ( $\delta_{21}$ ). It also introduces the corresponding topological relations ( $\delta_{17}$ ), and updates the available area ( $\delta_{15}$ ), the quality ( $\delta_{23}$ ), the cost ( $\delta_{24}$ ), and the history. Rule R2 encloses the floor. It adds the enclosing walls to the list of walls ( $\delta_{20}$ ), and updates the list of current spaces ( $\delta_{13}$ ), the available area ( $\delta_{15}$ ), the cost ( $\delta_{24}$ ), and the history ( $\delta_{25}$ ). Rules R3 and R4 adjust the thickness of the enclosing walls. They update the list of walls ( $\delta_{20}$ ), the cost ( $\delta_{24}$ ), and the history ( $\delta_{25}$ ). In Rule R4, the front elevation is widened due to the increase in the side wall thickness.

### **7.5.3.3 Step 2: Locate functional zones**

Rules R5 through 16 apply at this step. Rule R5 divides the first floor into inside and outside zones in five different ways, depending on the urban context ( $\alpha_1$ ), the number of floors ( $\alpha_8$ ), and the need for balconies on the second floor ( $\alpha_9$ ). For instance, for all urban context, if the yard is at the front, and there are balconies on the second floor, then split the first floor into two equal halves, so that the outside zone is at the front, and the inside zone is at the back. In this case, the rule subtracts the first floor and adds the two zones to the list of current spaces ( $\delta_{13}$ ), adds the splitting wall to the list of walls ( $\delta_{20}$ ),

and updates the available area ( $\delta_{15}$ ), the topological relations ( $\delta_{17}$ ), the cost ( $\delta_{24}$ ), and the history ( $\delta_{25}$ ). Rules R6 and R7 divide the second floor into inside and outside zones. In Rule R6, if the yard is at the front and there are no balconies on the second floor, or the yard is at the back, then it splits the second floor so that the splitting wall is above the one on the first floor. In Rule R7, if the yard is at the front, and there are balconies on the second floor, then it splits the second floor so that the splitting wall is one meter behind the one on the first floor. Rule R8 divides the third floor into inside and outside zones so that the splitting wall is above the one on the second floor. Rule R9 divides the inside zone to introduce an external corridor linking the street to the patio, if the patio is at the back, and there are no streets on the side and at the back. These rules update the appropriate features accordingly.

Rules 10 through 13 locate the functional zones—patio, living, sleeping, and service—either through assignment or through dissections perpendicular to the x axis. The rules for dissections perpendicular to the y axis were not included in the current grammar, given its illustrative purposes. The patio is the first zone to be located, because it has a major impact on functional organization by constraining window and door placement. Rule 10 locates the patio through assignment by turning the outside zone into the patio on all floors, if these are included in the current list of spaces ( $\delta_{24}$ ), the number of bedrooms is five ( $\alpha_5$ ), and the number of floors is two ( $\alpha_8$ ). It actuates by changing the name of records in all the relevant features. Rule 11 locates the patio by dissecting the outside zone into patio and an unnamed zone. It locates the patio on one side or the other of the outside zone, depending on the urban context. If there is a street on one side, the patio will most likely be located on the opposite side, to increase the possibility of opening windows. This decision follows the experimental results, which showed that

such a patio location yields better solutions in most of the cases, but in all of them. Rule 12 allocates one of the remaining functional zones through assignment using heuristics. Namely, it allocates the zone whose allocation is more important, that is, the zone whose allocation in the current design context will likely lead to the design solution that is the closest to the goal. Moreover, it locates the zone in all possible situations so that it can assess them later. The heuristic for determining the importance of a zone is explained below. Rule 13 locates the remaining zones in all possible situations thereby generating basic patterns.

Rule 14 assesses a basic pattern to determine its fitness. The heuristic for determining the fitness of a basic pattern is explained below. Rule 15 eliminates all the worst basic patterns, one by one, until only one is left, if all the six possible patterns have already been generated. Rule 16 chooses the best basic pattern to resume the derivation. Rule 17 replicates the basic pattern on the second and third floors. The description parts of all these rules, as well as the description part of the remaining rules will not be shown.

#### Heuristics for determining the importance of a zone

The importance of a zone is the weighted average between its spatiality importance and its topology importance as expressed in the equation

$$I_{\text{zone}} = \frac{I_{\text{spatiality}(\text{zone})} \times W_{\text{spatiality}} + I_{\text{topology}(\text{zone})} \times W_{\text{topology}}}{W_{\text{spatiality}} + W_{\text{topology}}},$$

where

$I_{\text{spatiality}(\text{zone})}$  is the importance of the zone's spatiality,  
 $W_{\text{spatiality}}$  is the weight assigned to spatiality in the housing program,  
 $I_{\text{topology}(\text{zone})}$  is the importance of the zone's topology, and  
 $W_{\text{topology}}$  is the weight assigned to topology in the housing program.

This equation implies an indirect comparison between spatiality importance and topology importance. To make such a comparison possible, it is necessary to translate them into a similar scale. This is achieved by comparing the importance of a zone from each viewpoint relative to the importance of the zone with bigger demands from the same viewpoint. Therefore, the relative importance has a value between 0 and 1, from any of the viewpoints.

#### Importance of the zone's spatiality

According to what was said above, the relative importance of the zone's spatiality is given by

$$I_{\text{topology}(\text{zone})} = \frac{\frac{\sum (a(\text{space}_i)) \times w_i}{\sum w_i}}{\frac{\sum (a(\text{space}_j)) \times w_j}{\sum w_j}},$$

where

$a(\text{space}_i)$  is the area of a space included in the zone,  
 $w_i$  is the average weight of the weights assigned to this spaces' spatiality features,  
 $a(\text{space}_j)$  is the area of a space included in the zone with bigger spatiality demands, and  
 $w_j$  is the average weight of the weights assigned to this spaces' spatiality features.

#### Importance of the zone's topology

Similarly, the relative importance of the zone's topology is given by

$$I_{\text{topology}(\text{zone})} = \frac{\frac{\sum r_i (\text{space}_z, \text{space}_n) \times w_i}{\sum w_i}}{\frac{\sum r_j (\text{space}_{z\text{max}}, \text{space}_n) \times w_j}{\sum w_j}},$$

where

$r_i (\text{space}_z, \text{space}_n)$  is a relation that involves one of the zone's spaces and any other space included in the housing program,  
 $w_i$  is the weight assigned to such a relation in the housing program,

$r_j$  (space<sub>zmax</sub>, space<sub>n</sub>) is a relation that involves a space of the zone with higher number of topological relations involving its spaces included in the housing program,  $w_j$  is the weight assigned to such a relation.

### Heuristics for determining the fitness of a basic pattern

The fitness of a basic pattern is the weighted average between its spatiality fitness and its topology fitness as expressed in the equation

$$f_{\text{pattern}} = \frac{f_{\text{spatiality}(\text{pattern})} \times W_{\text{spatiality}} + f_{\text{topology}(\text{pattern})} \times W_{\text{topology}}}{W_{\text{spatiality}} + W_{\text{topology}}},$$

where

$W_{\text{spatiality}}$  is the weight assigned to spatiality in the housing program,  
 $f_{\text{topology}(\text{zone})}$  is the pattern's topology fitness, and  
 $W_{\text{topology}}$  is the weight assigned to topology in the housing program.

Similar to what was said above about the importance of a space, the calculation of the fitness of a basic pattern implies an indirect comparison between spatiality fitness and topology fitness. This comparison also requires their translation into a similar 0-1 scale by measuring the fitness of a pattern from a viewpoint to the fitness of the best pattern from the same viewpoint.

### Spatiality fitness of a basic pattern

The pattern whose zone areas are closer to the respective requirements, and therefore, requires smaller area exchange among its zones, is considered the best from the spatiality viewpoint. The area requirements of a given zone are given by

$$a_{\text{required}}(Z_1) = \frac{\sum (a(\text{space}_i)) \times w_i}{\sum w_i}$$

where

$a(\text{space}_i)$  – is the area of each space included in the zone,  
 $w_i$  – is the average weight assigned to this space's spatiality requirements.

The area balance of a pattern is the sum of the area balances of its zones:

$$da_{\text{pattern}} = |a_{\text{allocated}}(z_1) - a_{\text{required}}(z_1)| + |a_{\text{allocated}}(z_2) - a_{\text{required}}(z_2)| + |a_{\text{allocated}}(z_3) - a_{\text{required}}(z_3)|$$

where

$a_{\text{allocated}}(z_n)$  is the area allocated for zone n, and  
 $a_{\text{required}}(z_1)$  is the required area for the same zone n.

The relative spatiality fitness of a pattern is, therefore, given by

$$f_{\text{spatiality}}(\text{current\_pattern}) = \frac{da_{\text{best\_pattern}}}{da_{\text{current\_pattern}}}$$

where

$da_{\text{best\_pattern}}$  – is the area difference of the pattern with the smallest area difference,  
 $da_{\text{current\_pattern}}$  – is the area difference of the current pattern.

### Topology fitness of a basic pattern

The topology fitness of a pattern is the sum of the ratio between the length of the wall separating two zones and the weighted number close relations between them; and the ratio between the weighted number of distance relations between them, and the length of the wall. The unit is added to the length of the wall to prevent the problem from becoming indeterminate when the zones do not share a wall.

$$p_{\text{topology}}(\text{current\_pattern}) = \frac{1 + l_{\text{wall}}(\text{zone}_i, \text{zone}_j)}{\sum r_{\text{ck}}(\text{zone}_i, \text{zone}_j) \times w_k} + \frac{\sum r_{\text{dk}}(\text{zone}_i, \text{zone}_j) \times w_k}{1 + l_{\text{wall}}(\text{zone}_i, \text{zone}_j)}$$

where

$i, j \in \{\text{patio, living, sleeping, service, front, left, back, right}\}$   
 $p_{\text{topology}}(\text{current\_pattern})$  is the topology performance of the current pattern,  
 $r_{\text{ck}}(\text{zone}_i, \text{zone}_j)$  is a close relation either between a space in zone i and space in zone j,  
or between a space in zone i and zone j (e.g. living room adjacent to south)  
 $r_{\text{dk}}(\text{zone}_i, \text{zone}_j)$  is a distance relation between a space in zone i and space in zone j, or  
between a space in zone i and zone j (e.g. living room adjacent to south),  
 $w_k$  is the weight associated with such a relation,  
 $l_{\text{wall}}(\text{zone}_i, \text{zone}_j)$  is the length of the wall between zone i and zone j.



The relative topology fitness of a pattern, it obtained by comparing its absolute topology fitness with that of the best pattern from this viewpoint

$$f_{\text{topology}}(\text{current\_pattern}) = \frac{P_{\text{topology}}(\text{current\_pattern})}{P_{\text{topology}}(\text{best\_pattern})}$$

### 7.5.3.4 Step 3: Define circulation scheme

To define the circulation scheme it is necessary to made three decisions. First, to choose the zone and the wall next to which the staircase will be placed; second, to select the type of staircase; and third, to chose the corridor on the second floor. A detailed study of the Malagueira existing and new houses made it possible to make the stair design rules more deterministic than the rules initially proposed, but not so deterministic as to avoid generate and testing alternatives.

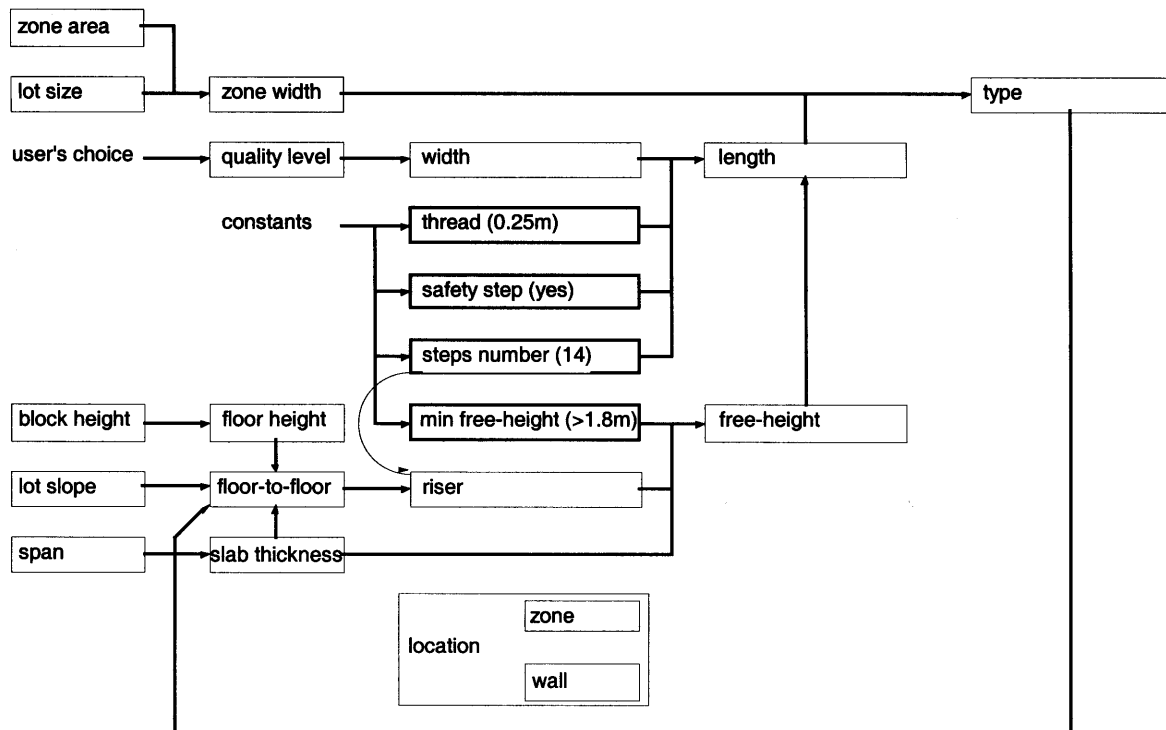


Figure 7.18 - The interdependency of stairs design variables in Malagueira houses.

There are many variables involved in the design of staircases such as type, width, rise, tread, number of steps, existence of safety step, minimum free height, and slope. These variables are dependent on each other as changing the value of one, affects the values of others, especially in tight design contexts. Part of the designer's effort is, therefore, to establish a hierarchy among these variables to decrease the range of possibilities and cope with the associated complexity. The analysis of the Malagueira houses revealed the hierarchy established by Siza, which is diagrammed in Figure 7.18.

Some variables are treated as constants. The number of steps is always 14 and the treads are always 0.25 m deep. All the stairs have a safety step, and the required free-height is 1.80m. Among the remaining variables, the only that seems to work as an independent variable is the width, whose value is determined by the user-chosen quality level, according to Table 7.30 in Section 7.4.10. The values of the other variables are dependent on the width or on the values of other design variables as explained below.

The riser height can vary depending on the floor-to-floor height, but it never exceeds the tread depth. The floor-to-floor height is given by the following system of equations and inequations

$$h_{\text{floor to floor}} = h_{\text{floor height}} + h_{\text{pavement}} \wedge h_{\text{floor height}} \geq 2.40\text{m} \wedge h_{\text{floor height}} = f(h_{\text{concrete blocks}})$$

where

$h_{\text{floor to floor}}$  is the floor to floor height,  
 $h_{\text{floor height}}$  is the floor height, and  
 $h_{\text{pavement}}$  is the thickness of the second floor's pavement.

As mentioned in Section 4.3.2, the floor-to-floor height also is determined by the rules that specify the height of walls on the front façade when the lot is not flat. In the current

version of the grammar, the rules for adjusting the height of walls on the front façade lot is inclined were not incorporated because it was not possible to confirm how they exactly worked<sup>1</sup>. The thickness of the second floor's pavement is determined by the height of the pre-stressed concrete beams, which in turn is determined by the maximum span on the first floor. For simplification purposes, variations in the pavement thickness also were not considered, but these could easily be incorporated by introducing a table relating the height of beams to the span. These tables are provided by the manufacturers of the pre-stressed concrete beams and tiles used in the construction of the pavement, and they are commonly used in practice by designers, including Siza. The floor height is constrained by existing regulation to be bigger than 2.40 m, which is the value it takes by default when the lot is flat. The floor height also is constrained by the height of the concrete blocks used for the walls, as mentioned in Section 4.3.2.

The riser height, and the floor height will determine the available free-height, which will have an impact on the stairs length. The stairs length and the zone width will determine the type of staircase that can be allocated in the zone.

### Types of staircases

There are three possible types of staircases, I-shaped (or straight), L-shaped, and U-shaped (Figure 7.19). The analysis of the houses in the corpus shows that I-shaped and L-shaped staircases are located in the living zone, whereas the U-shaped one is located in the sleeping or service zones. This analysis suggested that the type of staircase

---

<sup>1</sup> According to the Malagueira building regulations that can be consulted in Évora city hall, the owners of lots in the private promotion sector of the development must get the floor and façade walls heights from the city hall. This requirement, and the fact that such lots have a fairly steep topography suggests that there must exist rules to get the floor and façade heights, and that they must be know by the municipal architects. A new trip to the city hall should help to clarify these rules.

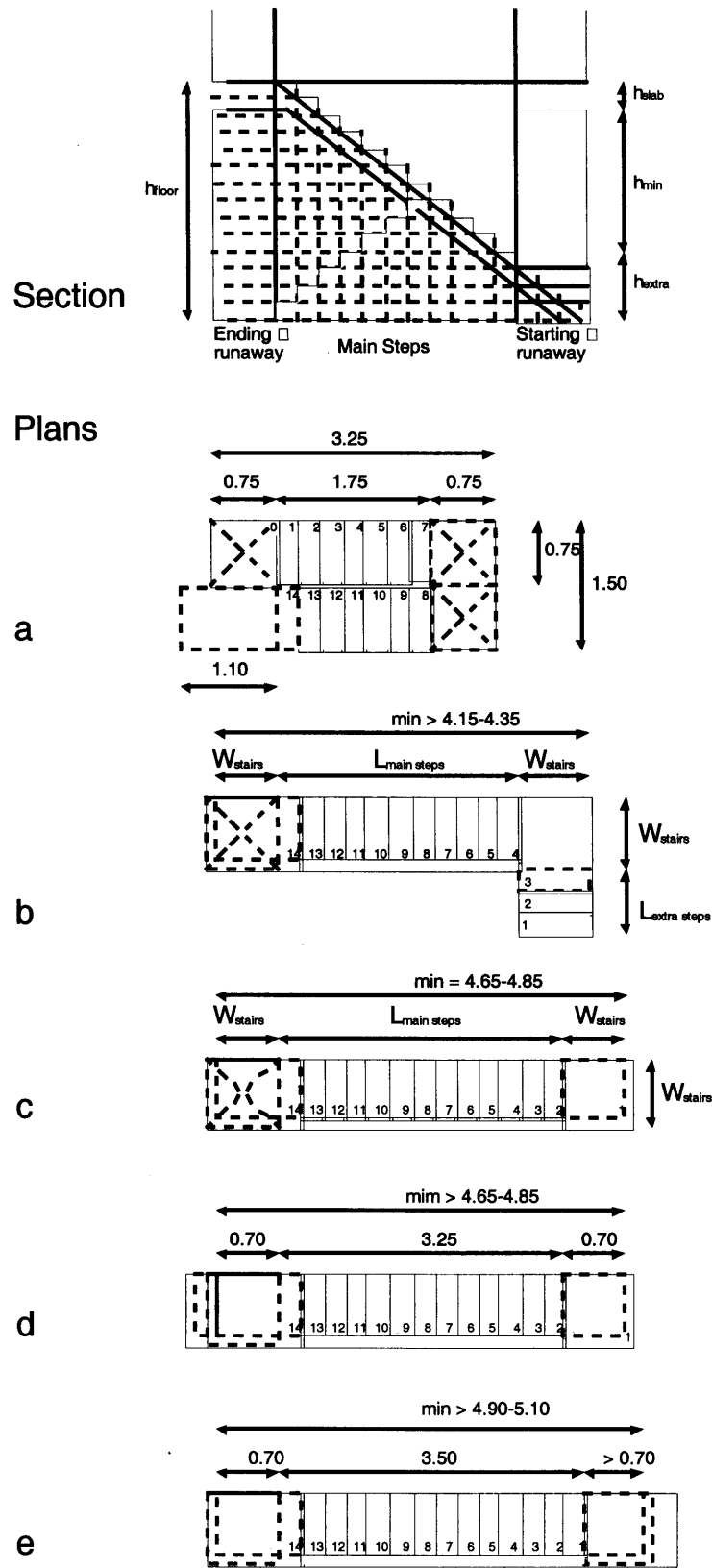


Figure 7.19 - Types and dimensions of staircases in Malagueira houses.

depended on the type of functional zone, and so this rule was followed in the first version of the Malagueira grammar, the one used in the experiments. However, a more careful analysis also showed that there was also a dependency between the widths of the functional zones where staircases are located, and the type of staircase. Namely, the widths of the zones with I-shaped staircases are bigger than those of the zones with L-shaped ones, and these are bigger than the widths of the zones with U-shaped staircases. This observation thus suggested that the type of staircase should depend on the width of the functional zone where the staircase is to be located. This rule does not contradict the previous analysis as it explains that the zones with I-shaped staircases are wider because living zones tend to have greater area requirements. An explanation of the computations used in the rules for stair design in accordance with the dependencies mentioned above is given below.

### I and L-shaped staircases

The stair design has to fulfill the following condition

$$(1) \quad h_{\min} > h_{\text{person}}$$

where  $h_{\min}$  is the height of the free space between the edge of the second floor and the edge of the step immediately below, and  $h_{\text{person}}$  is the threshold below which is the majority of the population's height. Up to the mid-eighties, this threshold was 1.80 m in Portugal, and so Siza used it in the design of Malagueira houses. However, due to the sharp increase of the average height of the Portuguese population during the past 15 years, this value has increased to 1.90 m, which the value used in the revised grammar.

Substituting in the condition above  $h_{\min}$  and  $h_{\text{person}}$  for the expression used to calculate  $h_{\min}$  and the value of  $h_{\text{person}}$  successively yields the conditions

$$\begin{aligned} & h_{\text{floor-to-floor}} - (h_{\text{slab}} + h_{\text{extra}}) > 1.90 \text{ m}, \\ & h_{\text{floor-to-floor}} - (h_{\text{slab}} + n \cdot \text{rise}) > 1.90 \text{ m}, \\ (2) \quad & h_{\text{floor-to-floor}} - [h_{\text{slab}} + n \cdot (h_{\text{floor-to-floor}} / 14)] > 1.90 \text{ m}, \end{aligned}$$

where

$h_{\text{floor}}$  is the floor to floor height,  
 $h_{\text{slab}}$  is the thickness of the second floor's pavement,  
 $h_{\text{extra}}$  is the height of the flight of stairs up to the minimum free space, and  
 $n$  is a number approximate to the maximum number of extra steps.

Solving this inequation in terms of  $n$ , it yields

$$(3) \quad n < (14 \cdot (h_{\text{floor-to-floor}} - h_{\text{slab}} + 1.80)) / h_{\text{floor-to-floor}}$$

In most cases, this number is not an integer, and therefore, the maximum number of steps in the extra flight of stairs is equal to the integer part of this number:

$$(4) \quad n_{\text{extra steps}} = \text{int}(n).$$

The *minimum number of steps* in the main flight of stairs is then given by subtracting the maximum number of steps in the extra flight of stairs from the total number of steps:

$$(5) \quad n_{\text{main steps}} = 14 - n_{\text{extra steps}}$$

The *minimum length* required to allocate an I-shaped staircase is given by

$$(6) \quad L_{\min} = L_{\text{starting runaway}} + L_{\text{main steps}} + L_{\text{ending runaway}}$$

Given that the starting and ending runways should be squares according to Siza's rules, the equation above is equivalent to

$$(7) \quad L_{\min} = 2 \cdot W_{\text{stairs}} + L_{\text{main steps}}$$

Finally, by finding the difference between the width (or length) of the functional zone and the minimum value require to allocate an I-shaped staircase

$$(8) \quad L_{\text{extra}} = W_{\text{li}} - L_{\text{mins}}$$

one can determine the shape of the staircase that can be allocated in the functional zone under consideration. There are four possible cases:

$$1. \quad L_{\text{extra}} < 0$$

If the difference is negative, the width of the functional zone is smaller than the minimum length required and, therefore, neither an I-shaped nor an L-shaped staircase can be allocated (Figure 7.19a).

$$2. \quad 0 \leq L_{\text{extra}} < n_{\text{extra steps}} \cdot \text{tread}$$

If the difference is non-negative and smaller than the maximum length of the extra flight of stairs, the staircase should be L-shaped, in which the lengths of the main and the extra flight of stairs are given by:

$$(9) \quad L_{\text{main steps}} = [(n_{\text{main steps}} + \text{integer}(L_{\text{extra}} / \text{tread})) \cdot \text{tread}]$$

$$(10) \quad L_{\text{extra steps}} = [(n_{\text{main steps}} - \text{integer}(L_{\text{extra}} / \text{tread})) \cdot \text{tread}]$$

The non-integer part of the division of the extra length by the tread is added to the ending runaway and the starting runaway remains a square (Figure 7.19b):

$$(11) \quad L_{\text{ending runaway}} = W_{\text{stairs}} + \text{remainder}(L_{\text{extra steps}} / \text{tread})$$

$$(12) \quad L_{\text{starting runaway}} = W_{\text{stairs}}$$

When the lot is flat, meaning that the floor height is 2.40m, and the floor thickness is maximum (0.30m), the minimum length required to allocate such a staircase is 4.15m ( $W_{\text{stairs}} = 0.70\text{m}$ ,  $L_{\text{runaway}} = W_{\text{stairs}}$ ,  $n_{\text{extra steps}} = 3$ ) and the maximum is 5.00m ( $W_{\text{stairs}} = 0.90\text{m} + 0.10\text{m}$ ,  $L_{\text{runaway}} = W_{\text{stairs}}$ ,  $n_{\text{extra steps}} = 2^2$ ). When the lot is so steep that the floor height is maximum, and the span is small enough for the floor thickness to be minimum, the minimum length required to allocate such a staircase is 4.15m still ( $W_{\text{stairs}} = 0.70\text{m}$ ,  $L_{\text{runaway}} = W_{\text{stairs}}$ ,  $n_{\text{extra steps}} = 2$ ) and the maximum is 5.00m ( $W_{\text{stairs}} = 0.90\text{m} + 0.10\text{m}$ ,  $L_{\text{runaway}} = W_{\text{stairs}}$ ,  $n_{\text{extra steps}} = 3$ ).

$$3. \quad n_{\text{extra steps}} \cdot \text{tread} \leq L_{\text{extra}} < (n_{\text{extra steps}} + 1) \cdot \text{tread}$$

If the difference between the available width and the minimum width is bigger than the maximum length of the extra flight of stairs, but smaller than the maximum length of the extra flight of stairs plus one, the staircase should be I-shaped. The length of the flight of stairs is given by the same equation as above, but in this case, it yields the value 3.00m. The lengths of the start and ending runaways also are calculated as above (Figure 7.19c and d):

The minimum length required to allocate such a staircase is 4.65m ( $W_{\text{stairs}} = 0.70\text{m}$ ,  $L_{\text{runaway}} = W_{\text{stairs}}$ ) and the maximum is 5.25m ( $W_{\text{stairs}} = 0.90\text{m} + 0.1\text{m}$ ,  $L_{\text{runaway}} = W_{\text{stairs}}$ ).

$$4. \quad L_{\text{extra}} \geq (n_{\text{extra steps}} + 1) \cdot \text{tread}$$

---

<sup>2</sup> When the number of extra steps is 1, the landing becomes the extra step, and the staircase is I-shaped.



If the difference between the available width and the minimum width exceeds the maximum length of the extra flight of stairs plus one, the allocated staircase is I-shaped. The length of the flight of stairs is obtained as above, but the non-integer part of the ratio between the extra length and the tread is added to the starting runaway, and it is the ending runaway that remains a square (Figure 7.19e):

$$(13) \quad L_{\text{ending runaway}} = W_{\text{stairs}} + \text{safety step}$$

$$(14) \quad L_{\text{main steps}} = [(n_{\text{main steps}} + \text{integer}(L_{\text{extra}} / \text{tread})) \cdot \text{tread}]$$

$$(15) \quad L_{\text{starting runaway}} = W_{\text{stairs}} + \text{remainder}(L_{\text{extra}} / \text{tread})$$

In this case, the starting runaway does not constitute a step. Although none of the houses in the corpus followed this rules, it was followed in the new design in Experiment 2, which was shown to Siza in one of the experiments, who validate the design.

Therefore, the rule was considered correct.

The minimum length required to allocate such a staircase is 4.90m ( $W_{\text{stairs}} = 0.70\text{m}$ ,  $L_{\text{runaway}} = W_{\text{stairs}}$ ) and the maximum length is the maximum width (or length) of functional zones (7.8m).

### U-shaped staircase

If the functional zone is not wide enough to allocate an I-shaped or an L-shaped staircase, a U-shaped staircase can be allocated. The minimum width ( $L_{\text{min}}$ ) required to allocate this type of staircase is given by the equation:

$$(16) \quad L_{\text{min}} = 2 \cdot (W_{\text{stairs}} + 0.10) + n_{\text{steps}} \cdot \text{tread},$$

where

$W_{\text{stairs}}$  is the width of stairs,  
0.10 is the thickness of the wall between the two flights of stairs, and

$n_{\text{steps}}$  is the number of steps of in each flight of stairs.

Thus the minimum width required to allocate such a staircase is 3.25m ( $W_{\text{stairs}} = 0.75$ ), and the maximum is 3.65m. Thus, there is a gap between the maximum width required to allocate a U-shaped staircase (3.65m), and the minimum width required to allocate an L-shaped one (4.00m). In such circumstances, the width of the functional zone is adjusted to the closest value. The hypothetical alternative of allocating a J-shaped staircase is not valid because it would use a larger area, or require to carve a J-shaped stairwell out of the second floor's pavement, which is not possible for the structural reasons mentioned above.

In summary, the design variables that decide the type of staircase that can be allocated in a functional zone are its width (or length), and the stairs length. If the functional zone is large enough -- at least twice the width of the stairs plus the minimum length of the main steps flight, the staircase will be I-shaped (Rule R19). In this type of staircase, twelve of the fourteen steps constitute the body of the stairs, which is bounded by runaways on each side that form the remaining two-steps. If the functional zone is not large enough to allocate an I-shaped staircase, the linear staircase takes the form of an L (Rule 20), or invades the neighboring zone (Rule 21). The minimum length of the long tail of the L-shaped staircase is restricted to 10 steps to guarantee that a person does not hit the ceiling when climbing the stairs. This restriction is imposed by structural constraints that require the stairwell to be a rectangle so that the ceiling's pre-stressed concrete beams can run parallel to the stairs without interruption. All the rules for placing the staircase adjust the dimension of the zone in which the staircase is placed to comply with the rule for stair design just described. If the functional zone is not large

enough to allocate an L-shaped staircase, a U-shaped staircase is allocated (Rule 22 and 23).

The relation between the width of functional zones and the type of staircases is diagrammed in Figure 7.20.

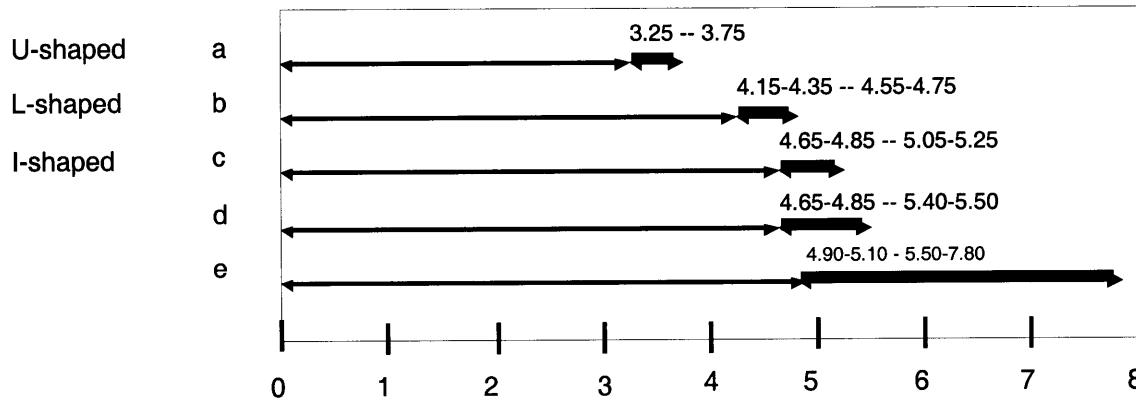


Figure 7.20 – The dependency between the width of zones and the type of staircase allocated

### Locating the staircase

By considering the width of the zones, it is possible to narrow the choices of staircase placement down. Nevertheless, it is not enough to determine in which zone and wall to locate the staircase. Therefore, the solution is to generate all of the alternatives, and then to pick up the one with the highest housing quality. This is accomplished by using Rules R18, and R24-30.

Rule 18 generates all the possible proto-stair patterns, which will potentially generate the stair patterns shown in Figure 5.13 if the placement of a staircase by any of the rules 19-23 mentioned above succeeds. Rules R25 through R26 delete proto-stair patterns if none of those rules succeed in turning them into stair patterns. Rule R27 assesses a stair pattern and records its fitness using the heuristic:

$$\text{Fitness} = V_p - V_d$$

Where

$V_p$  is the quality of the program description, and

$V_d$  is the quality of the current design description.

Rule R28 deletes the functional zones, one by one, of all the stair patterns that are not the best. Rule R29 completes the elimination of such patterns by erasing their identification and fitness markers. Rule R30 selects the best pattern for continuing with the derivation.

#### Create the circulation scheme on the second floor

The definition of the house's circulation scheme is completed with the creation of the corridor on the second floor. This is accomplished with Rules R31-34, which are identical to rules 122-125 of the initial grammar.

#### **7.5.3.5 Step 4: Divide zones into rooms**

Once a stair pattern has been generated, space-allocation rules apply to generate the basic layout. These rules apply to both the first and the second floors, as long as the right shapes and conditions are found in the evolving design, as explained below.

The space allocation problem is illustrated in Figure 7.21. There are five spaces involved in the computation. The computation starts with a space one wishes to introduce in the design -- the space to allocate, and a space where one will attempt to allocate it -- the existing space. If allocation is successful, two spaces will result: the allocated space, and the remaining space. If the allocated space is larger than the space to allocate, the difference is an extra space.

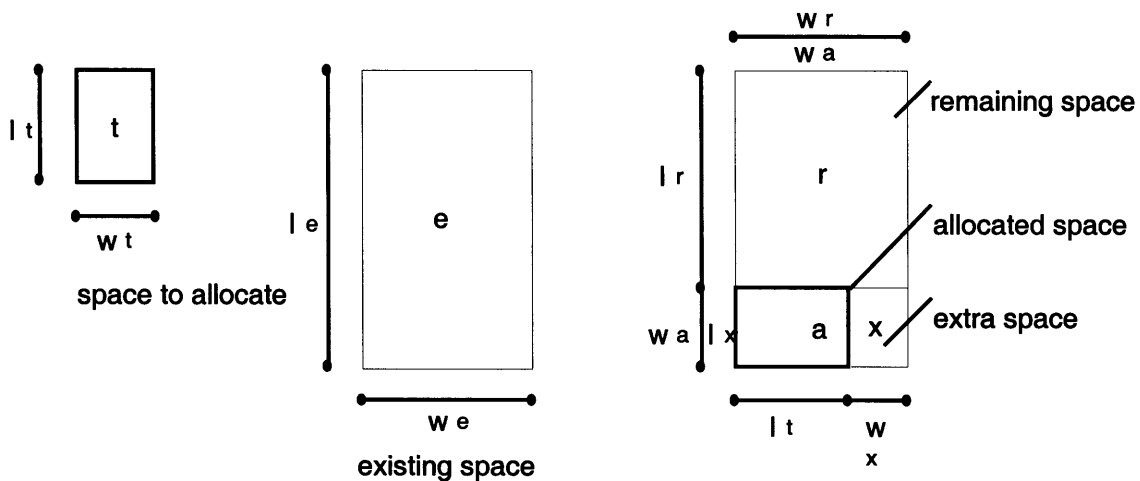


Figure 7.21 - The space allocation problem. There are five spaces involved in the computation: the space to allocate ( $t$ ), the existing space ( $e$ ), the extra space ( $x$ ), the allocated space ( $a$ , gray), and the remaining space ( $r$ ).

There are six different dissecting rules (Rules R36-41). Rules R36 and R37 dissect an existing space with a wall that is perpendicular to its width. In Rule 37, the extra space is big enough to let the allocated space be dissected again, whereas in Rule R36 it is too small to let it happen. This is shown in the rule by shading the allocated space. Rules R38 and R39 dissect the existing space with a wall that is perpendicular to its length, whereas Rules R40 and R41 dissect it with a wall that is perpendicular to the previous level dissection. The assigning rule can be seen as a special case in which the width of the remaining space is zero, although it is considered separately. The basic steps of the algorithm behind these rules, including the conditions that determine the type of space allocating rule that is fired, are outlined below and explained in following:

1. Pick up a space to allocate.
2. Pick up a space where to allocate it.
3. Get the dimensions of the space to allocate.
4. Get the dimensions of the existing space.
5. Does the space to allocate fit in the existing space?
6. If it does, which rule and transformation to use to allocate the space?
7. If it does not, can the requirements be lowered to make it fit?
  - 7.1 If it can, find how and allocate the space.
  - 7.2 If it cannot, give up allocating the space and report the failure.

## 1. Pick up a space to allocate

The first step is to select a space to allocate. The order in which allocation proceeds has an impact on what can be done in subsequent steps of the generation. Random selection would not solve the problem in the best way, as it would most likely require frequent backtracking. In fact, it would be better appropriate for using in a stochastic process. Heuristic selection is proposed as an alternative. Heuristic selection has the potential to decrease backtracking and shorten the search process. The basis of the proposed heuristic is twofold. First, it stems from the analysis of Siza's houses derivations proposed after the first version of the grammar. Such an analysis revealed that bigger spaces were allocated first. This observation was confirmed by the analysis of the experimental results. A deeper analysis revealed that bigger spaces also tended to have higher demands regarding other design features such as topology and proportion. Therefore, the proposed heuristic selects the space with the heaviest requirements, which is given by the equation

$$\text{space}_t = \max (h (\text{space}_j)), \text{space} \in \alpha_{13} \wedge \delta_{13} \leftarrow \delta_{13}$$

where  $\text{space}_t$  is the space to allocate, and  $h (\text{space}_j)$  is the function

$$h (\text{space}_j) = \frac{\sum V_{\text{space}_j} \cdot W_{\text{space}_j}}{\sum W_{\text{space}_j}}$$

where  $V_{\text{space}_j}$  is the value required for a given requirement, and  $w_{\text{space}_j}$  is the weight associated with such requirement as specified by the user.

## 2. Pick up a space where to allocate it

Once a space to allocate has been selected, the next steps are to find the zone in which the space can be allocated, given by

$$\text{zone} = \text{get\_zone} (\text{space}_j)$$

and then determine whether such a zone exists in the evolving design

$$\delta_{13} \leftarrow \delta_{13}, \exists [\text{zone}, \text{id}_e, \emptyset, ((x_e, y_e, z_e), dx_e, dy_e, dz_e, a_e), n] \in \delta_{13}$$

If the zone does not exist, the space cannot be allocated and the rule will not be fired.

The space with the next heaviest requirements is then chosen, and so on until a space and the corresponding required zone are found.

### 3. Get the dimensions of the space to allocate

The next step retrieves the area, width, and length of the space to allocate:

$$\begin{aligned} a_t &= \text{get\_area}(\text{space}_i), \\ w_t &= \text{get\_width}(\text{space}_i), \text{ and} \\ l_t &= a_t / w_t. \end{aligned}$$

### 4. Get the dimensions of the existing space

The next step retrieves the linear dimensions of the existing space and finds which is the width and which is the length:

$$\begin{aligned} dx_e < dy_e &\Rightarrow dx_e = w_e \wedge dy_e = l_e, \text{ and} \\ dx_e > dy_e &\Rightarrow dx_e = l_e \wedge dy_e = w_e. \end{aligned}$$

### 4. Does the space to allocate fit in the existing space?

The next series of steps are aimed at finding whether the space to allocate fits the existing space. This is determined by comparing the areas, width, and length of these spaces. The problem is illustrated in Figure 7.22.

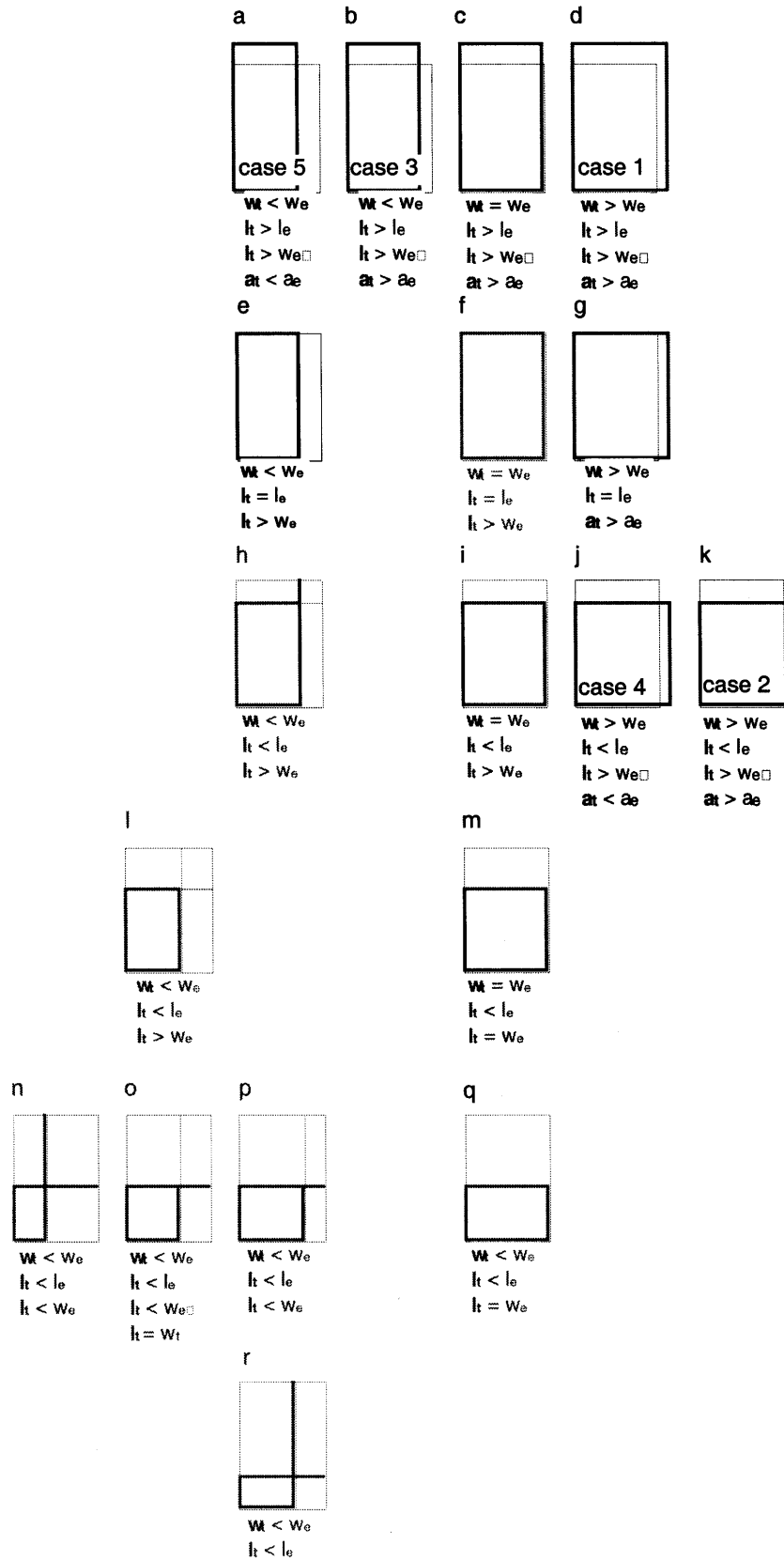


Figure 7.22 - Matching the space to allocate (bold line) with the existing space (thin line).



The space to allocate fits the existing space when the width, length, and area of the space to allocate are smaller than those of the existing space:

$$\begin{aligned} a_t &< a_e, \\ l_t &\leq l_e, \text{ and} \\ w_t &\leq w_e. \end{aligned}$$

The cases in which these conditions are true are illustrated in Figure 7.22e, f, h, and l-r.

The choice of a specific allocation rule depends on the fitting situation.

#### 6. The space to allocate fits into the existing space.

If the both the widths and lengths match (Figure 7.22f), or if the width and length do not match but the difference is so small that the remaining space is not big enough to place one of spaces that are left to allocate, or if there are no spaces left to allocate in the zone, then the assigning rule will be triggered (Rule 35). The difference between the widths or lengths of the space to allocate and the existing space are considered too small if they are half the width or length of the smallest space that remains to be allocated.

If none of the previous conditions holds, then a dissecting rule will be applied. The choice of a specific dissecting rule will be made as follows. If the lengths match (Figure 7.22e), then a dissection perpendicular to the width will take place (Rules 36 and 37). If the widths match (Figure 7.22i, m, and q), then a dissection perpendicular to the length will occur (Rules 38 and 39). If the both the width and the length of the space to allocate are smaller than those of the existing space (Figure 7.22h, i, n, o, p, and r), several situations can occur. If the length of the space to allocate is bigger than the width of the existing space (Figure 7.22i), a dissection perpendicular to the width will occur.

If the condition above does not hold, the rule that makes a dissection perpendicular to the previous level dissection (Rules 40 and 41) will be fired if triggered because it has a higher salience than the other dissecting rules. The higher salience of this rule is justifiable by the analysis of Siza's designs, which showed a clear bias towards the recursive use of this rule. Instead of using a label attached to dissecting line to indicate the last dissection, as in the first version of the grammar, the discursive grammar keeps track of the dissection level that originates the space. The lot's dissection level is zero, the inside and outside zones' dissection level are one, and so on. Naturally, the dissection level can either be an odd or an even number. Even numbers correspond to dissections parallel to the y axis. Such is the case, for instance, of the lot and the sleeping, living, service, and patio zones. Odd numbers correspond to dissections parallel to the x axis.

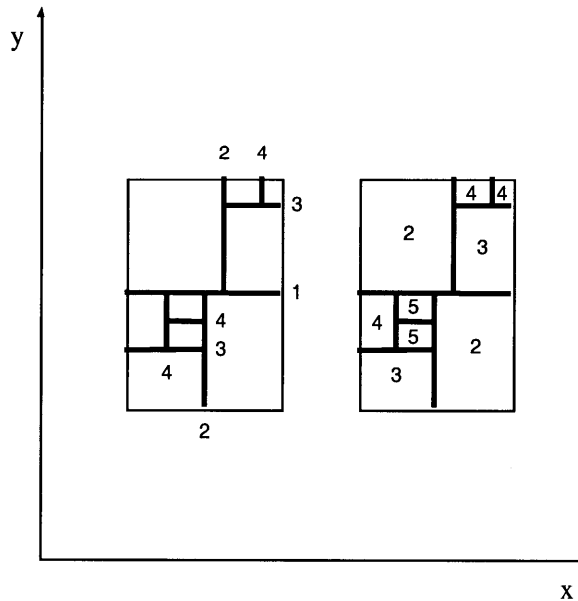


Figure 7.23 - Dissection levels. "Even dissections" are parallel to the y axis whereas "odd dissections" are parallel to the x axis. The dissection level of a space is the level of the dissection that originated it.

If this rule is not triggered, meaning that no space could be allocated using the previous rule, either of the remaining dissecting rules can occur. The choice of rule will depend on the following heuristic

$$h(\text{space}_x, \text{space}_i) = \min(|\text{area}_{\text{space}_x} - \sum \text{area}_{\text{space}_i}|), \text{space}_i \in \text{zone}$$

where  $\text{area}_{\text{space}_x}$  is the area of extra space that results from the allocation, and  $\sum \text{area}_{\text{space}_i}$  is the sum of the areas of the spaces in the zone that are left to allocate. The rule that minimizes the difference between the two areas will be fired.

Once a rule is chosen, it is still necessary to select the transformation under which the rule applies. For each dissection there are eight possible transformations (Figure 7.24). All these possibilities are generated, and then evaluated using the same heuristic used for evaluating the stair pattern. The one that leads to a higher quality solution will be selected to carry on to the following stage of the computation.

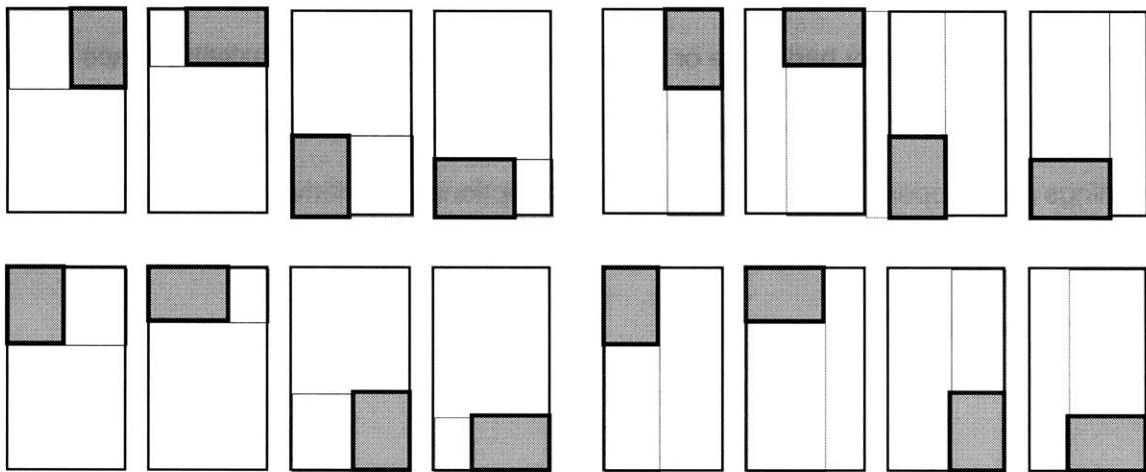


Figure 7.24 - Dissecting rules and transformations. Left columns represent dissections perpendicular to the length, and right columns represent dissections perpendicular to the width. Dissections in the top row are equivalent to dissections on the bottom row.

7. The space to allocate does not fit into the existing space.

If the space to allocate does not fit into the existing space (Figure 7.22a-d, g, j, and k) it is necessary to determine how bad the situation is. If it is not too bad—none of the dimensions of the existing space is smaller than 60% of the dimensions of the space to allocate—the solution is to lower one or more of the dimensions of the space to allocate until it fits into the existing space.

To select the dimension or dimensions to lower, first it is necessary to determine why the space to allocate does not fit into the existing space to find all the dimension-lowering rules that can be applied to the situation. (Figure 7.25) Then, it is necessary to apply a heuristic to choose among these rules the one that keeps the design description closer to the program description. (Figure 7.26) Finally, it is necessary to record that the allocated space is smaller than specified in the housing program. The actual rules are not shown but their format is very similar to the ones used for generating and assessing basic and stair patterns.

If the situation is very bad—one or more of the dimensions of the existing space is smaller than 60% of the corresponding space to allocate dimension, then one of two things can happen. If the space to allocate is optional, the derivation proceeds without allocating the space. If the space is obligatory, the derivation is halted, and failure is announced.

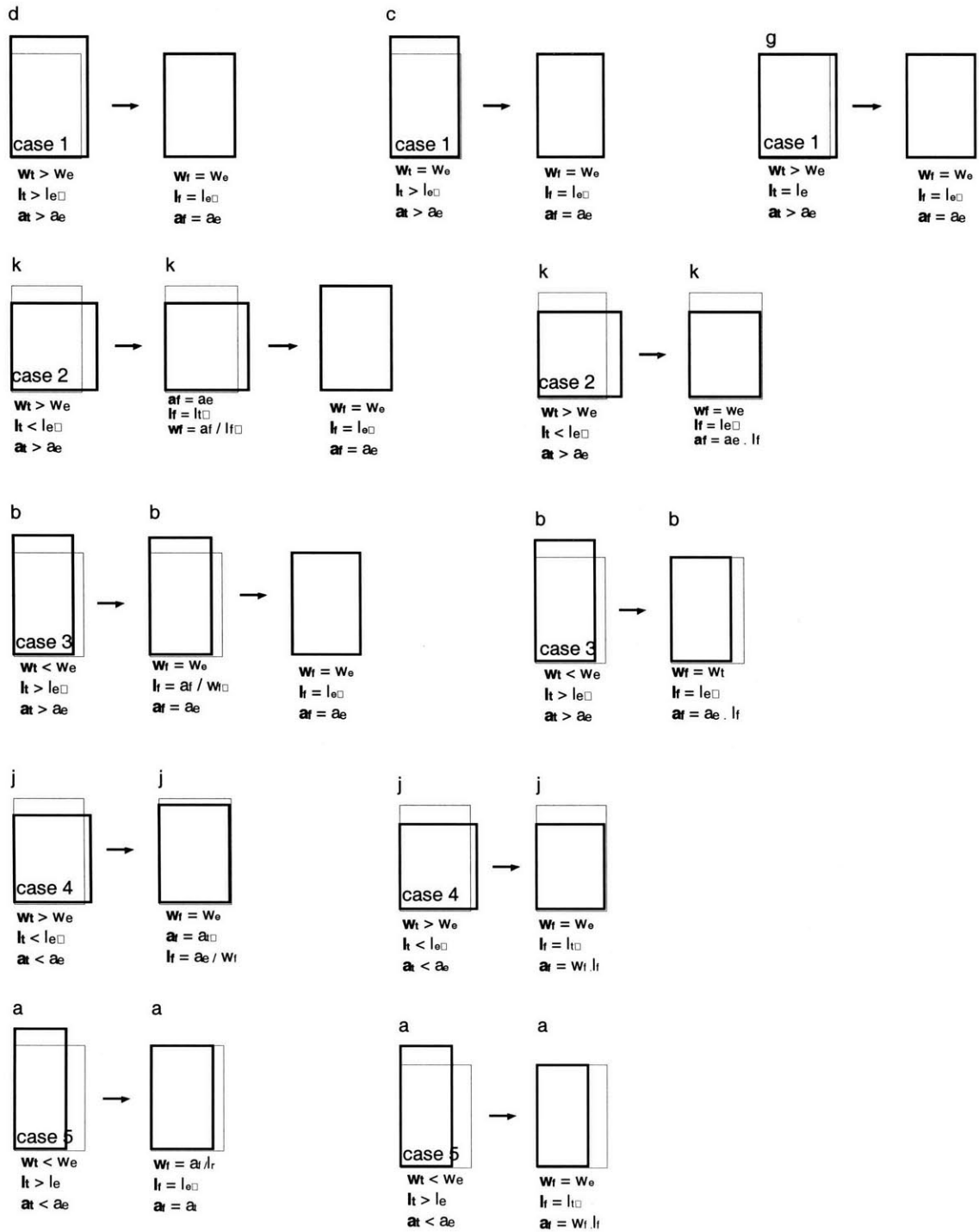


Figure 7.25 – The dimension-lowering rules.

**Heuristic for lowering the requirements:**

lower the requirements in such a way that the distance to the client's requirements is the shortest:  
minimize [space's desired quality level - space's allocated quality level]

Choose the least important requirement among:

- proportion (designer)
- capacity (user)
- spaciousness (user)
- articulation (user)

Can it be lowered?

a. If it cannot chose the next least important requirement

b. If it can, lower it to the next level down

b.1 Calculate the resulting quality level, record it

b.2 Do the changes make the space fit?

If they do, go to a again

If not, lower it to the next level down, and go to b.1 again

c. After choosing all the changeable requirements and calculate the corresponding quality, make the changes that correspond to the smallest difference in quality

Figure 7.26 – The requirement-lowering heuristics.

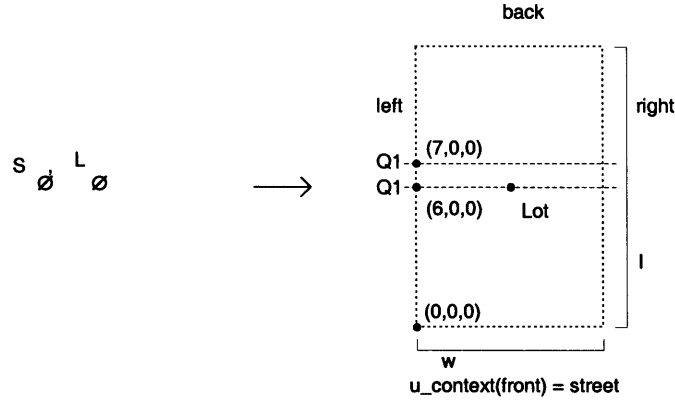
**7.5.3.5 Completing Step 4 and performing Steps 5 and 6**

To complete the generation of the layout it would be necessary to apply rules to concatenate spaces, as well as rules to introduce details and openings, as shown for the initial grammar. These rules were not included in the current revised grammar because they were not essential to illustrate the heuristic search mechanism embedded into the discursive grammar.

**Table 7.38 – The Malagueira designing grammar rules**

**Step 0: Introduce initial shape**

**R0: Introduce initial shape**



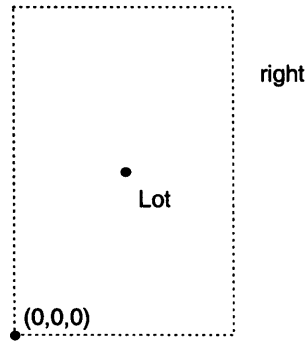
$\alpha_0 \leftarrow \alpha_0$   $l_{lot} = \text{length}(\text{lot}), w_{lot} = \text{width}(\text{lot}), a_{lot} = \text{area}(\text{lot})$   
 $\alpha_1 \leftarrow \alpha_1$   $s_{\text{orientation}_{\text{front}}} = s_{\text{orientation}}(\text{front}), s_{\text{orientation}_{\text{left}}} = s_{\text{orientation}}(\text{left}),$   
 $s_{\text{orientation}_{\text{back}}} = s_{\text{orientation}}(\text{back}), s_{\text{orientation}_{\text{right}}} = s_{\text{orientation}}(\text{right})$   
 $\alpha_2 \leftarrow \alpha_2,$   $u_{\text{context}_{\text{front}}} = u_{\text{context}}(\text{front}), u_{\text{context}_{\text{left}}} = u_{\text{context}}(\text{left}),$   
 $u_{\text{context}_{\text{back}}} = u_{\text{context}}(\text{back}), u_{\text{context}_{\text{right}}} = u_{\text{context}}(\text{right})$   
 $\alpha_{13} \leftarrow \alpha_{13}$   
 $\alpha_{17} \leftarrow \alpha_{17}$   
 $\alpha_{25} \leftarrow \alpha_{25}, R_0 \in \alpha_{25}$

$\delta_1 \leftarrow \delta_1 + \alpha_1$   
 $\delta_2 \leftarrow \delta_2 + \alpha_2$   
 $\delta_{13} \leftarrow \delta_{13} + < [\text{lot}, l_0, (\text{house}), \emptyset, \emptyset, \emptyset, ((0, 0, 0), w_{lot}, l_{lot}, 0.20, a_{lot}),$   
 $[\text{front}, l_1, (\text{street}, s_{\text{orientation}_{\text{front}}}), \emptyset, \emptyset, \emptyset, ((0, -l(\text{lot}), 0), w_{lot}, l_{lot}, 0.20, a_{lot}),$   
 $[\text{left}, l_2, (u_{\text{context}_{\text{left}}}, s_{\text{orientation}_{\text{left}}}), \emptyset, \emptyset, \emptyset, (-w(\text{lot}), 0, 0), w_{lot}, l_{lot}, 0.20, a_{lot}),$   
 $[\text{back}, l_3, (u_{\text{context}_{\text{back}}}, s_{\text{orientation}_{\text{back}}}), \emptyset, \emptyset, \emptyset, ((0, l(\text{lot}), 0), w_{lot}, l_{lot}, 0.20, a_{lot}),$   
 $[\text{right}, l_4, (u_{\text{context}_{\text{right}}}, s_{\text{orientation}_{\text{right}}}), \emptyset, \emptyset, \emptyset, ((w(\text{lot}), 0, 0), w_{lot}, l_{lot}, 0.20, a_{lot})]$   
 $\text{lot}, \text{front}, \text{left}, \text{back}, \text{right} \in \alpha_{13} \wedge \notin \delta_{13}$

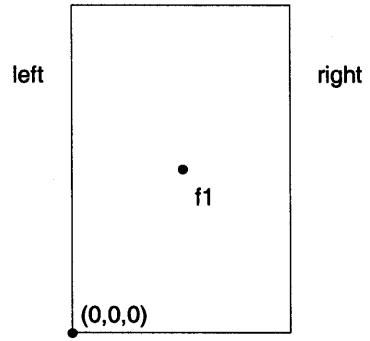
$\delta_{17} \leftarrow \delta_{17} + < (\text{id}_{?space1}, \text{id}_{?space2}, \text{adjacent}), ?space_1 = \text{lot}, ?space_2 \in \{\text{lot}, \text{front}, \text{left}, \text{back}, \text{right}\}$   
 $\alpha_{25} \leftarrow \alpha_{25} + < [R_0, 0] >$

**Step 1: Start**

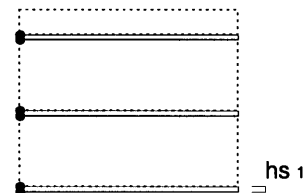
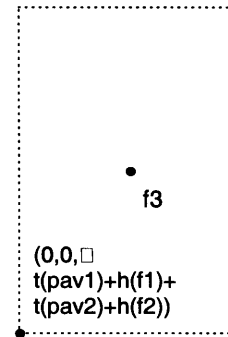
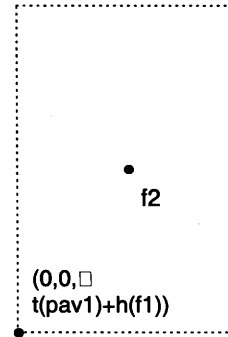
R1: Introduce pavements and floors



$u\_context(front) = street$



$u\_context(front) = street$

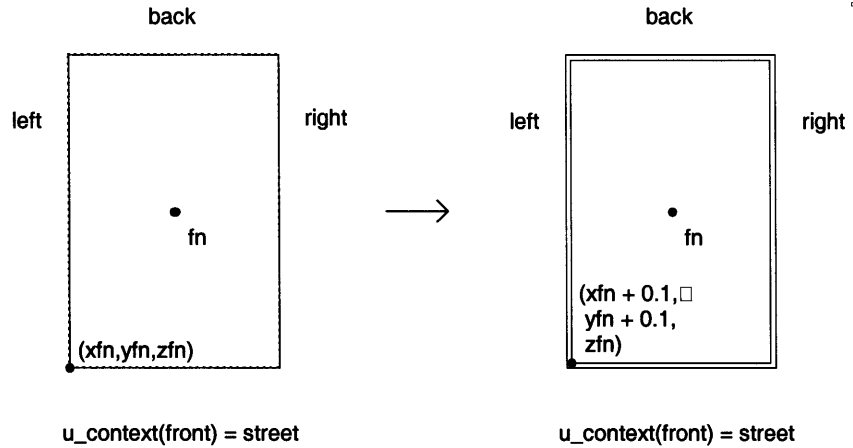




$\alpha_0 \leftarrow \alpha_0$   
 $\alpha_1 \leftarrow \alpha_1 + \alpha_1$   
 $\alpha_2 \leftarrow \alpha_2 + \alpha_2$   
 $\alpha_8 \leftarrow \alpha_8$   
 $\alpha_{13} \leftarrow \alpha_{13}$   
 $\alpha_{17} \leftarrow \alpha_{17}$   
 $\beta_4 \leftarrow \beta_4$

$\delta_0 \leftarrow \delta_0 + \alpha_0$   
 $\delta_1 \leftarrow \delta_1 + \alpha_1$   
 $\delta_2 \leftarrow \delta_2 + \alpha_2$   
 $\delta_8 \leftarrow \delta_8 + \alpha_2$   
 $\delta_{13} \leftarrow \delta_{13} + < [(f1, 106, \emptyset, ((0, 0, h(\text{lot}) + h(\text{pav1}), (0, 0, 0), dx_{\text{lot}}, dy_{\text{lot}}, h(f1), a(\text{lot})),$   
 $[(f2, 107, \emptyset, ((0, 0, h(\text{lot}) + h(\text{pav1}) + h(f1) + h(\text{pav2})), (0, 0, 0), dx_{\text{lot}}, dy_{\text{lot}}, h(f2), a(\text{lot})),$   
 $\alpha_8 = 2 \Rightarrow$   
 $[(f3, 109, \emptyset, ((0, 0, h(\text{lot}) + h(\text{pav1}) + h(f1) + h(\text{pav2}) + h(f2) + h(\text{pav3})), (0, 0, 0), dx_{\text{lot}}, dy_{\text{lot}}, h(f3), a(\text{lot}))]$   
 $f1, f2, f3, \in \alpha_{13} \wedge \notin \delta_{13}$   
 $\delta_{15} \leftarrow \delta_{15} + < \text{available}, (0, 0, 2 \cdot a(\text{lot}), 2 \cdot a(\text{lot})), 1 >$   
 $\delta_{17} \leftarrow \delta_{17} + (\text{id}_{\text{space1}}, \text{id}_{\text{space2}}, \text{adjacent}),$   
 $\text{space}_1 \in \{f1, f2, f3\}, \text{space}_2 \in \{\text{front, left, back, right}\}$   
 $\delta_{21} \leftarrow \delta_{21} + < [\text{pav1}, 301, \emptyset, ((0, 0, 0.0), (0, 0, 0), dx_{\text{lot}}, dy_{\text{lot}}, 0.20, a_{\text{lot}}),$   
 $[\text{pav2}, 302, \emptyset, ((0, 0, t(\text{pav2}) + 0.20), (0, 0, 0), dx_{\text{lot}}, dy_{\text{lot}}, 0.20, a_{\text{lot}}),$   
 $[\text{pav3}, 303, \emptyset, ((0, 0, 2 \cdot t(\text{pav3}) + 0.20), (0, 0, 0), dx_{\text{lot}}, dy_{\text{lot}}, 0.20, a_{\text{lot}}) >$   
 $\delta_{23} \leftarrow \delta_{23}, \text{update\_quality}$   
 $\delta_{24} \leftarrow \delta_{24} + 3 \times \text{pavement\_cost}(a_{\text{lot}}, \text{unit\_cost}(\text{pavement}, \text{material}))$   
 $\delta_{25} \leftarrow \delta_{25} + < [R1, 0] >$

**R2: Enclose floor**



$$\alpha_0 \leftarrow \alpha_0$$

$$\beta_2 \leftarrow \beta_2$$

$$\delta_{13} \leftarrow \delta_{13}, fn \in \delta_{13}, n \in \{1, 2, 3\}$$

$$- [(fn, id_n, \emptyset, ((x, y, z), dx_m, dy_m, dz_m, a_m)],$$

$$+ [(fn, id_3, \emptyset, ((x + 0.10, y + 0.10, z), dx_m, -2 \cdot 0.10, dy_m - 2 \cdot 0.10, dz_m, a_m)]$$

$$\delta_{15} \leftarrow \delta_{15} + < \text{available}, ($$

$$2 \cdot dx_{f1} \cdot 0.10 + 2 \cdot 0.10 \cdot dy_{f1},$$

$$0,$$

$$-(2 \cdot dx_{f1} \cdot 0.10 + 2 \cdot 0.10 \cdot dy_{f1}),$$

$$-(2 \cdot dx_{f1} \cdot 0.10 + 2 \cdot 0.10 \cdot dy_{f1}),$$

$$(A_u - (2 \cdot dx_m \cdot 0.10 + 2 \cdot 0.10 \cdot dy_m)) / (A_g - (2 \cdot dx_m \cdot 0.10 + 2 \cdot 0.10 \cdot dy_m)) >$$

$$\delta_{20} \leftarrow \delta_{20} + < [wall, \max(id) + n, (fn, front), ((x_m, y_m, z_m), dx_m, 0.10, dz_m, dx_m \cdot 0.10),$$

$$[wall, \max(id) + 3 + n, (fn, back), ((x_m, y_m + dy_m - 0.10, z_m), dx_m, 0.10, dz_m, dx_m \cdot 0.10),$$

$$[wall, \max(id) + n, (fn, front), ((x_m, y_m + 0.10, z_m), 0.10, dy_m - 2 \cdot 0.10, dz_m, dy_m \cdot 0.10),$$

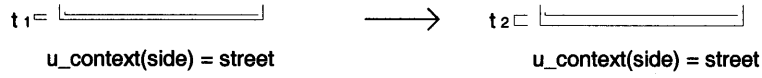
$$[wall, \max(id) + n, (fn, front), ((x_m + dx_m - 0.10, y_m + 0.10, z_m), 0.10, dy_m - 2 \cdot 0.10, dz_m, 0.10 \cdot dy_m)] >$$

$$dx_m = w \text{ (lot)}$$

$$\delta_{24} \leftarrow \delta_{24} + 2 \cdot \text{wall\_cost}(dx_{f1} \cdot 0.10, \text{unit\_cost}(\text{wall}, \text{material})) + 2 \cdot \text{wall\_cost}(0.10 \cdot dy_{f1}, \text{unit\_cost}(\text{wall}, \text{material}))$$

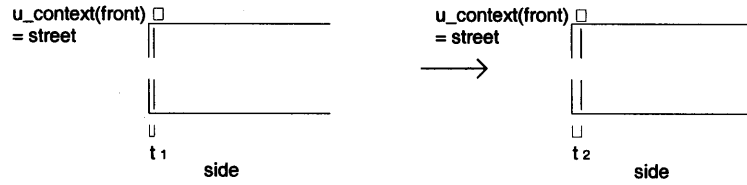
$$\alpha_{25} \leftarrow \alpha_{25} + < [R2, 0] >$$

**R3: Adjust enclosing wall thickness**



$\delta_{20} \leftarrow \delta_{20}, \exists wall \in \{wall \mid wall, adjacent?(street, fn)\}$   
 $- < [wall, id, (street, fn), ((x, y, z), dx, dy, dz, a)] >$   
 If  $dx > dy \wedge y_m > y_{wall} \Rightarrow$   
 $+ < [wall, id, (street, fn), (x, y - 0.10, z), dx, dy + 0.10, dz, a)] >$   
 $w = dy$   
 If  $dx > dy \wedge y_m < y_{wall} \Rightarrow$   
 $+ < [wall, id, (street, fn), (x, y, z), dx, dy + 0.10, dz, a)] >$   
 $w = dy$   
 If  $dx < dy \wedge x_m > x_{wall} \Rightarrow$   
 $+ < [wall, id, (street, fn), (x - 0.10, y, z), dx + 0.10, dy, dz, a)] >$   
 $w = dx$   
 If  $dx < dy \wedge x_m < x_{wall} \Rightarrow$   
 $+ < [wall, id, (street, fn), (x, y, z), dx + 0.10, dy, dz, a)] >$   
 $w = dx$   
 $\delta_{24} \leftarrow \delta_{24} - wall\_cost(a, unit\_cost(wall, w, material))$   
 $+ wall\_cost(a, unit\_cost(wall, w, material))$   
 $\alpha_{25} \leftarrow \alpha_{25} + < [R3, 0] >$

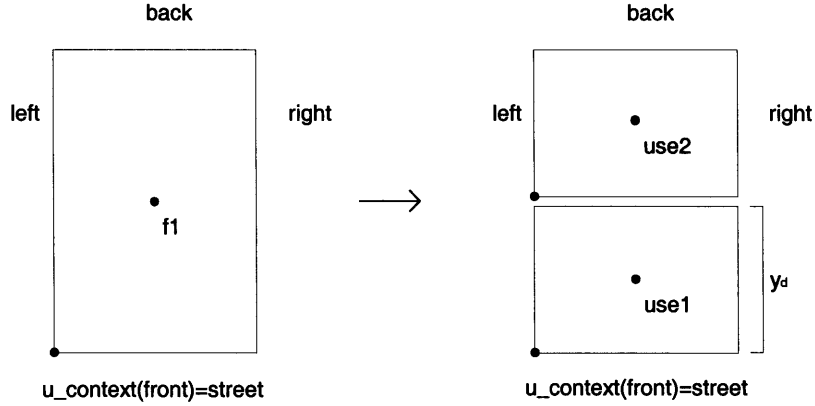
**R4: Adjust enclosing wall thickness**



$\delta_{20} \leftarrow \delta_{20}, \exists wall \in \{wall \mid wall, adjacent?(street, fn)\}$   
 $- < [wall, id, (street, fn), ((x, y, z), dx, dy, dz, a)] >$   
 If  $dx > dy \wedge y_m > y_{wall} \Rightarrow$   
 $+ < [wall, id, (street, fn), (x, y - 0.10, z), dx, dy + 0.10, dz, a)] >$   
 $w = dy$   
 If  $dx > dy \wedge y_m < y_{wall} \Rightarrow$   
 $+ < [wall, id, (street, fn), (x, y, z), dx, dy + 0.10, dz, a)] >$   
 $w = dy$   
 If  $dx < dy \wedge x_m > x_{wall} \Rightarrow$   
 $+ < [wall, id, (street, fn), (x - 0.10, y, z), dx + 0.10, dy, dz, a)] >$   
 $w = dx$   
 If  $dx < dy \wedge x_m < x_{wall} \Rightarrow$   
 $+ < [wall, id, (street, fn), (x, y, z), dx + 0.10, dy, dz, a)] >$   
 $w = dx$   
 $\delta_{24} \leftarrow \delta_{24} - wall\_cost(a, unit\_cost(wall, w, material))$   
 $+ wall\_cost(a, unit\_cost(wall, w, material))$   
 $\alpha_{25} \leftarrow \alpha_{25} + < [R3, 0] >$

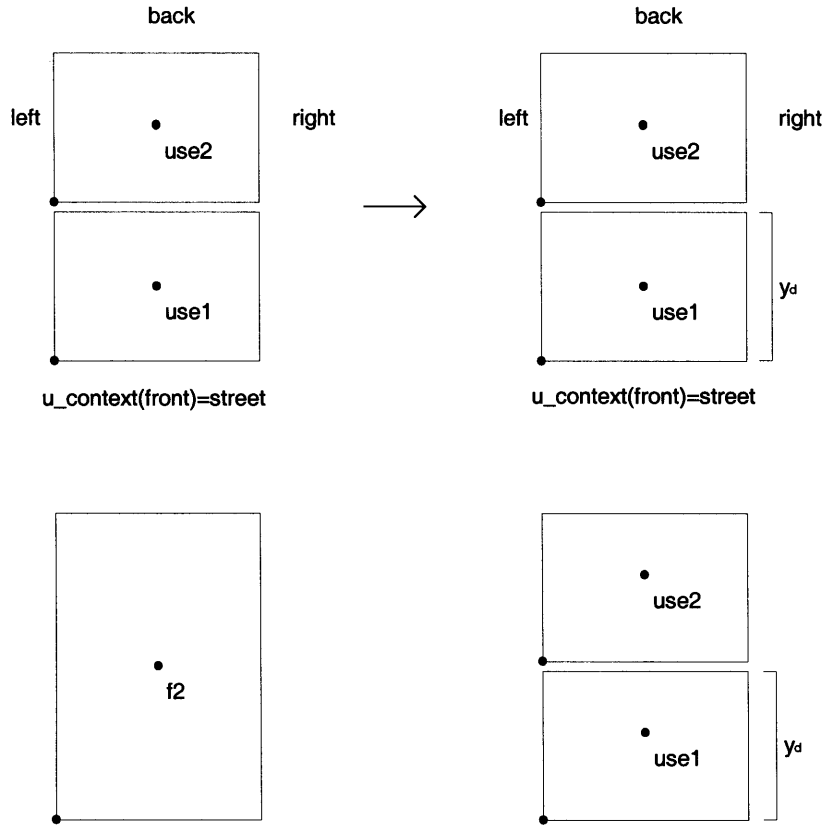
## Step 2: Locate functional zones

### R5: Locate inside/outside zones on the first floor



$$\begin{aligned}
 \alpha_1 &\leftarrow \alpha_1 \\
 \alpha_8 &\leftarrow \alpha_8 \\
 \alpha_9 &\leftarrow \alpha_9, \forall \alpha_1, \alpha_8 = \text{frontyard} \wedge \alpha_9 = \text{true} \\
 &\quad \Rightarrow \text{use1} = \text{outside1} \wedge \text{use2} = \text{inside1} \wedge y_d = 6.00 \wedge a_{in} = a_{use2} \wedge a_{ou} = a_{use1} \\
 \forall \alpha_1, \alpha_8 = \text{frontyard} \wedge \alpha_9 = \text{false} \\
 &\quad \Rightarrow \text{use1} = \text{outside1} \wedge \text{use2} = \text{inside1} \wedge y_d = 7.00 \wedge a_{in} = a_{use2} \wedge a_{ou} = a_{use1} \\
 \alpha_1 &= < \text{street}, ?\text{use}, \text{street}, ?\text{use} >, \forall ?\text{use} \wedge \alpha_8 = \text{backyard}, \forall \alpha_9 \\
 &\quad \Rightarrow \text{use1} = \text{inside1} \wedge \text{use2} = \text{outside1} \wedge y_d = 7.00 \wedge a_{in} = a_{use1} \wedge a_{ou} = a_{use2} \\
 \alpha_1 &= < \text{street}, ?\text{use}, \text{street}, ?\text{use} >, \forall ?\text{use} \wedge \alpha_8 = \text{backyard}, \alpha_9 = \text{true} \\
 &\quad \Rightarrow \text{use1} = \text{outside1} \wedge \text{use2} = \text{inside1} \wedge y_d = 6.00 \wedge a_{in} = a_{use1} \wedge a_{ou} = a_{use2} \\
 \alpha_1 &= < \text{street}, ?\text{use}, \text{street}, ?\text{use} >, \forall ?\text{use} \wedge \alpha_8 = \text{backyard}, \alpha_9 = \text{false} \\
 &\quad \Rightarrow \text{use1} = \text{outside1} \wedge \text{use2} = \text{inside1} \wedge y_d = 5.00 \wedge a_{in} = a_{use1} \wedge a_{ou} = a_{use2} \\
 \delta_{13} &\leftarrow \delta_{13} - < [(f1, id_{f1}, \emptyset, ((x_{f1}, y_{f1}, z_{f1}), dx_{f1}, dy_{f1}, dz_{f1}, a_{f1})) > \\
 &\quad + < [(use1, id_{use1}, \emptyset, ((x_{use1}, y_{use1}, z_{use1}), dx_{use1}, dy_{use1} - (dx_{f1} - y_d + 2 \cdot 0.10), dz_{use1}, dx_{use1} \cdot dy_{use1} - (f1_{dy} - y_d + 2 \cdot 0.10)), \\
 &\quad \quad [(use2, \max(id) + 1, \emptyset, ((x_{use2}, y_{use2} + y_d, z_{use2}), dx_{use2}, dy_{use2} - y_d, dz_{use2}, dx_{use2} \cdot dy_{use2} - y_d))] > \\
 \delta_{15} &\leftarrow \delta_{15} + < \text{available}, (f1_{ax} \cdot 0.20, a_{in}, -(a_{in} + f1_{ax} \cdot 0.20), -f1_{ax} \cdot 0.20), -A_u / A_g + A_u - f1_{ax} \cdot 0.20 / A_g > \\
 \delta_{17} &\leftarrow \delta_{17} - < [id_{f1}, id_{?space}, \text{adjacent}], ?\text{space} \in \{\text{front}, \text{left}, \text{back}, \text{right}\} \\
 &\quad + < [id_{inside1}, id_{?left}, \text{adjacent}], \\
 &\quad \quad [id_{inside1}, id_{?right}, \text{adjacent}], \\
 &\quad \quad [id_{outside1}, id_{?left}, \text{adjacent}], \\
 &\quad \quad [id_{outside1}, id_{?right}, \text{adjacent}], \\
 &\quad \quad [id_{inside1}, id_{?space1}, \text{adjacent}] \\
 &\quad \quad [id_{outside1}, id_{?space2}, \text{adjacent}] \\
 &\quad \quad \alpha_8 = \text{frontyard} \Rightarrow ?\text{space}_1 = \text{back} \wedge ?\text{space}_2 = \text{front} \\
 &\quad \quad \alpha_8 = \text{backyard} \Rightarrow ?\text{space}_1 = \text{front} \wedge ?\text{space}_2 = \text{back} \\
 \delta_{20} &\leftarrow \delta_{20} + < [\text{wall}, \max(id) + 1, (\text{inside}, \text{outside}), ((x_{f1}, y_d - 0.10, z_{f1}), dx_{f1}, 0.20, dz_{f1}, dx_{f1} \cdot dz_{f1}) > \\
 \delta_{24} &\leftarrow \delta_{24} + \text{wall\_cost}(dx_{f1} \cdot dz_{f1}, \text{unit\_cost}(\text{wall}, 0.20, \text{material})) \\
 \alpha_{25} &\leftarrow \alpha_{25} + < [R4, 0] >
 \end{aligned}$$

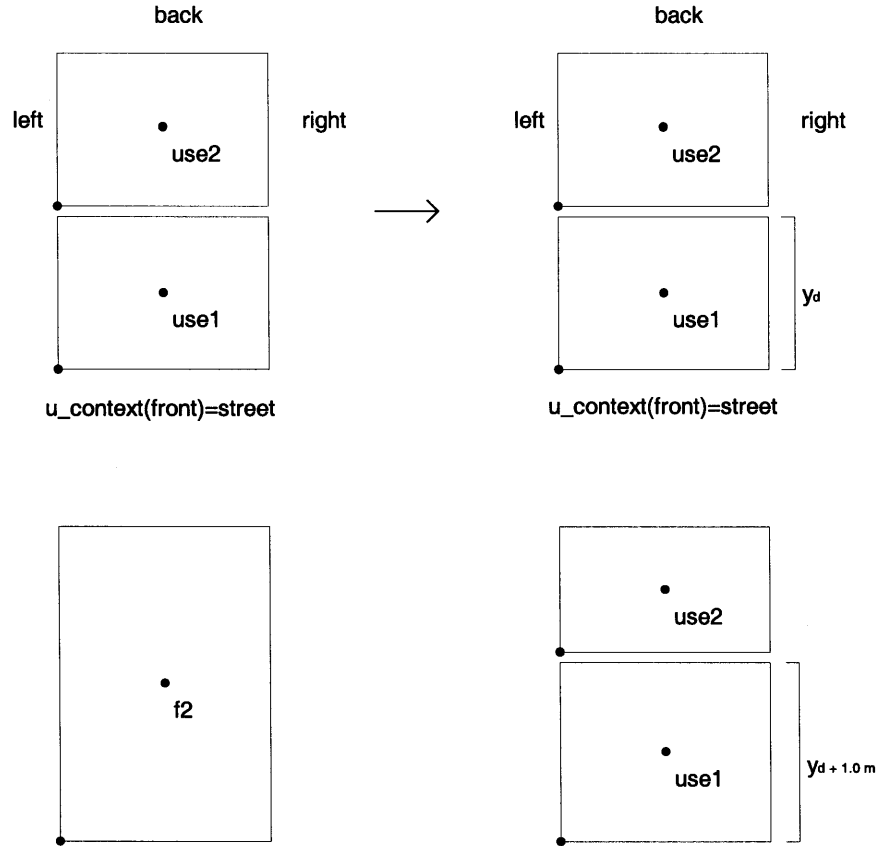
**R6: Locate inside/outside zones on the second floor** (if the yard is at the front and there are no balconies or yard is at the back)



$\exists (use_n, f1, on) \Rightarrow use1, use2 \in \{inside1, outside1\}$   
 $\exists (use_n, f2, on) \Rightarrow use1, use2 \in \{inside2, outside2\}$

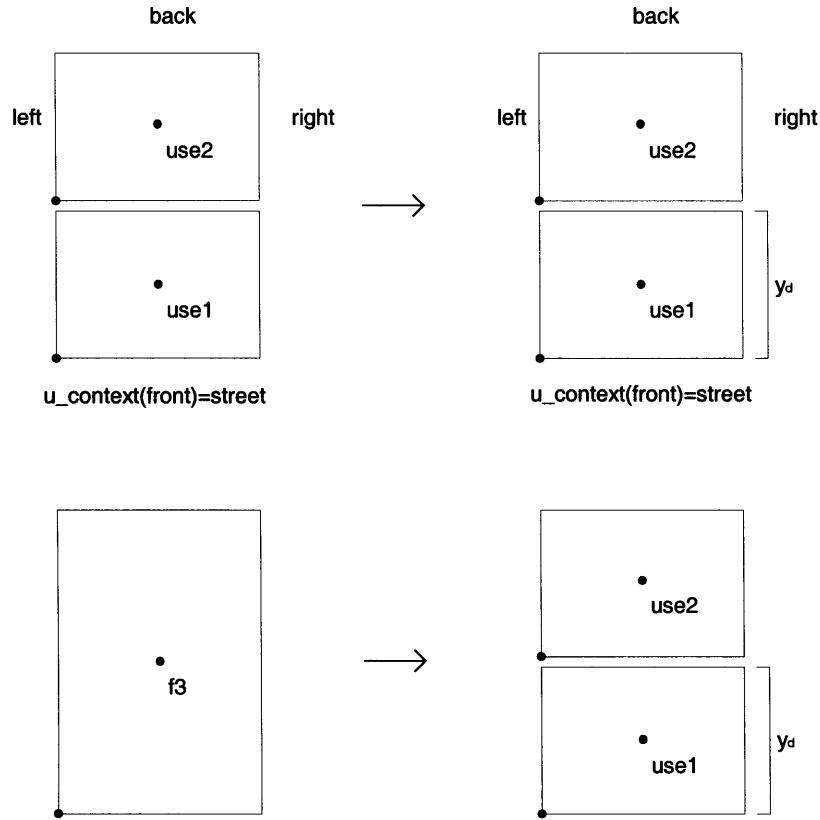
$\alpha_8 \leftarrow \alpha_8$   
 $\alpha_9 \leftarrow \alpha_9, (\alpha_8 = \text{frontyard} \wedge \alpha_9 = \text{false}) \vee \alpha_8 = \text{backyard}$   
 $\delta_{13} \leftarrow \delta_{13}, \exists [inside1, id_{in}, \emptyset, ((x_{in}, y_{in}, z_{in}), dx_{in}, dy_{in}, dz_{in}, a_{in})] > \wedge$   
 $[outside1, id_{ou}, \emptyset, ((x_{ou}, y_{ou}, z_{ou}), dx_{ou}, dy_{ou}, dz_{ou}, a_{ou})] > \in \delta_{13}$   
 $- < [f2, id_{f2}, \emptyset, ((x_{f2}, y_{f2}, z_{f2}), dx_{f2}, dy_{f2}, dz_{f2}, a_{f2})] >$   
 $+ < [(inside2, id_{i2}, \emptyset, ((x_{in}, y_{in}, z_{i2}), dx_{in}, dy_{in}, dz_{i2}, a_{in}),$   
 $[(outside2, \max(id) + 1, \emptyset, ((x_{ou}, y_{ou}, z_{i2}), dx_{ou}, dy_{ou}, dz_{i2}, a_{ou})] >$   
 $\delta_{15} \leftarrow \delta_{15} + < \text{available}, (dx_{i2} \cdot 0.20, a_{in}, -(a_{in} + dx_{i2} \cdot 0.20), -dx_{i2} \cdot 0.20), -A_u / A_g + A_u - dx_{i2} \cdot 0.20 / A_g >$   
 $\delta_{17} \leftarrow \delta_{17}, \exists [id_{inside1}, id_{?space}, adjacent] \wedge [id_{outside1}, id_{?space}, adjacent] \in \delta_{17}, \forall ?space$   
 $- < [id_{i2}, id_{?space1}, adjacent], ?space1 \in \{\text{front}, \text{left}, \text{back}, \text{right}\}$   
 $+ < [id_{inside2}, id_{?space}, adjacent],$   
 $[id_{outside2}, id_{?space}, adjacent] >$   
 $\delta_{20} \leftarrow \delta_{20}, \exists < [wall, id_w, (inside1, outside1), ((x_w, y_w, z_w), dx_w, dy_w, dz_w, a_w)] >$   
 $+ < [wall, \max(id) + 1, (inside2, outside2), ((x_w, y_w, z_{i2}), dx_w, dy_w, dz_{i2}, dx_w \cdot dy_w)] >$   
 $\delta_{24} \leftarrow \delta_{24} + \text{wall\_cost}(dx_w \cdot dz_{i2}, \text{unit\_cost}(\text{wall}, 0.20, \text{material}))$   
 $\alpha_{25} \leftarrow \alpha_{25} + < [R5.1, 0] >$

R7: Locate inside/outside zones on the second floor (if yard is at the front and there are balconies)



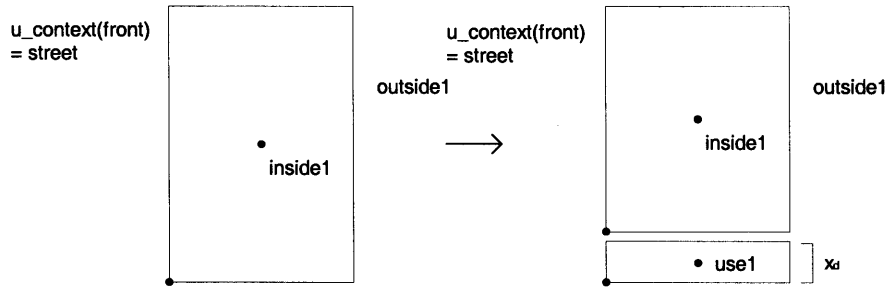
$\alpha_8 \leftarrow \alpha_8$   
 $\alpha_9 \leftarrow \alpha_9, \alpha_8 = \text{frontyard} \wedge \alpha_9 = \text{true}$   
 $\delta_{13} \leftarrow \delta_{13}, \exists [\text{inside1}, \text{id}_{in}, \emptyset, ((x_{in}, y_{in}, z_{in}), dx_{in}, dy_{in}, dz_{in}, a_{in})] > \wedge$   
 $[\text{outside1}, \text{id}_{ou}, \emptyset, ((x_{ou}, y_{ou}, z_{ou}), dx_{ou}, dy_{ou}, dz_{ou}, a_{ou})] > \in \delta_{13}$   
 $- < [f2, \text{id}_{f2}, \emptyset, ((x_{f2}, y_{f2}, z_{f2}), dx_{f2}, dy_{f2}, dz_{f2}, a_{f2})] >$   
 $+ < [(\text{inside2}, \text{id}_{i2}, \emptyset, ((x_{in}, y_{in} + 1.0, z_{i2}), dx_{in}, dy_{in} - 1.0, dz_{i2}, a_{in}),$   
 $[(\text{outside2}, \max(\text{id}) + 1, \emptyset, ((x_{ou}, y_{ou}, z_{i2}), dx_{ou}, dy_{ou} + 1.0, dz_{i2}, a_{ou})] >$   
 $\delta_{15} \leftarrow \delta_{15} + < \text{available}, (dx_{i2} \cdot 0.20, a_{in}, - (a_{in} + dx_{i2} \cdot 0.20), - dx_{i2} \cdot 0.20), - A_u / A_g + A_u \cdot dx_{i2} \cdot 0.20 / A_g >$   
 $\delta_{17} \leftarrow \delta_{17}, \exists [\text{inside1}, ?\text{space}, \text{adjacent}] \wedge [\text{outside1}, ?\text{space}, \text{adjacent}] \in \delta_{17}, \forall ?\text{space}$   
 $- < [\text{id}_{i2}, \text{id}_{?space1}, \text{adjacent}], ?\text{space1} \in \{\text{front}, \text{left}, \text{back}, \text{right}\}$   
 $+ < [\text{id}_{\text{inside2}}, \text{id}_{?space}, \text{adjacent}],$   
 $[\text{id}_{\text{outside2}}, \text{id}_{?space}, \text{adjacent}] >$   
 $\delta_{20} \leftarrow \delta_{20}, \exists < [\text{wall}, \text{id}_w, (\text{inside1}, \text{outside1}), ((x_w, y_w, z_w), dx_w, dy_w, dz_w, a_w)] >$   
 $+ < [\text{wall}, \max(\text{id}) + 1, (\text{inside2}, \text{outside2}), ((x_w, y_w + 1.00, z_{i2}), dx_w, dy_w, dz_w, dx_w \cdot dy_w)] >$   
 $\delta_{24} \leftarrow \delta_{24} + \text{wall\_cost}(dx_w \cdot dz_{i2}, \text{unit\_cost}(\text{wall}, 0.20, \text{material}))$   
 $\alpha_{25} \leftarrow \alpha_{25} + < [R5.2, 0] >$

**R8: Locate inside/outside zones on the third floor**



$$\begin{aligned}
 \delta_{13} &\leftarrow \delta_{13}, \exists [\text{inside2}, \text{id}_{in}, \emptyset, ((x_{in}, y_{in}, z_{in}), dx_{in}, dy_{in}, dz_{in}, a_{in})] > \wedge \\
 &[\text{outside2}, \text{id}_{ou}, \emptyset, ((x_{ou}, y_{ou}, z_{ou}), dx_{ou}, dy_{ou}, dz_{ou}, a_{ou})] > \in \delta_{13} \\
 &- < [f3, \text{id}_{f3}, \emptyset, ((x_{f3}, y_{f3}, z_{f3}), dx_{f3}, dy_{f3}, dz_{f3}, a_{f3})] > \\
 &+ < [(\text{inside3}, \text{id}_{f3}, \emptyset, ((x_{in}, y_{in}, z_{f3}), dx_{in}, dy_{in}, dz_{f3}, a_{in}), \\
 &[\text{outside3}, \max(\text{id}) + 1, \emptyset, ((x_{ou}, y_{ou}, z_{f3}), dx_{ou}, dy_{ou}, dz_{f3}, a_{ou})] > \\
 \delta_{17} &\leftarrow \delta_{17}, \exists [\text{inside2}, ?\text{space}, \text{adjacent}] \wedge [\text{outside2}, ?\text{space}, \text{adjacent}] \in \delta_{17}, \forall ?\text{space} \\
 &- < [f3, ?\text{space1}, \text{adjacent}], ?\text{space1} \in \{\text{front}, \text{left}, \text{back}, \text{right}\} \\
 &+ < [\text{inside3}, ?\text{space}, \text{adjacent}], \\
 &[\text{outside3}, ?\text{space}, \text{adjacent}] > \\
 \delta_{20} &\leftarrow \delta_{20}, \exists < [\text{wall}, \text{id}_w, (\text{inside1}, \text{outside1}), ((x_w, y_w + 1.00, z_w), dx_w, dy_w, dz_w, a_w)] > \\
 &+ < [\text{wall}, \max(\text{id}) + 1, (\text{inside3}, \text{outside3}), ((x_w, y_w, z_w), dx_w, dy_w, dz_w, dx_w \cdot dy_w)] > \\
 \delta_{24} &\leftarrow \delta_{24} + \text{wall\_cost}(dx_w, dz_{f3}, \text{unit\_cost}(\text{wall}, 0.20, \text{material})) \\
 \alpha_{25} &\leftarrow \alpha_{25} + < [R6, 0] >
 \end{aligned}$$

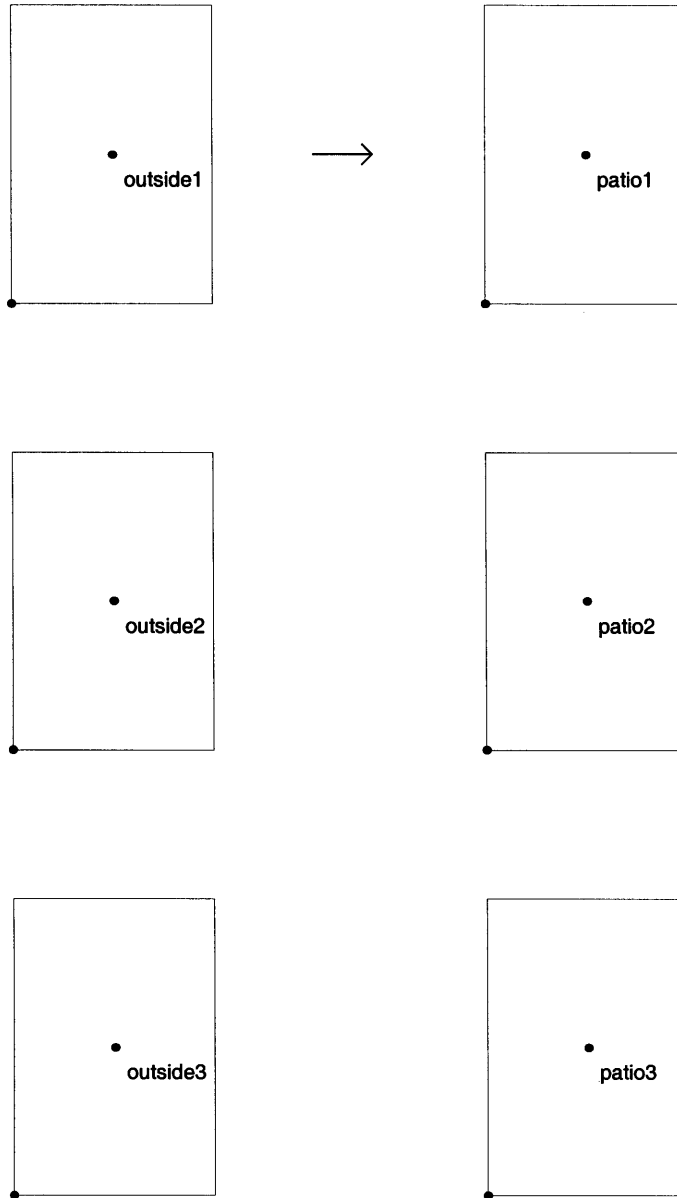
**R9: Locate the backyard corridor on the first floor**



$$\begin{aligned}
 \delta_{13} &\leftarrow \delta_{13}, \exists < [?neighbor, id_{house}, (house, ?orientation), ((x_h, y_h, z_h), dx_h, dy_h, dz_h, a_h)], ?neighbor \in \{left, right\} \\
 &\quad [inside1, id_{in}, \emptyset, ((x_{in}, y_{in}, z_{in}), dx_{in}, dy_{in}, dz_{in}, a_{in})] \in \delta_{13} \wedge [id_{inside}, id_{house}, adjacent] \in \delta_{17} \\
 &\quad dx_h > dx_{in} \Rightarrow x_{in} = x_{in} \wedge x_{co} = x_{in} + (dx_{in} - 1.10) \wedge x_{wall} = x_{in} + (dx_{in} - 1.30) \\
 &\quad dx_h < dx_{in} \Rightarrow x_{in} = x_{in} + 1.10 + 0.20 \wedge x_{co} = x_{in} + 1.30 \wedge x_{wall} = x_{in} + 1.30 \\
 &\quad - < [inside1, id_{in}, \emptyset, ((x_{in}, y_{in}, z_{in}), dx_{in}, dy_{in}, dz_{in}, a_{in})] > \\
 &\quad + < [inside1, id_{in}, \emptyset, ((x_{in}, y_{in}, z_{in}), dx_{in} - 1.30, dy_{in}, dz_{in}, (dx_{in} - 1.30) \cdot dy_{in})], \\
 &\quad \quad [corridor, \max(id) + 1, \emptyset, ((x_{co}, y_{in}, z_{in}), 1.10, dy_{in}, dz_{in}, 1.10 \cdot dy_{in})] > \\
 \delta_{15} &\leftarrow \delta_{15} + < available, (0, 0, -1.10 \cdot dy_{in}, -1.10 \cdot dy_{in}), 0 > \\
 \delta_{16} &\leftarrow \delta_{16} + < used, (0, 0, 1.10 \cdot dy_{in}, 1.10 \cdot dy_{in}), 0 > \\
 \delta_{17} &\leftarrow \delta_{17}, \forall < [id_{outside}, ?id_{space}, adjacent] > \Rightarrow space \neq street \\
 &\quad - < [id_{inside}, id_{house}, adjacent] >, \\
 &\quad + < [id_{inside}, id_{corridor}, adjacent], \\
 &\quad \quad [id_{outside}, id_{corridor}, adjacent], \\
 &\quad \quad [id_{corridor}, id_{house}, adjacent], \\
 &\quad \quad [id_{corridor}, id_{street}, adjacent] >, \\
 \delta_{20} &\leftarrow \delta_{20} + < [wall, \max(id) + 1, (inside1, corridor), ((x_{co}, y_{in}, z_{in}), 0.20, dy_{in}, dz_{in}, 0.20 \cdot dy_{in})] > \\
 \delta_{24} &\leftarrow \delta_{24} + wall\_cost(dx_w, dz_{f3}, unit\_cost(wall, 0.20, material)) \\
 \alpha_{23} &\leftarrow \alpha_{23} + < [R7, 0] >
 \end{aligned}$$



**R10: Locate the patio zone through assignment on all floors**



$\alpha_5 \leftarrow \alpha_5$

$\alpha_8 \leftarrow \alpha_8, \alpha_8 = 2 \wedge \alpha_5 = 2$

$\delta_{13} \leftarrow \delta_{13}, \exists [\text{outside1}, \text{id}_{\text{outside}}, \emptyset, ((x_{\text{ou}}, y_{\text{ou}}, z_{\text{ou}}), dx_{\text{ou}}, dy_{\text{ou}}, dz_{\text{ou}}, a_{\text{ou}})] \in \delta_{13},$

$\exists [\text{outside2}, \text{id}_{\text{outside}}, \emptyset, ((x_{\text{ou}}, y_{\text{ou}}, z_{\text{ou}}), dx_{\text{ou}}, dy_{\text{ou}}, dz_{\text{ou}}, a_{\text{ou}})] \in \delta_{13},$

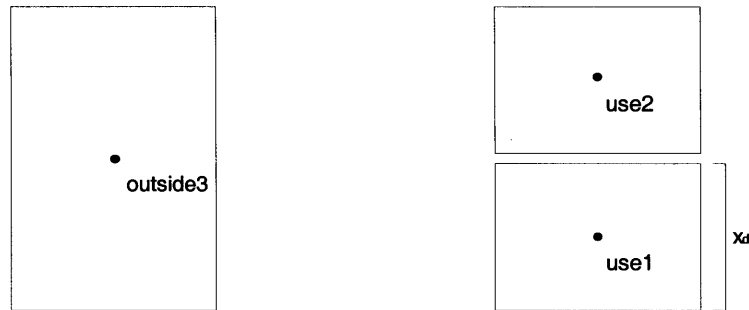
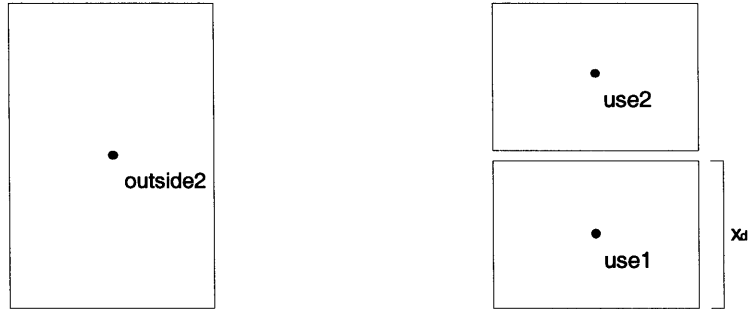
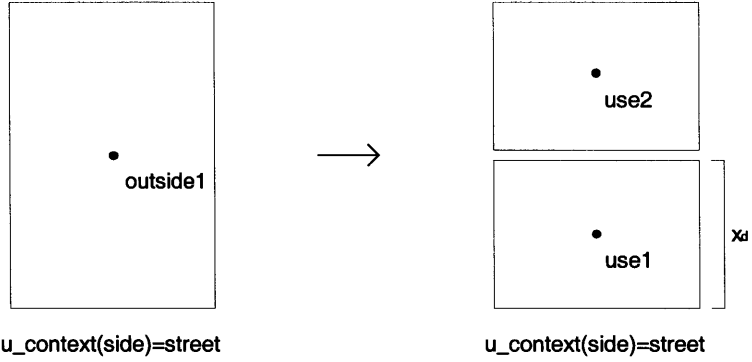
$\exists [\text{outside3}, \text{id}_{\text{outside}}, \emptyset, ((x_{\text{ou}}, y_{\text{ou}}, z_{\text{ou}}), dx_{\text{ou}}, dy_{\text{ou}}, dz_{\text{ou}}, a_{\text{ou}})] \in \delta_{13},$

$\delta_{14} \leftarrow \delta_{14} + \langle \text{patio1}, \emptyset, a_{\text{ou}} \rangle$

$\delta_{17} \leftarrow \delta_{17}$

$\text{change\_name}(\text{outside}_n, \text{patio}_n), n \in \{1, 2, 3\}$

**R11: Locate the patio zone through a dissection perpendicular to its width on all floors**



$\alpha_{13} \leftarrow \alpha_{13}, a_{pa} = a \text{ (patio)}$   
 $\delta_{13} \leftarrow \delta_{13}, yard \notin \delta_{13}$   
 $(\exists [\text{outside1}, id_{\text{outside}}, \emptyset, ((x_{ou}, y_{ou}, z_{ou}), dx_{ou}, dy_{ou}, dz_{ou}, a_{ou})] > \in \delta_{13} \wedge$   
 $\exists [?side, id_{\text{street}}, (\text{street}, ?orientation), ((x_s, y_s, z_s), dx_s, dy_s, dz_s, a_s)] \in \delta_{13},$   
 $?side \in \{\text{left}, \text{right}\}$   
 $\Rightarrow p(t=0) = 0.9 \wedge p(t=1) = 0.1)$   
 $(\exists [\text{outside1}, id_{\text{outside}}, \emptyset, ((x_{ou}, y_{ou}, z_{ou}), dx_{ou}, dy_{ou}, dz_{ou}, a_{ou})] > \in \delta_{13} \wedge$   
 $\sim \exists [?side, id_{\text{street}}, (\text{street}, ?orientation), ((x_s, y_s, z_s), dx_s, dy_s, dz_s, a_s)]$   
 $\in \delta_{13}, ?side \in \{\text{left}, \text{right}\})$

$$\Rightarrow p(t = 0) = 0.5 \wedge p(t = 1) = 0.5$$

$t = 0 \Rightarrow \text{use2} = \text{patio}, \text{use1} = \text{anyzone}$

$t = 1 \Rightarrow \text{use1} = \text{patio}, \text{use2} = \text{anyzone}$

$$x_{u1} = x_{ou} + a_{u2} / dy_{ou} + 0.20 \wedge x_{u2} = x_{ou} \wedge x_{wall} = x_{ou} + a_{u2} / dy_{ou}$$

- < [outside1, id<sub>ou</sub>, ∅, ((x<sub>ou</sub>, y<sub>ou</sub>, z<sub>ou</sub>), dx<sub>ou</sub>, dy<sub>ou</sub>, dz<sub>ou</sub>, a<sub>ou</sub>)] >

+ < [use1, id<sub>u1</sub>, ∅, ((x<sub>u1</sub>, y<sub>ou</sub>, z<sub>ou</sub>), a<sub>u1</sub> / dy<sub>ou</sub>, dy<sub>ou</sub>, dz<sub>ou</sub>, a<sub>u1</sub>),

[use2, id<sub>u2</sub>, ∅, ((x<sub>ou</sub>, y<sub>ou</sub>, z<sub>ou</sub>), dx<sub>ou</sub> - (a<sub>u1</sub> / dy<sub>ou</sub>), dy<sub>ou</sub>, dz<sub>ou</sub>, a<sub>ou</sub> - a<sub>u1</sub>)] >

$$\delta_{14} \leftarrow \delta_{14} + \text{< patio, } \emptyset, a(\text{patio}) \text{>}$$

$$\delta_{17} \leftarrow \delta_{17} - \text{< [id}_{\text{outside}}, \text{id}_{\text{?space}}, \text{adjacent}] \text{>, } \forall \text{ space}$$

+ < [id<sub>use1</sub>, id<sub>use2</sub>, adjacent], [id<sub>use1</sub>, id<sub>in</sub>, adjacent], [id<sub>use2</sub>, id<sub>in</sub>, adjacent],

[id<sub>use1</sub>, id<sub>right</sub>, adjacent], [id<sub>use2</sub>, id<sub>left</sub>, adjacent] >

∃ [id<sub>outside</sub>, id<sub>front</sub>, adjacent] ⇒ + < [id<sub>use1</sub>, id<sub>front</sub>, adjacent], [id<sub>use2</sub>, id<sub>front</sub>, adjacent] >

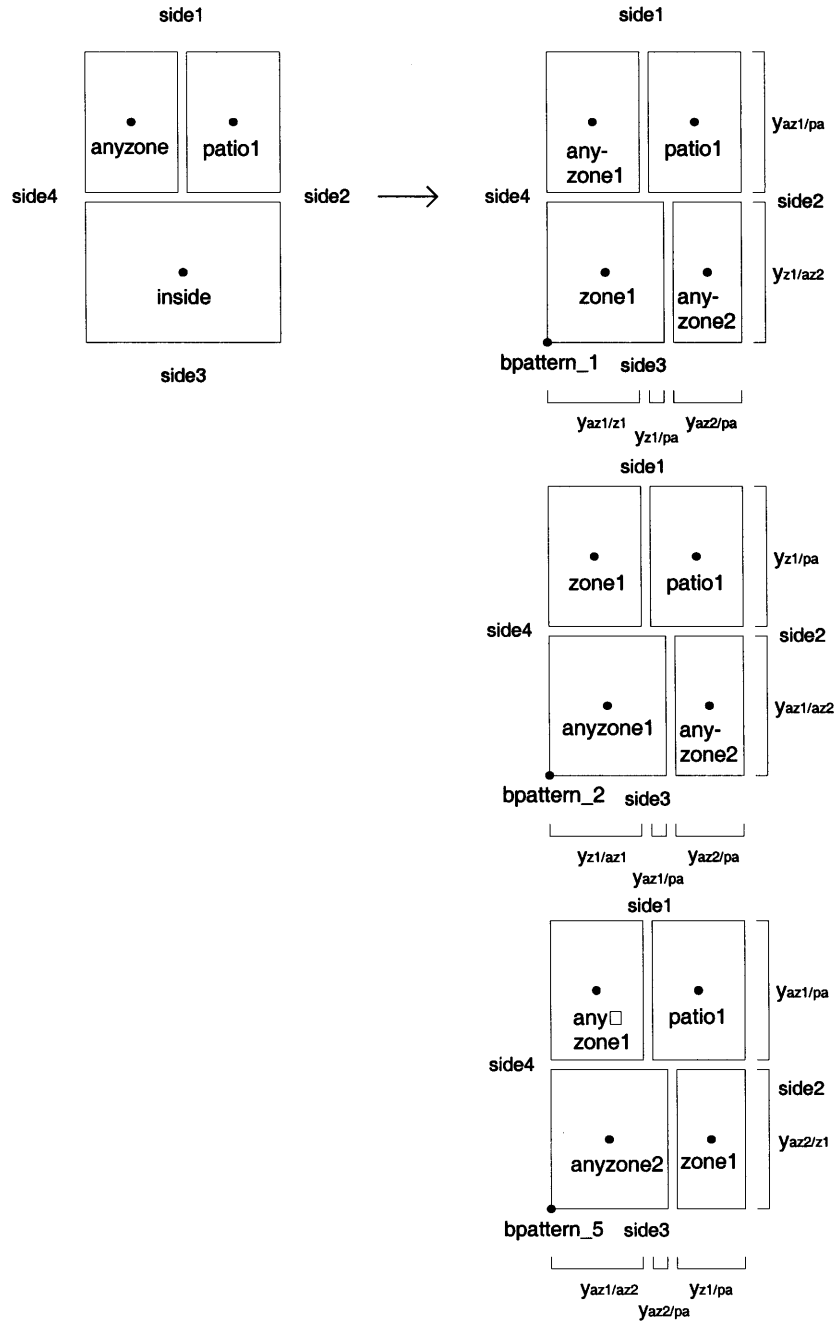
∃ [id<sub>outside</sub>, id<sub>back</sub>, adjacent] ⇒ + < [id<sub>use1</sub>, id<sub>back</sub>, adjacent], [id<sub>use2</sub>, id<sub>back</sub>, adjacent] >

$$\delta_{20} \leftarrow \delta_{20} + \text{< [wall, max(id) + 1, (yard, ?zone), ((x}_{\text{ou}}, \text{y}_{\text{ou}}, \text{z}_{\text{ou}}), 0.20, \text{d}_{\text{y}_{\text{ou}}}, \text{d}_{\text{z}_{\text{ou}}}, 0.20 \cdot \text{d}_{\text{y}_{\text{ou}}}) \text{>}$$

$$\delta_{24} \leftarrow \delta_{24} + \text{wall\_cost (d}_{\text{y}_{\text{ou}}} \cdot \text{d}_{\text{z}_{\text{ou}}}, \text{unit\_cost (wall, 0.20, material))}$$

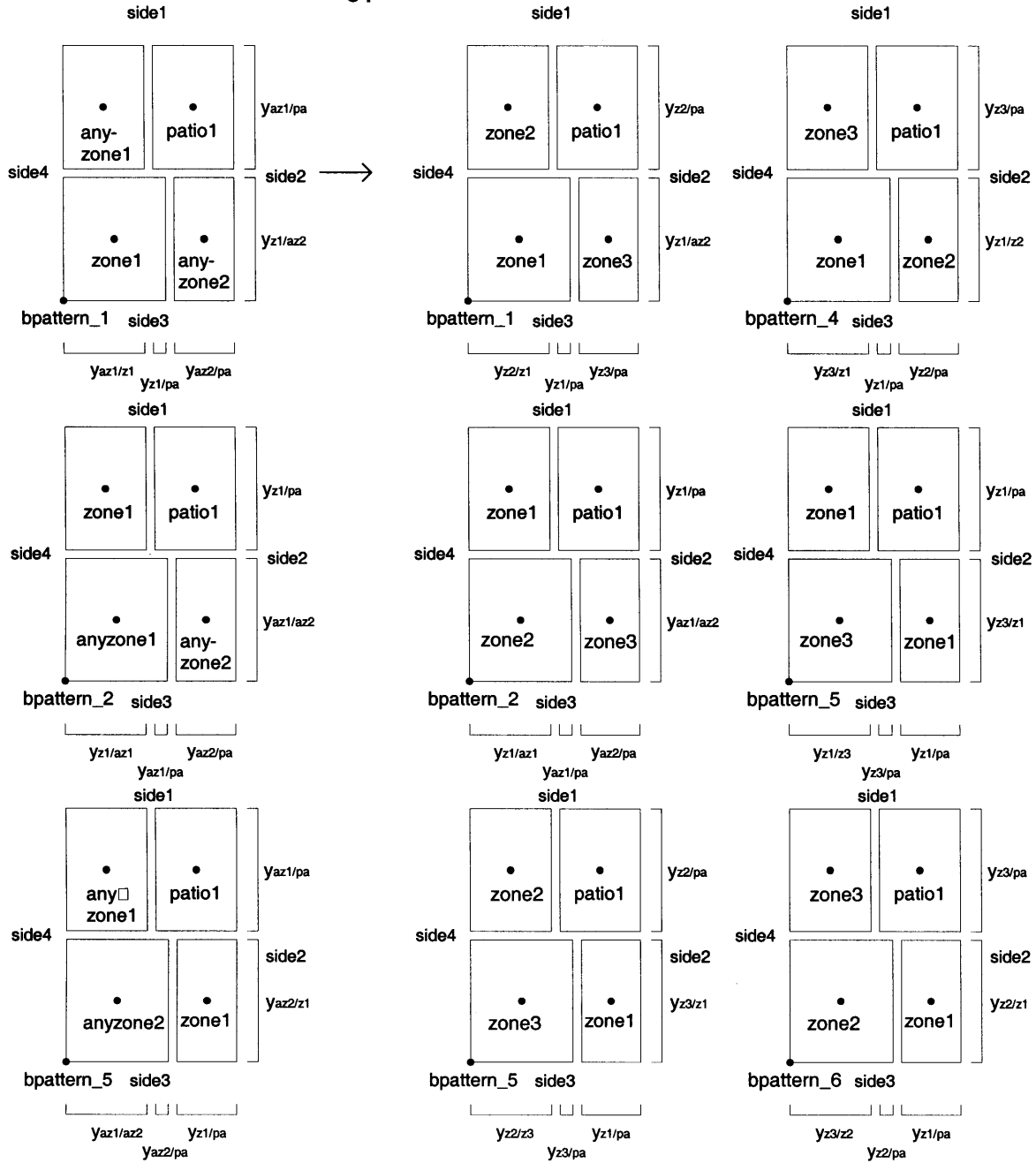
$$\delta_{25} \leftarrow \delta_{25} + \text{< R8, t >}$$

**R12: Locate on the first floor the most constrained of the remaining zones in all possible locations on the first floor**



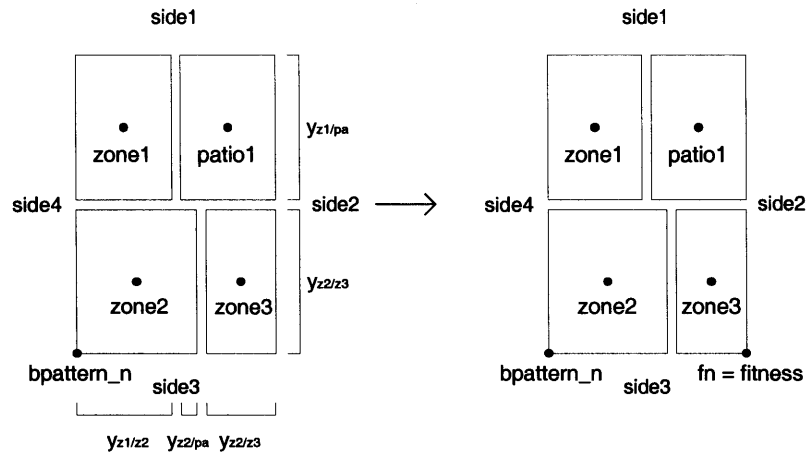
Description part of the rule not shown

**R13: Locate the remaining patterns on the first floor**



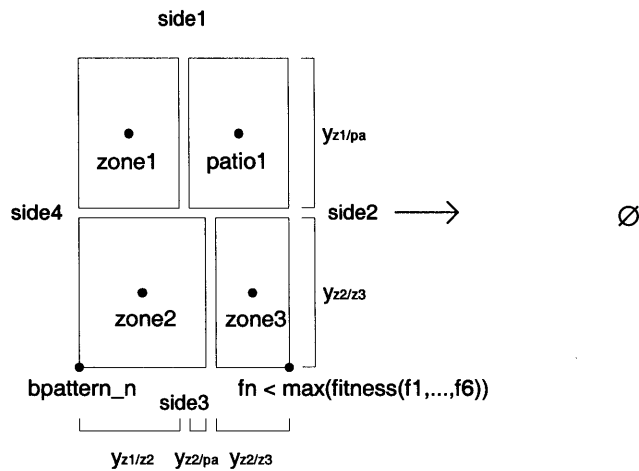
Description part of the rule not shown

**R14: Assess a basic pattern**



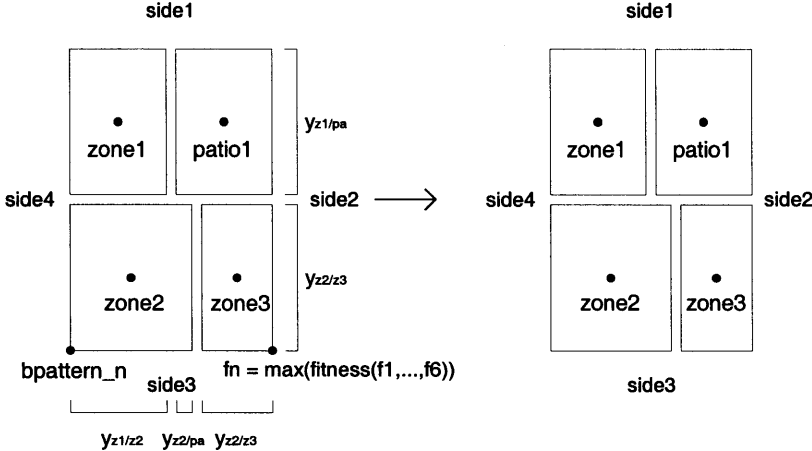
zone1, zone2, zone3 ∈ {living, sleeping, service}.  
 number of assessed patterns = number of assessed basic patterns + 1  
 number of existing basic patterns = number of existing basic patterns + 1  
*Note: the remaining part of the description is not shown.*

**R15: Eliminate a bad basic pattern**



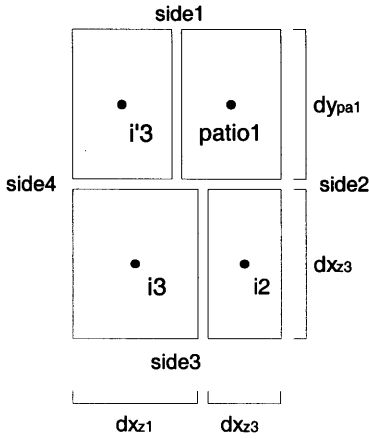
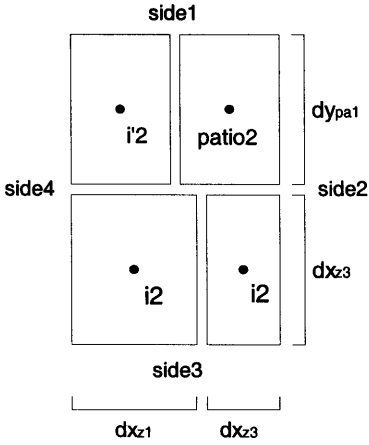
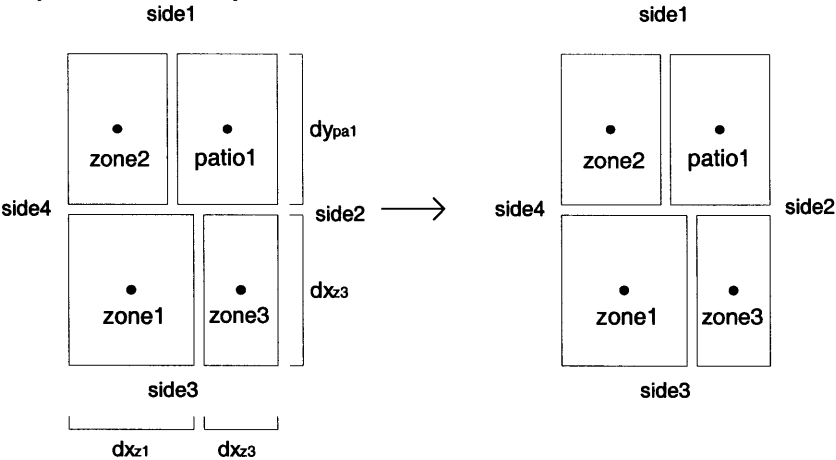
number of assessed patterns = 6.  
 number of existing basic patterns > 1  
 number of existing basic patterns = number of existing basic patterns - 1  
*Note: the remaining part of the description is not shown.*

**R16: Select the best basic pattern**



number of existing basic patterns = number of existing basic patterns - 1  
 number of existing basic patterns = 1  
 Note: the remaining part of the description is not shown.

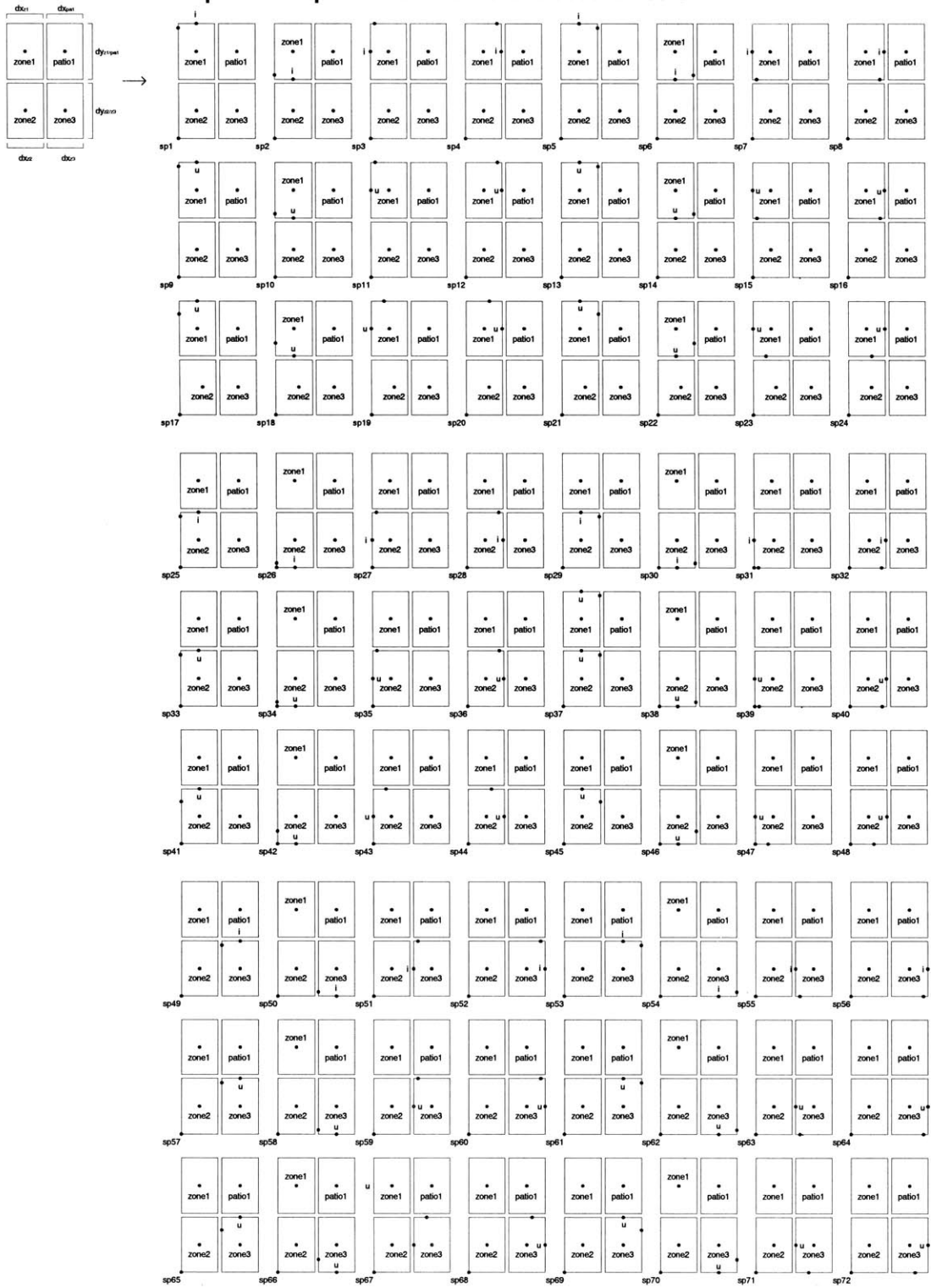
**R17: Replicate the basic pattern on the second and third floors**



*Note: the description part of the rule is not shown.*

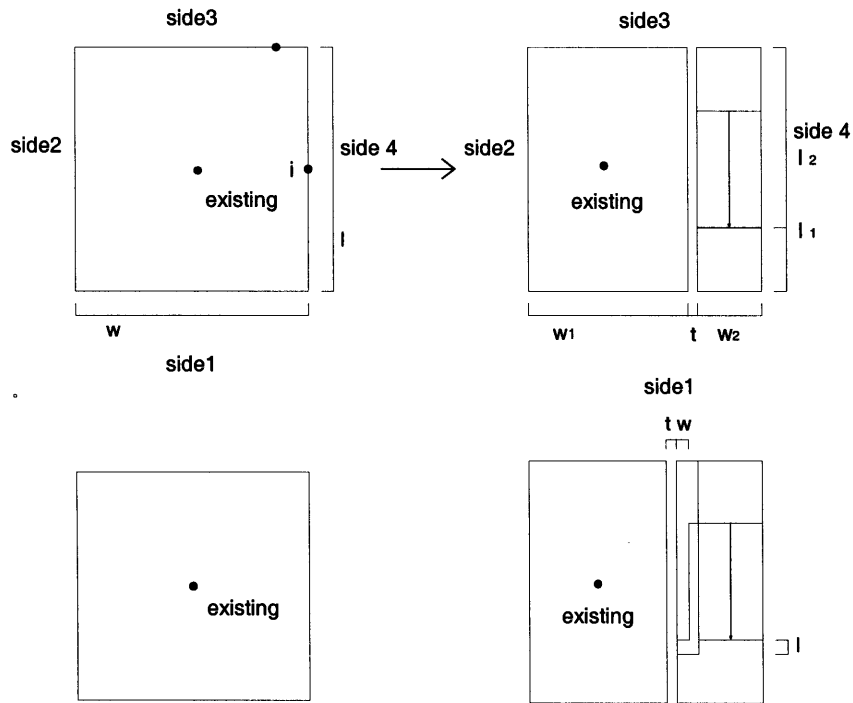


### R18: Generate proto-stair patterns on the first and second floors



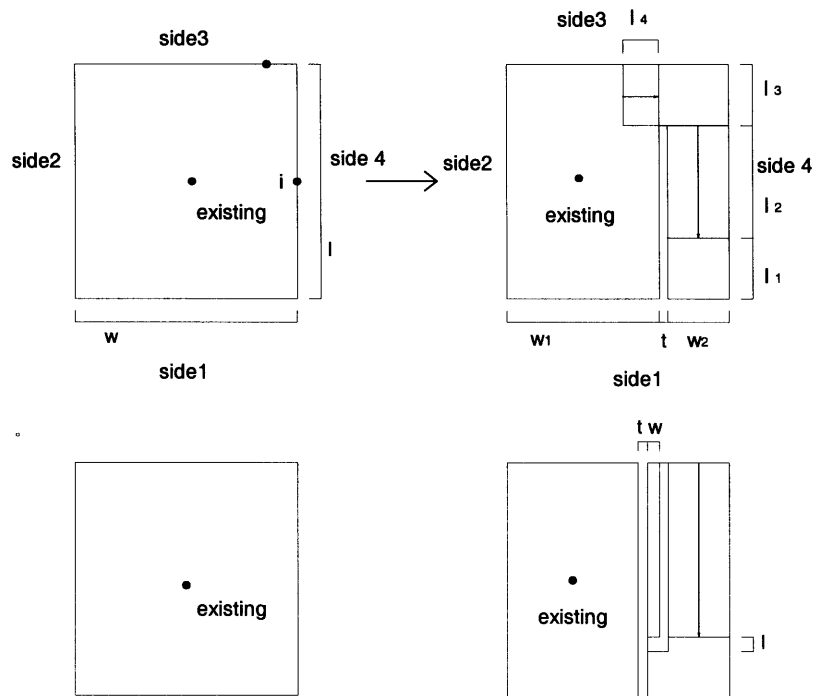


**R19: Locate I-shaped staircase for pattern assessment**



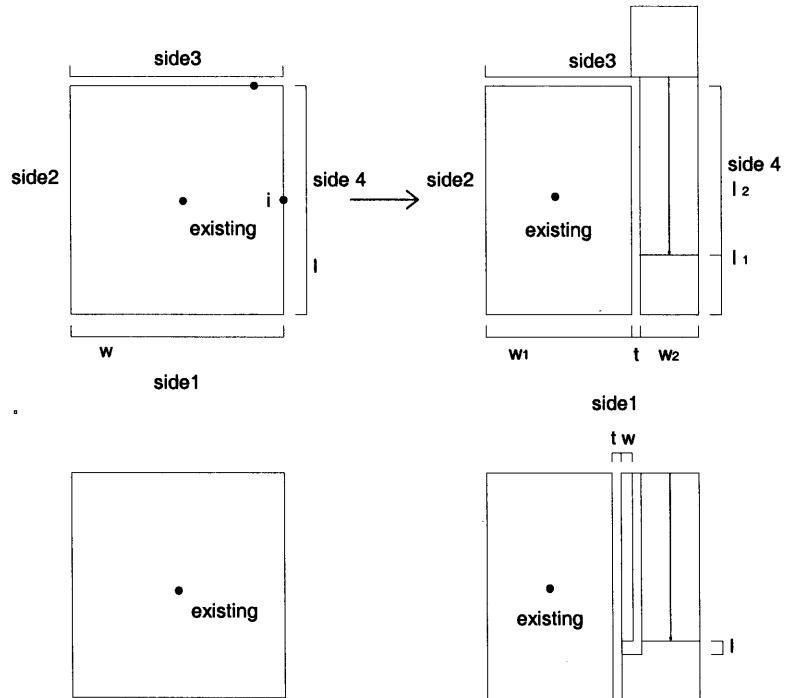
*Note: the description part of the rule is not shown.*

**R20: Locate L-shaped staircase for pattern assessment**



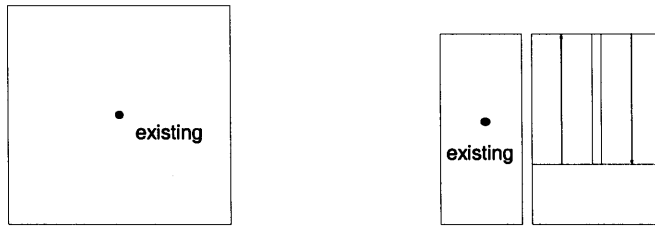
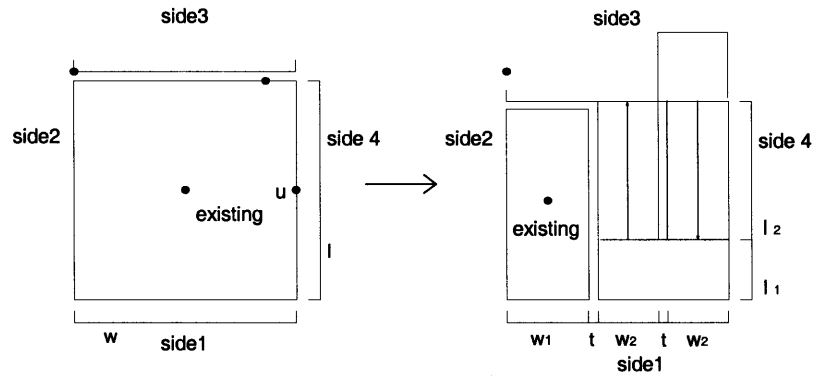
*Note: the description part of the rule is not shown.*

**R21: Locate I-shaped staircase for pattern assessment**



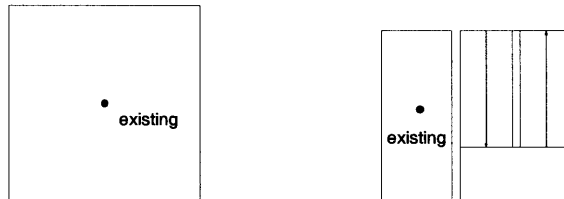
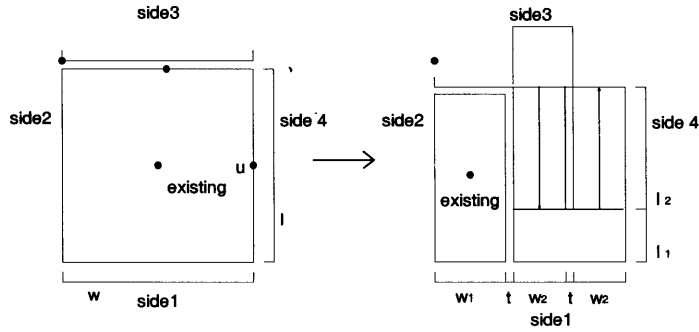
*Note: the description part of the rule is not shown.*

**R22: Locate U-shaped staircase for pattern assessment**



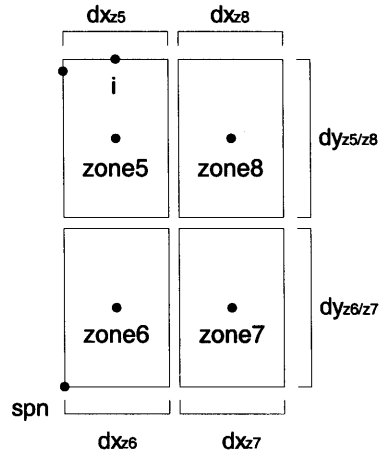
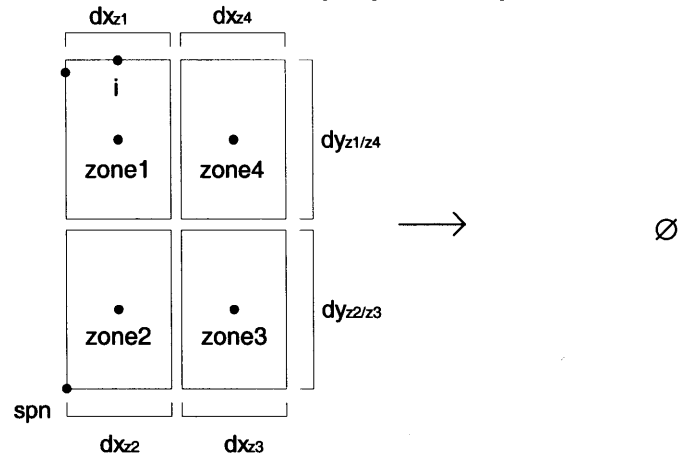
*Note: the description part of the rule is not shown.*

**R23: Locate reflected U-shaped staircase for pattern assessment**



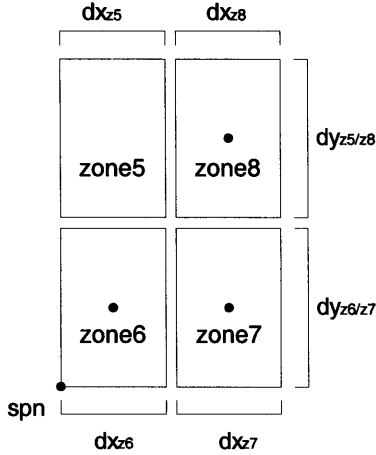
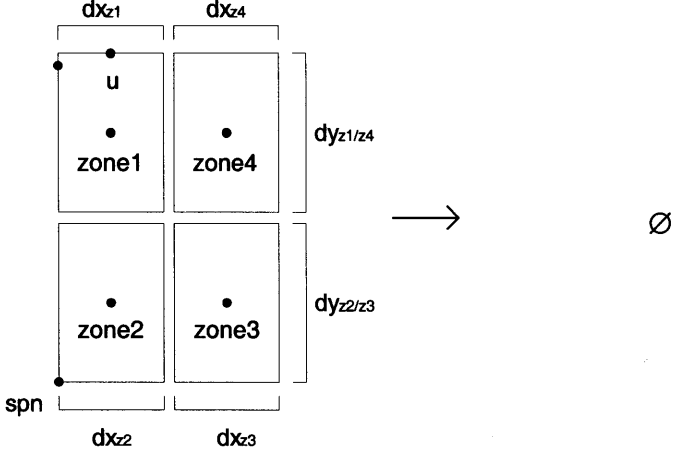
*Note: the description part of the rule is not shown.*

**R24: Eliminate failed I and L-shaped proto-stair pattern**



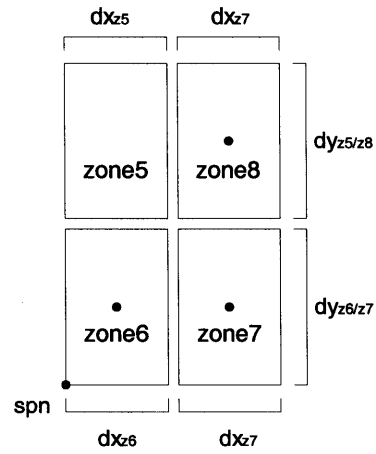
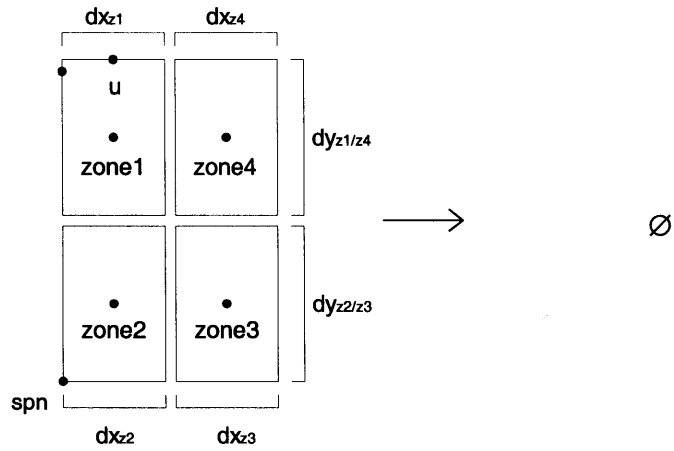
*Note: the description part of the rule is not shown.*

**R25: Eliminate failed U-shaped proto-stair pattern**



*Note: the description part of the rule is not shown.*

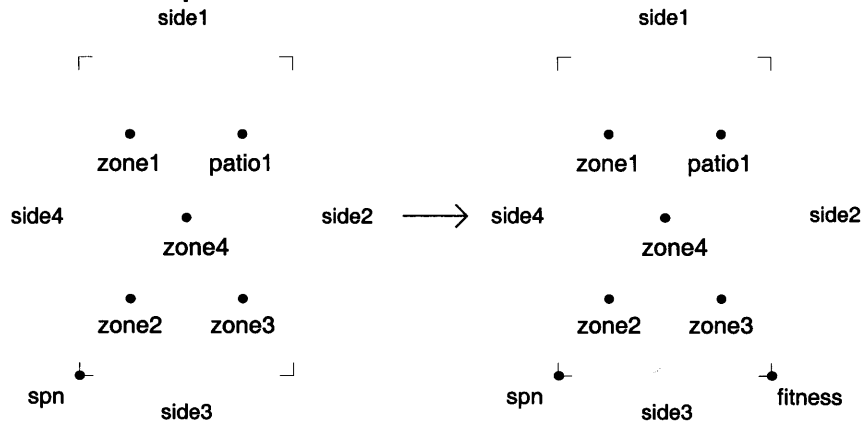
**R26: Eliminate reflected U-shaped proto-stair pattern**



*Note: the description part of the rule is not shown.*

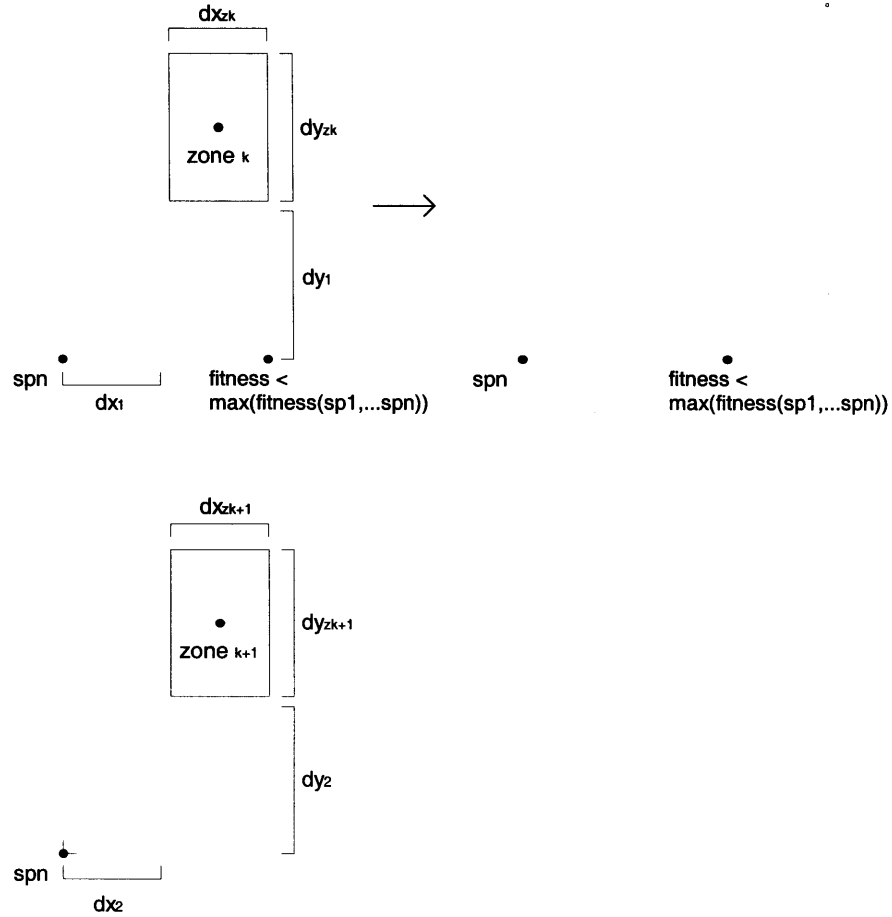


**R27: Assess a stair pattern**



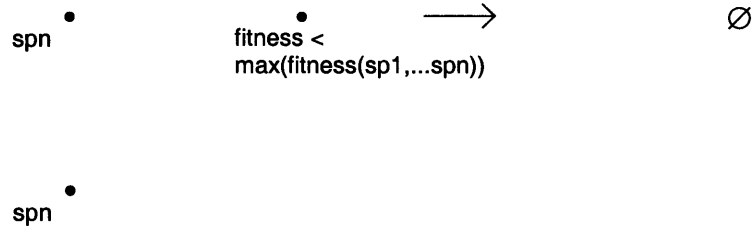
*Note: the description part of the rule is not shown.*

**R28: Eliminate a zone of one of the worst stair patterns**



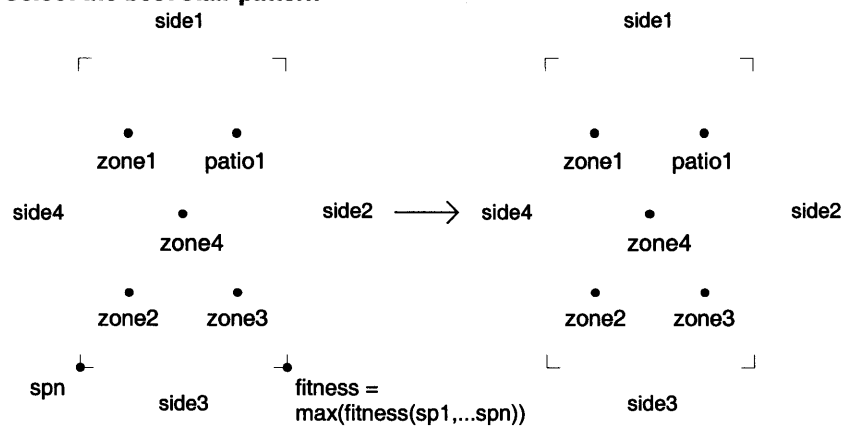
*Note: the description part of the rule is not shown.*

**R29: Eliminate the markers of one of the worst stair patterns**



*Note: the description part of the rule is not shown.*

**R30: Select the best stair pattern**

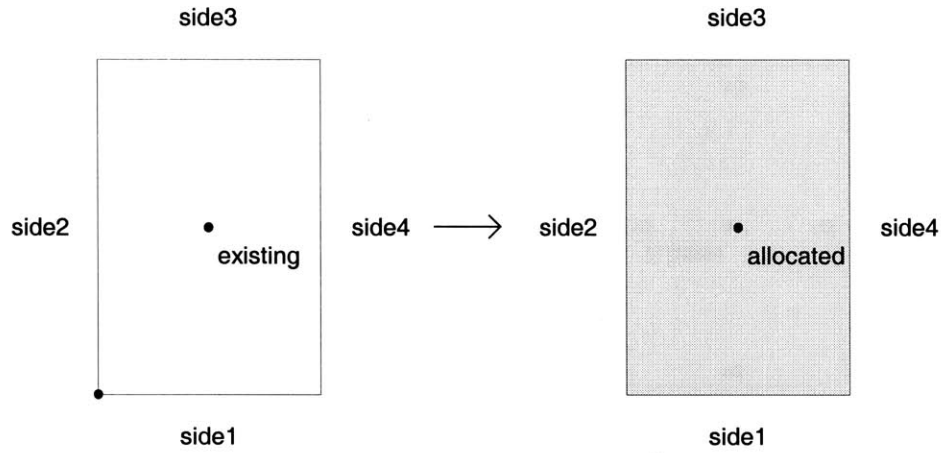


*Note: the description part of the rule is not shown.*

**R31-34: Create circulation scheme**

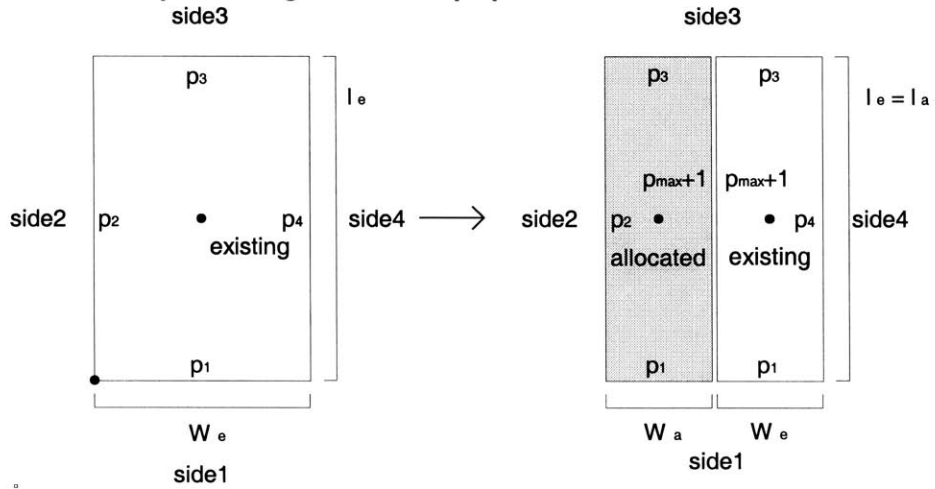
*Note: Rules 122-125 of the initial grammar apply. The description part of the rule is not shown.*

**R35: Allocate space through assignment**



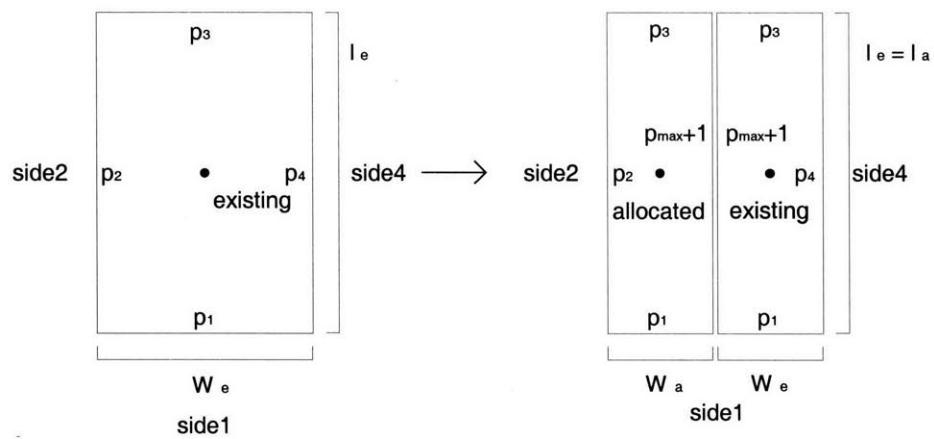
*Note: the description part of the rule is not shown. Please see conditions for rule application below.*

**R36: Allocate space through dissection perpendicular to the width**



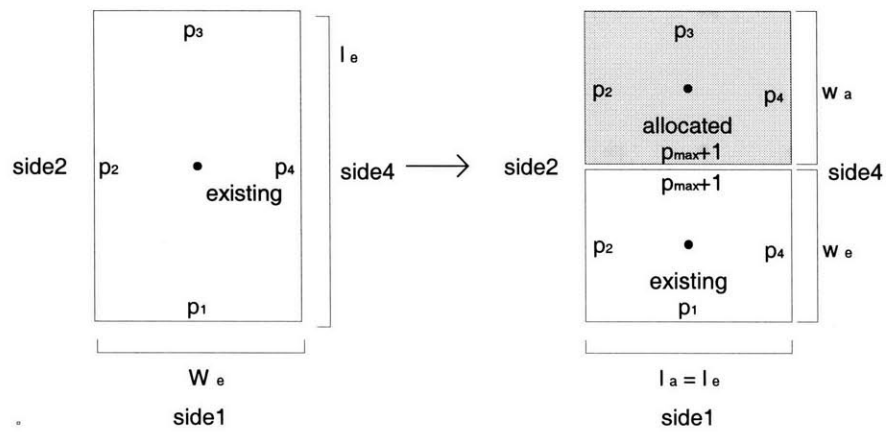
*Note: the description part of the rule is not shown. Please see conditions for rule application below.*

**R37: Allocate space through dissection perpendicular to the width with extra space left**



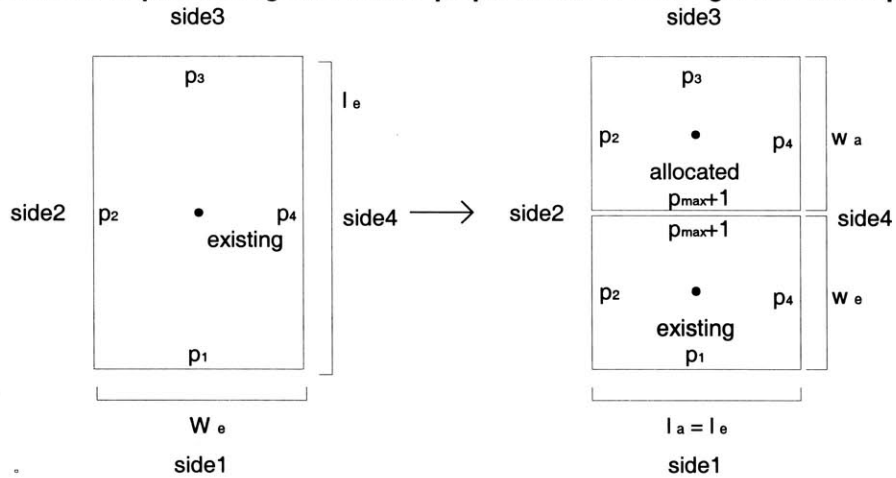
*Note: the description part of the rule is not shown. Please see conditions for rule application below.*

**R38: Allocate space through dissection perpendicular to the length**



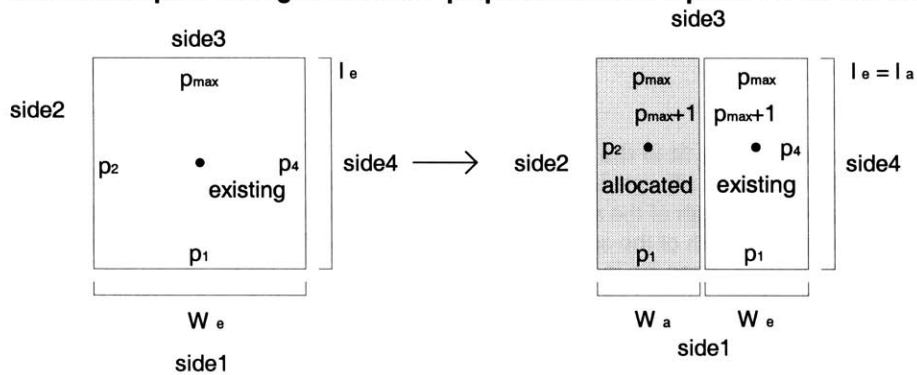
*Note: the description part of the rule is not shown. Please see conditions for rule application below.*

**R39: Allocate space through dissection perpendicular to the length with extra space left**



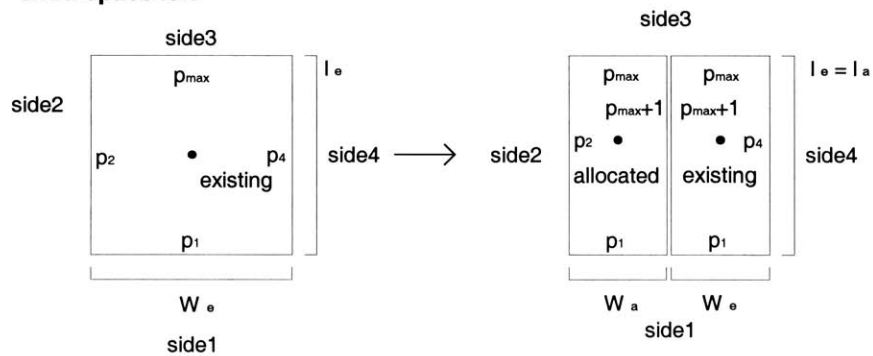
Note: the description part of the rule is not shown. Please see conditions for rule application below.

**R40: Allocate space through dissection perpendicular to the previous level dissection**



Note: the description part of the rule is not shown. Please see conditions for rule application below.

**R41: Allocate space through dissection perpendicular to the previous level dissection with extra space left**



Note: the description part of the rule is not shown. Please see conditions for rule application below.

## Conditions for applying space allocating rules

### 1. Pick up a space to allocate

$$\alpha_{13} \leftarrow \alpha_{13},$$

$\text{space}_i = \max (h (\text{space}_j)), \text{space} \in \alpha_{13} \wedge \notin \delta_{13} \leftarrow \delta_{13}$  use heuristic function to select the space with the heaviest requirements

$$h (\text{space}_j) = \frac{\sum V_{\text{space}_j} \cdot W_{\text{space}_j}}{\sum W_{\text{space}_j}}$$

### 2. Pick up a space where to allocate it

$\alpha_{14} \leftarrow \alpha_{14}, \text{zone} = \text{get\_zone} (\text{space\_to\_allocate})$  gets the zone of the space to allocate

$\delta_{13} \leftarrow \delta_{13}, \exists [\text{zone}, \text{id}_e, \emptyset, ((x_e, y_e, z_e), dx_e, dy_e, dz_e, a_e), n] \in \delta_{13}$

There must exist a zone of the appropriated kind or a space in such a zone with extra space to allocate the space.

### 3. Get the dimensions of the space to allocate

$a_t = \text{get\_area} (\text{space}_i)$

$w_t = \text{get\_width} (\text{space}_i)$

$l_t = a_t / w_t$

Get area of the space to allocate.

Get the width of the space to allocate.

Find the length of the space to allocate.

### 4. Get the dimensions of the existing space

$dx_e < dy_e \Rightarrow dx_e = w_e \wedge dy_e = l_e$

$dx_e > dy_e \Rightarrow dx_e = l_e \wedge dy_e = w_e$

Find which is the width and which is the length of the existing space.

### 5. Does the space to allocate fit in the existing space?

$a_t < a_e$  if the area to allocate is smaller than the area of the existing space,

$l_t \leq l_e$  and the length of the space to allocate is smaller than the length of the existing space,

$w_t \leq w_e$  and the width of the space to allocate is smaller than the width of the existing space, then the spaces fits in the existing space.

### 6. If it does, how does it fit and which rule and transformation to use to allocate the space?

$$6.1 \quad w_t = w_e \wedge l_t = l_e \\ w_t - w_e < w_{t+1} / 2 \wedge l_t - l_e < l_{t+1} / 2$$

If both the width and length match (Fig. xx f), or they do not match but the difference is so small that it is not big enough to place another space in the remaining space if a dissection rule is used, or there are no spaces left to allocate in the zone,

then  $\text{rooms}_t = \text{rooms} (\alpha_{14}, \text{zone})$

$\text{rooms}_a = \text{rooms} (\delta_{14}, \text{zone})$

$\text{rooms}_t - \text{rooms}_a = 1$

use assignment rule.

Problem: the last space to allocate, the least important one according to the heuristics used to pick up spaces for allocating, might be benefited.

$$6.2 \quad \sim (w_t = w_e \wedge l_t = l_e)$$

If neither the widths, nor the lengths match then use a dissecting rule.

#### 6.2.1 Which dissecting rule to use?

$$6.2.1.1 \quad w_t = w_e \wedge l_t < l_e$$

If the widths match (Fig. xx and m), then make a dissection perpendicular to the length

Get the dimensions of the allocated space:

$w_a = w_t$

The width of the allocated space is the width of the space to allocate

$l_a = w_e$

The length of the allocated space is the width of the existing space

6.2.1.2  $w_t < w_e \wedge l_t = l_e$  If the lengths match, then make a dissection perpendicular to the width.

Get the dimensions of the allocated space:

$w_a = w_t$  The width of the allocated space is the width of the space to allocate  
 $l_a = l_e$  The length of the allocated space is the length of the existing space

6.2.1.3  $w_t < w_e \wedge l_t < l_e$

If both the width and the length of the spaces to allocate are smaller than those of the existing space (Fig. ?? h, l, n, o, p, and r), then it can be either one.

even (n) = true

The last dissection was parallel to the x axis

Get the insertion point:  
 allocating zone to the left

$x_{r1} = x_e + a_a / dy_e + 0.20 \wedge$   
 $x_{a1} = x_e \wedge$   
 $x_{wall1} = x_e + a_a / dy_e$

allocating zone to the right

$x_{r2} = x_{in} \wedge$   
 $x_{a2} = x_e + (dx_e - a_a / dy_e) \wedge$   
 $x_{wall2} = x_e + (dx_e - (a_a / dy_e + 0.20))$

Get the width, length, height, and area of the allocated and remaining spaces:

even (n) = false

The last dissection was parallel to the y axis:  
 Get the insertion point:

allocating zone to the back

$x_{r1} = x_e + a_a / dy_e + 0.20 \wedge$   
 $x_{a1} = x_e \wedge$   
 $x_{wall1} = x_e + a_a / dy_e$

allocating zone to the front

$x_{r2} = x_{in} \wedge$   
 $x_{a2} = x_e + (dx_e - a_a / dy_e) \wedge$   
 $x_{wall2} = x_e + (dx_e - (a_a / dy_e + 0.20))$

Get the width, length, height, and area of the allocated and remaining spaces.

6.2.2 Introduce the spaces for evaluation:

+ < [allocated, id<sub>allocated1</sub>,  $\emptyset$ , (( $x_{a1}$ ,  $y_a$ ,  $z_a$ ),  $dx_a$ ,  $dy_a$ ,  $dz_a$ ,  $a_a$ ), n + 1, temp, 1],  
 [remaining, id<sub>remaining1</sub>,  $\emptyset$ , (( $x_{r1}$ ,  $y_r$ ,  $z_r$ ),  $dx_r$ ,  $dy_r$ ,  $dz_r$ ,  $a_r$ ), n + 1, temp, 1],  
 [allocated, id<sub>allocated2</sub>,  $\emptyset$ , (( $x_{a2}$ ,  $y_a$ ,  $z_a$ ),  $dx_a$ ,  $dy_a$ ,  $dz_a$ ,  $a_a$ ), n + 1, temp, 2],  
 [remaining, id<sub>remaining2</sub>,  $\emptyset$ , (( $x_{r2}$ ,  $y_r$ ,  $z_r$ ),  $dx_r$ ,  $dy_r$ ,  $dz_r$ ,  $a_r$ ), n + 1, temp, 2] >

7. If the space to allocate does not fit into the existing space (Fig. Xx a-c, and e, h)

Is any of the dimensions of the existing space 60% smaller than the minimum required?

If yes,

If the space is optional, proceed with the derivation without allocating the space  
 If the space is obligatory, halt derivation, announce failure, and explain why it failed

If not, try to allocate the space with reduced the requirements. Find why it does not fit and lower the dimensions that exceed those of the existing space until the space to allocate fits:

**Case 1:** the area, the width, and the length of the space to allocate is bigger than those of the existing space

$a_t > a_e$   
 $w_t > w_e$   
 $l_t > l_e$

diminish all the dimensions: assignment rule

$$a_f = a_e$$

$$w_f = w_e$$

$$l_f = l_e$$

**Case 2:** the area and the width are bigger but the length is smaller than that of the existing space

$$a_t > a_e$$

$$w_t > w_e$$

$$l_t < l_e$$

$$a_f = a_e \quad \text{diminish the area}$$

$$l_f = l_t \quad \text{keep the same length}$$

$$w_f = a_f / l_f \text{ the width is still bigger}$$

it ends up being an assignment rule

diminish the width, keep the length the same, find the area

$$w_f = w_e \quad \text{diminish the width to fit}$$

$$l_f = l_t \quad \text{keep the same length}$$

$$a_f = w_f \cdot l_f \text{ find the area, which is smaller to that of the space to allocate}$$

it ends up being a dissection rule but the quality is lower (not a good idea) check proportion

**Case 3:** the area and the length are bigger but the width is smaller than that of the existing space

$$a_t > a_e$$

$$w_t < w_e$$

$$l_t > l_e$$

$$a_f = a_e \quad \text{diminish the area}$$

$$w_f = w_t \quad \text{keep the same width}$$

$$l_f = a_f / w_f \text{ find the length, which is still bigger}$$

it ends up being an assignment rule

diminish the length to fit

$$l_f = l_e \quad \text{diminish the length to fit}$$

$$w_f = w_t \quad \text{keep the same width}$$

$$a_f = w_f \cdot l_f \text{ find the area, which is smaller than that of the space to allocate}$$

it ends up being a dissection rule but the quality is lower (not a good idea) check proportion

**Case 4:** the area and length are smaller but the width is bigger than that of the existing space

$$a_t < a_e$$

$$w_t > w_e$$

$$l_t < l_e$$

$$w_f = w_e \quad \text{diminish the width}$$

$$a_f = a_t \quad \text{keep the same area}$$

$$l_f = a_f / w_f \text{ find the length which increases as a result}$$

$$w_f = w_e \quad \text{diminish the width}$$

$$l_f = l_t \quad \text{keep the same length}$$

$$a_f = w_f \cdot l_f \text{ find the area which decreases as a result}$$

**Case 5:** the area and width are smaller but the length is bigger than that of the existing space

$$a_t < a_e$$

$$w_t < w_e$$

$$l_t > l_e$$

$$l_f = l_e \quad \text{diminish the length}$$

$$a_f = a_t \quad \text{keep the same area}$$

$$w_f = a_f / l_f \text{ find the width, which increases as a result}$$

$$l_f = l_e \quad \text{diminish the length}$$

$$w_f = w_t \quad \text{keep the same width}$$

$$a_f = w_f \cdot l_f \text{ find the area, which decreases as a result}$$

If one or more of the dimensions of the allocated space are smaller than those that correspond to the desired and/or the minimum quality levels, record the deficits of each dimension relatively to each of these levels.



## 7.6 Summary

In this chapter, it was proposed a model for the problem of finding a goal-matching design within a specific architectural style called discursive grammar. This model includes a programming grammar and a designing grammar. The programming grammar generates the design brief in such a way that the opportunity for the problem to be over-constrained or ill-defined is considerably reduced, thereby overcoming two of the major hurdles that plagued other models of design. The designing grammar generates the design solution using heuristics to select at each state of the derivation the rule that takes the evolving design closer to the design goal. The use of heuristics contributes to speed up the derivation, which is a major drawback in previous goal-oriented approaches. On the other hand, the use of heuristics does not guarantee that the proposed model generates the best solutions. The proposed model is illustrated with a discursive grammar for Siza's Malagueira houses, in which the programming grammar was modeled after the Portuguese guidelines for the design of social housing, and the designing grammar was developed after Siza's Malagueira houses design rules.

## References

- HERBERT, G; SWOPE, M; et al – Some Performance Guidelines for the Design and Evaluation of Environmental Spaces in the Dwelling. Haifa, 1978.
- KNIGHT T W, 1989a, "Color grammars: designing with lines and colors" *Environment and Planning B: Planning and Design* **16** 417-449.
- MARCH, L. - Architectonics of Humanism: Essays on Number in Architecture, John Wiley & Son Ltd., New York, 1998.
- MITCHELL W J, 1989 *The Logic of Architecture*, MIT Press, Cambridge, MA.
- MANSO, A. Costa; FONSECA, M. Santos; ESPADA, J. Carvalho «Informação sobre custos: Fichas de rendimentos. (2 vols)» Lisboa, 1997. 1030 p.

MANSO, A. Costa; FONSECA, M. Santos; ESPADA, J. Carvalho «Informação sobre custos: Fichas de actualização» Lisboa, 2000. 452 p.

PALLADIO, Andrea - I Quattro Libri dell'Architettura, Ed. L. Magagnato and P. Marini, Milan, 1980.

PEDRO, J B – Programa Habitacional. Espaços e compartimentos. Coleção Informação Técnica e Arquitectura, nº 5, Lisboa, LNEC, 1999a.

PEDRO, J B – Programa Habitacional. Habitação. Coleção Informação Técnica e Arquitectura, nº 5, Lisboa, LNEC, 1999b.

PEDRO, J B – Indicadores de Qualidade Arquitectónica Habitacional. Dissertação apresentada em cumprimento das exigências de provas de doutoramento na Faculdade de Arquitectura da Universidade do Porto. Porto, Ed. FAUP, 2000.

PORTAS, N – Funções e Exigências das Áreas de Habitação. Coleção Informação Técnica de Edifícios, nº 4. Lisboa, Ed. LNEC, 1969.

REDDY G, CAGAN J, 1995 "An improved shape annealing algorithm for truss topology generation" *ASME Journal of Mechanical Design* 117 315-321.

STINY G, GIPS J, 1972, "Shape Grammars and the Generative Specification of Painting and Sculpture" in C V Freiman (ed) *Information Processing 71* (Amsterdam: North-Holland) 1460-1465. Republished in Petrocelli O R (ed) 1972 *The Best Computer Papers of 1971*: Auerbach, Philadelphia 125-135.

STINY G, 1981, "A note on the description of designs" *Environment and Planning B: Planning and Design* 8 257-267.

STINY G, 1990, "What is design?" *Environment and Planning B: Planning and Design* 17 97-103

## **8. Implementation**

### **8.1 Introduction**

This chapter describes three important, steps towards the implementation of the envisioned system for the design of mass housing. These steps represent preliminary efforts, rather than finished work. The first step is the development of the PAHPA-Malag interpreter, the computer program encoding the discursive grammar. The section on the interpreter describes the proposed system's architecture, as well as the specific computer tools used in the implementation. The second step is the development of a Web site on the grammar that functions as a catalog of existing houses, a tool for teaching the grammar, and a tool for generating new houses on-line. The section on the Web site describes its structure, including its modules, navigation strategies, and user-interaction. The third step is the framework of the proposed design system, which includes rapid prototyping and visualization techniques, in addition to the interpreter, and the Web site. The section on the framework, explains how such visualization techniques can be used for communicating the housing solution to the client.

### **8.2 The interpreter**

A grammar can be used as an analytical and as a generative tool. As an analytical tool, the Malagueira grammar allows one to understand the rules behind the design of Malagueira houses. As a generative tool, it permits the generation of a large set of design solutions based on such rules. As a generative tool, a grammar can be used by hand, as did the subjects in the experiments described in Chapter 6. However, the full generative power of a grammar is unleashed only when the grammar is codified into a

computer program. Moreover, such a program, called an interpreter, becomes an effective design assistant only when it has a user-friendly interface. Effort was, thus, placed in building a user-friendly interpreter for the PAHPA-Malag grammar.

Previous implementations of shape grammars can be grouped into two groups, based on the type of representation used for shapes. The first group includes visual implementations, such as the one by Tapia (1999). They are said visual because shape rules match directly on the geometry, as "we see it," by closely following Stiny's mathematical foundation of shape grammars. The second group includes symbolic implementations, such as those developed for engineering grammars like Reddy's and Cagan's truss design grammar (1995), or Shea's dome design grammar (1996). In the symbolic implementations, shape rules match on symbolic descriptions of shapes. As a result of the different representations, visual implementations support shape emergence, whereas symbolic ones do not. On the other hand, symbolic implementations make it easier to deal with parametric shape rules. Because shape emergence is not a feature of the PAHPA-Malagueira grammar, whereas parametric shape rules are, the proposed implementation is symbolic.

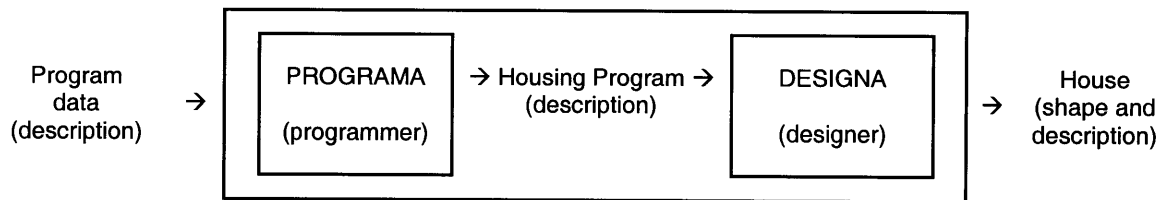


Figure 8.1 - From program data to a housing solution: the two modules of MALAG.

The PAHPA-Malagueira interpreter, called MALAG, consists of two main modules: the programmer, and the designer, as shown in Figure 8.1. The programmer, called

PROGRAMA, processes the program data to generate the housing program. The designer, called DESIGNA, takes the housing program and generates a housing solution within the Malagueira language.

The system's architecture of MALAG is diagrammed in Figure 8.2. The programming languages used in the implementation are HTML, Java, Clips, and VRML. These languages were chosen to allow MALAG to run on the web. The system's architecture consists of four modules: an interface, an expert system, a solid modeler, and a display. The interface of PROGRAMA is a Java applet. The expert system uses the Java expert system shell, called JESS, developed by Sandia Corporation. This shell is an interpreter of the Clips language written in Java. The description part of the programmer and the designer's grammar rules are written in Clips, a dialect of Lisp. The solid modeler is written in Java that use indexed lists to represent shapes. Both the modeler and the expert system constitute the core of DESIGNA. The output of MALAG is a VRML file, which is read by the VRML viewer Cosmo Player, to display the evolving design on the web.

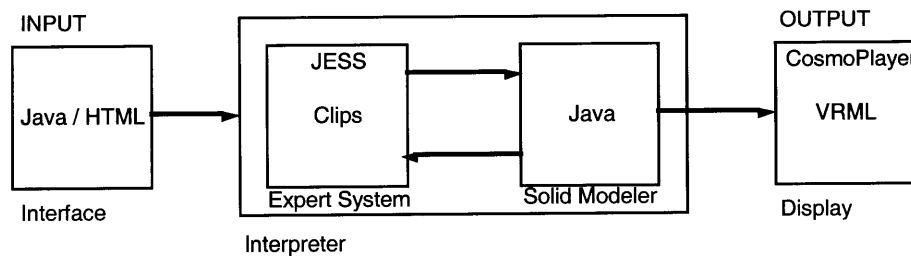


Figure 8.2 - The MALAG system's architecture.

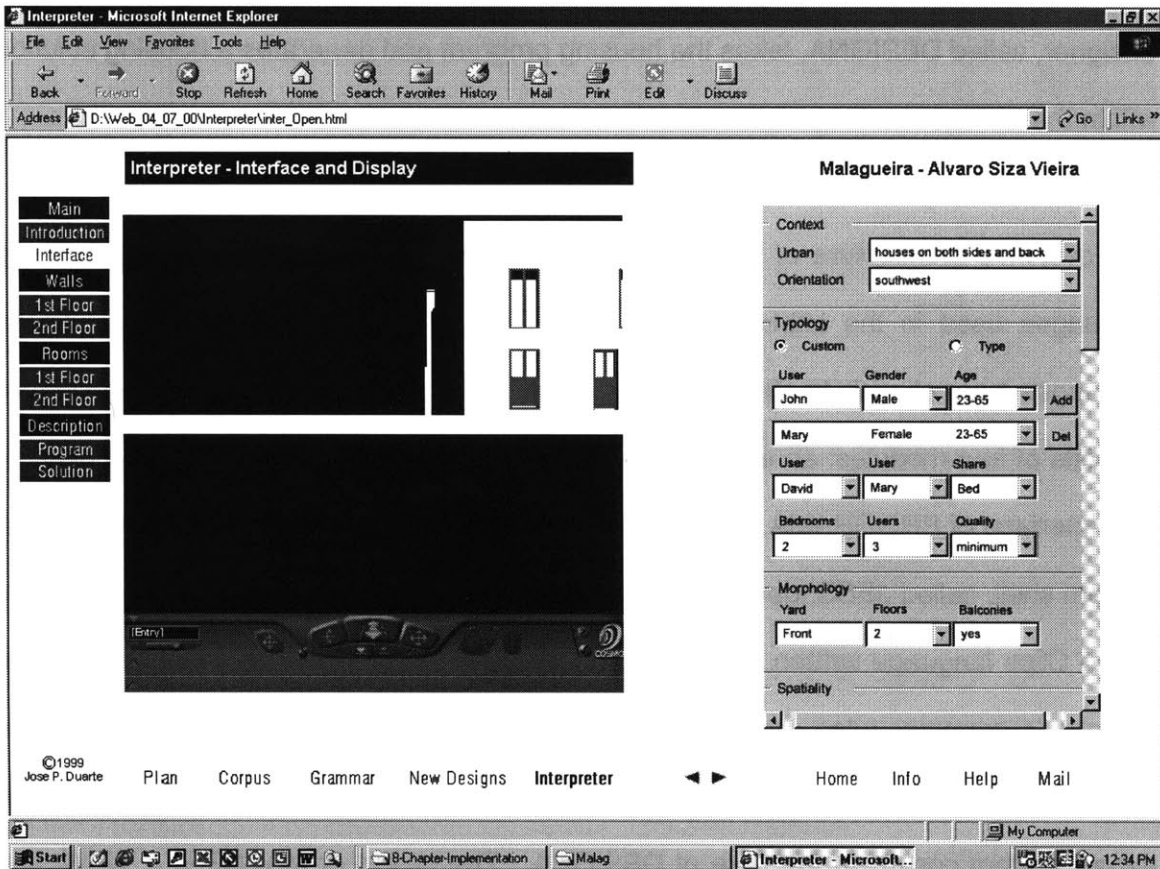


Figure 8.3 – The MALAG Web interface. The PROGRAMA's interface on the right is used to prompt the program data, whereas the DESIGNA's display on the left is used to see walk through the solution. The side bar on the left launches other views.

The MALAG's Web interface is shown on Figure 8.3. It includes: on the right, the PROGRAMA's interface used to prompt the program data; in the middle, the DESIGNA's display to visualize the housing solution: and on the left, a side bar to launch different views. The use of MALAG during one work session is briefly explained below. The user uses the interface to interactively describe the dwellers and the site, and to refine the housing program. As explained in Chapter 7, the housing program features are organized into constraints, qualities, and cost. As the user provides information on the constraints (context, typology, and morphology), PROGRAMA provides the default values and weights of quality features, and the possible ranges of variation of such values and weights. In addition, it also provides the quality level and the estimated cost

of the house described in the housing program. The user can then modify this default program by changing the features values and weights within the allowed ranges. As the user makes such changes, PROGRAMA updates the quality level and the cost.

When the user is satisfied with the housing program, the user notifies PROGRAMA by hitting the 'done' button at the bottom of its interface. The user can visualize the housing program by hitting the 'program description' button on the side bar of the Web interface. The user also can name and save the housing program, or send it to DESIGNA, by hitting the appropriate 'save' and 'send' buttons at the bottom of PROGRAMA's interface. Then, DESIGNA will generate a solution, using a set of heuristics to minimize the distance between the program description and the design description. The features weights are used in this process to decide which requirements to satisfy in first place, thereby minimizing the distance between the program and the design descriptions.

The user can follow this generation from different views by hitting different buttons on the side bar of the Web interface. (Figure 8.4) Namely, the user can follow the generation of either floor from a 'walls' or a 'rooms' view. A 'walls' view shows the walls and other building elements such as, windows, doors, and so on. A 'rooms' view is an abstraction that shows only the rooms that form the floor, using colors to identify the different types. Once the housing solution becomes available, the user can walk through the house, or look at its description to compare it with the program description. Then, user can name, save, and retrieve the solution.

Given an existing housing program, the user can resend it to DESIGNA to generate another solution. Because PROGRAMA is not fully deterministic, the new solution will likely be different from the previous one. The user also can reset the features weights

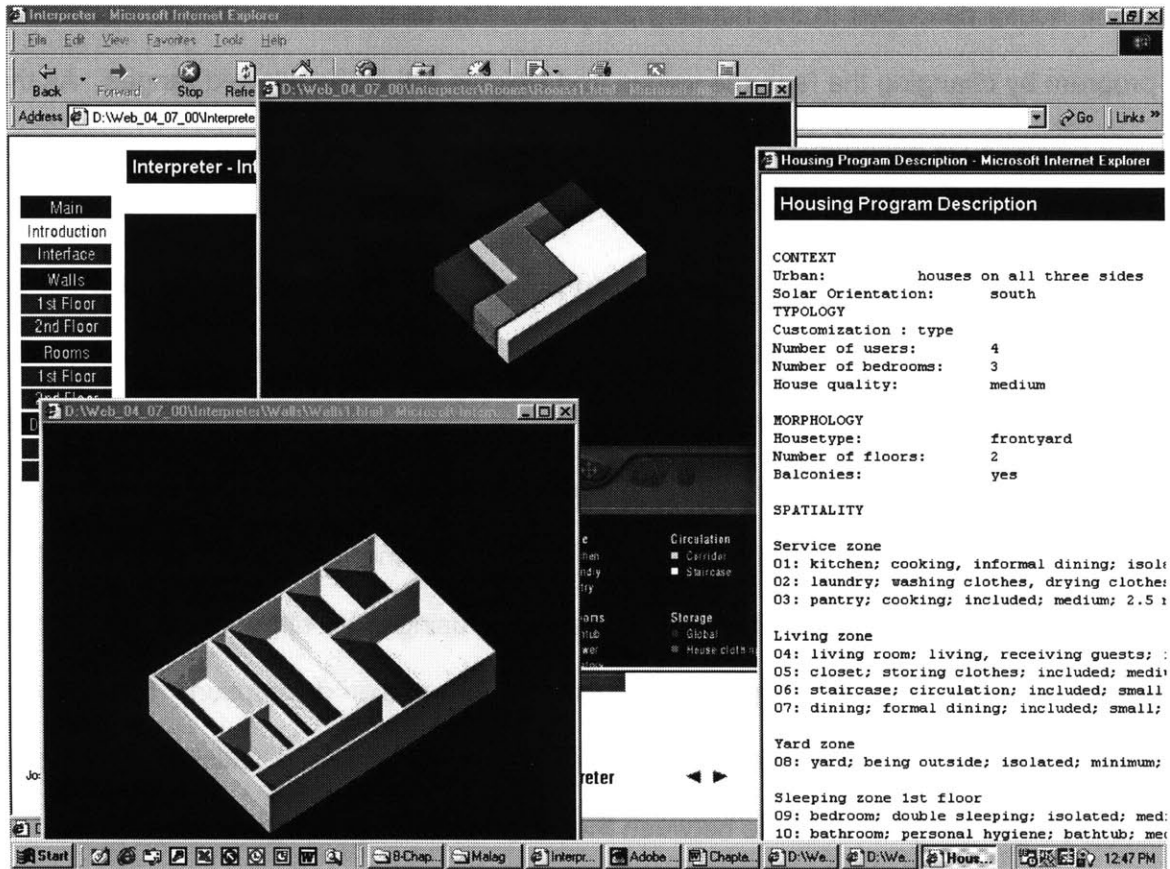


Figure 8.4 – Different views permitted by MALAG. From left to right: 'walls', 'rooms', and 'description' views.

before sending it to DESIGNA. The new solution will represent a different balance among the requirements. Finally, after seeing a solution, the user can change some of the programmatic requirements, to refine the design problem. The use of default values and limits for the variation of values helps PROGRAMA to define the problem and to avoid over-constraints. The use of heuristics allows DESIGNA to find a solution, even if the problem is still over-constrained.

## 8.2 The Web site

The Web site was designed with three goals in mind: to constitute a catalog of existing houses, to teach designers how to generate new designs using the grammar by hand,



and to automatically generate new houses on-line. The site is organized into five main sections—Plan, Corpus, Grammar, New Designs, and Interpreter—that constitute the core of the site, and three subsidiary pages—Home, Info, and Help—with additional information. The structure of the Web site is diagrammed in Figure 8.5. The Home page is the front page, and like all the others, it contains direct links to the main sections and subsidiary pages. The Info and Help pages are mainly descriptive and provide the user with background information.

The Plan section describes the concepts behind the design of the Malagueira urban plan, and it presents aerial and ground views of the neighborhood, the housing, the streets, the public spaces, and the urban facilities. The pages in this section allow the user to explore thematic maps by placing the cursor in their legends. The themes included are urban uses, housing types and variations, phases of construction, and type of promotion. The last page in this section contains links to sub-pages that lead the user to detailed information on a typical housing block, including an interactive evolution of the block. (Figure 8.6)

The Corpus presents the catalog of existing designs used to infer the Malagueira housing grammar and it is classified into subtypes and variations. For each subtype, there are four pages with detailed information on the subtype and its variations. These pages include pictures of existing houses, 3D digital models, schemes of their functional organization, plans, sections and elevations, numerical data, and VRML models. For most of the subtypes, there is also a page that presents a step by step derivation of the design according to the rules of the grammar. (Figure 8.7)

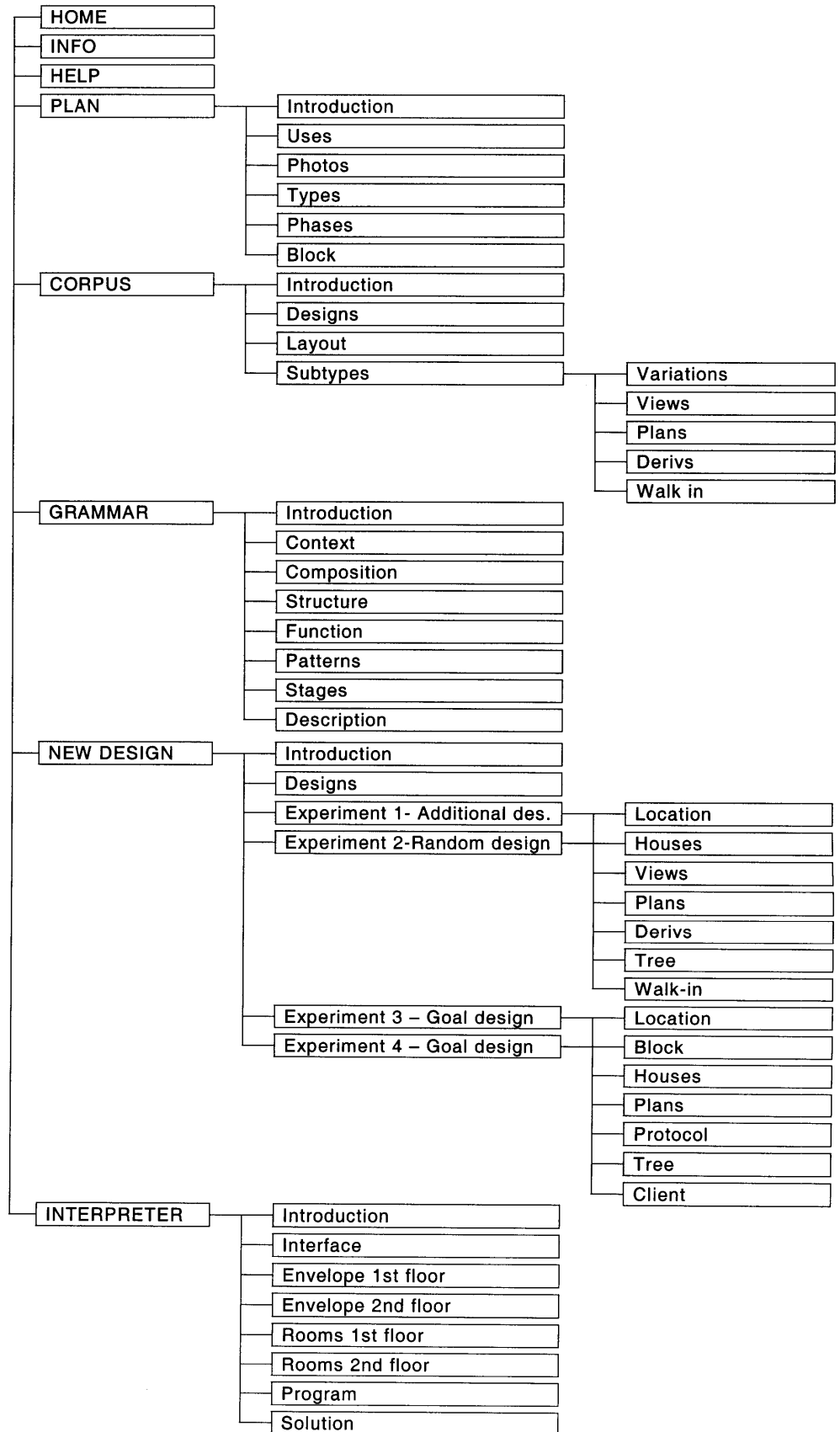


Figure 8.5 - The tree structure of the Web site.

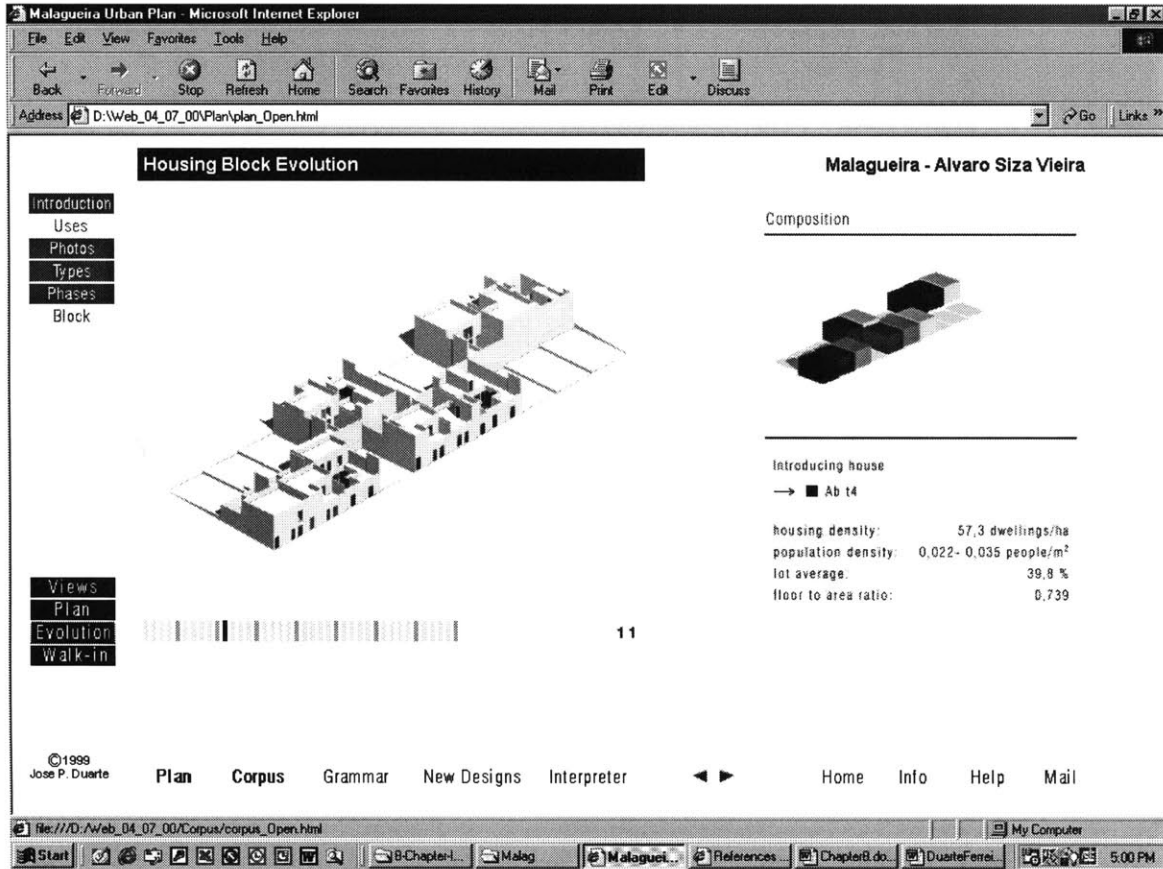


Figure 8.6 – One of the pages in the Plan section of the Malagueira Web site.

The Grammar section explains the details of the grammar, aiming at showing, in a visually understandable way, how the Malagueira grammar can be used in the design of new houses by hand. The information is grouped into eight thematic pages that brief the user on shape grammars, the compositional rules of the grammar, the structural system, the urban context, the universe of possible designs, the functional organization, the stages and steps in the generation of designs, and the connection between the description grammar and the shape grammar. (Figure 8.8)

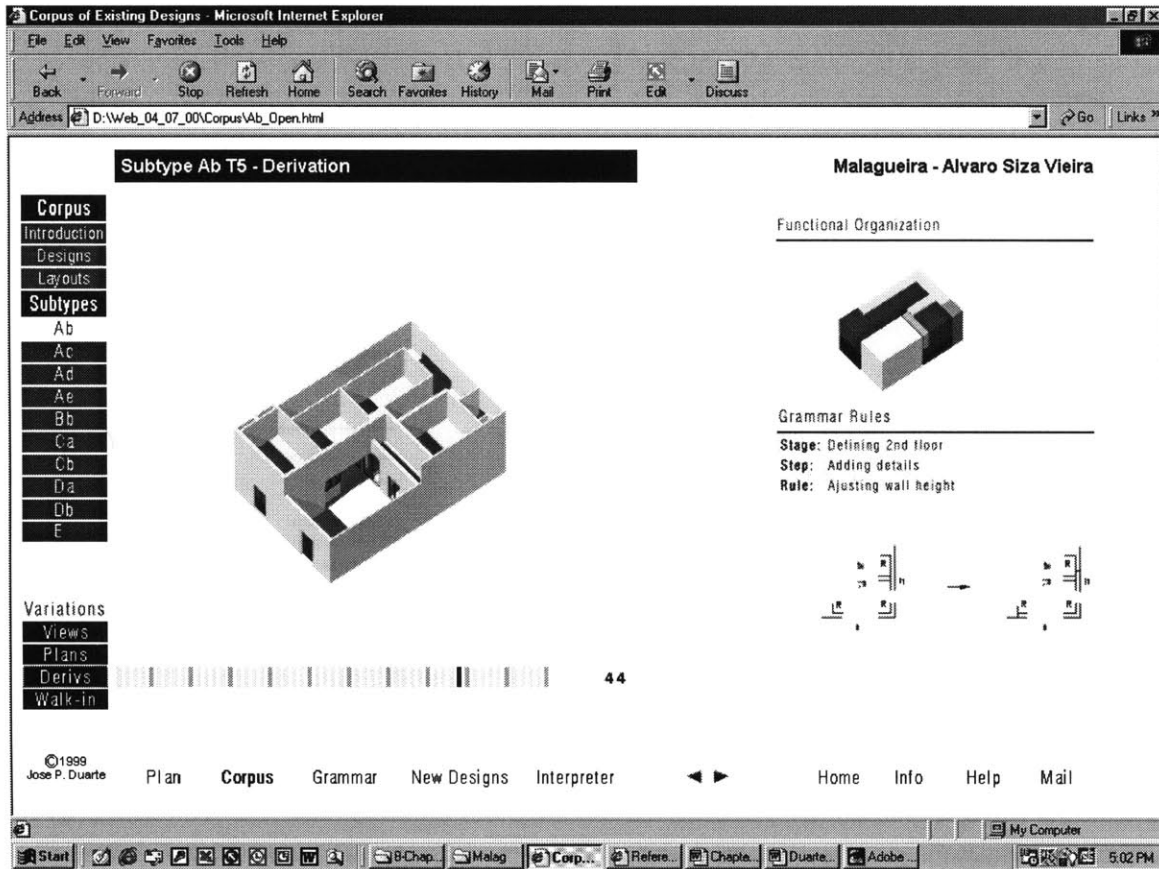


Figure 8.7 – One of the pages in the Corpus section of the Malagueira Web site.

The New Designs section contains the immediate outcome of the four experiments described in Chapter 6, in which subjects used the grammar to design new houses. It includes a text page that describes the goals, subjects, setting, task, procedure, and the main conclusions of each of the experiments, as well as pages showing the new designs. (Figure 8.9)

The Interpreter section contains the applet of the computer program described in the Section 8.1, which users can use the to generate on-line customized houses in the Siza's Malagueira style.

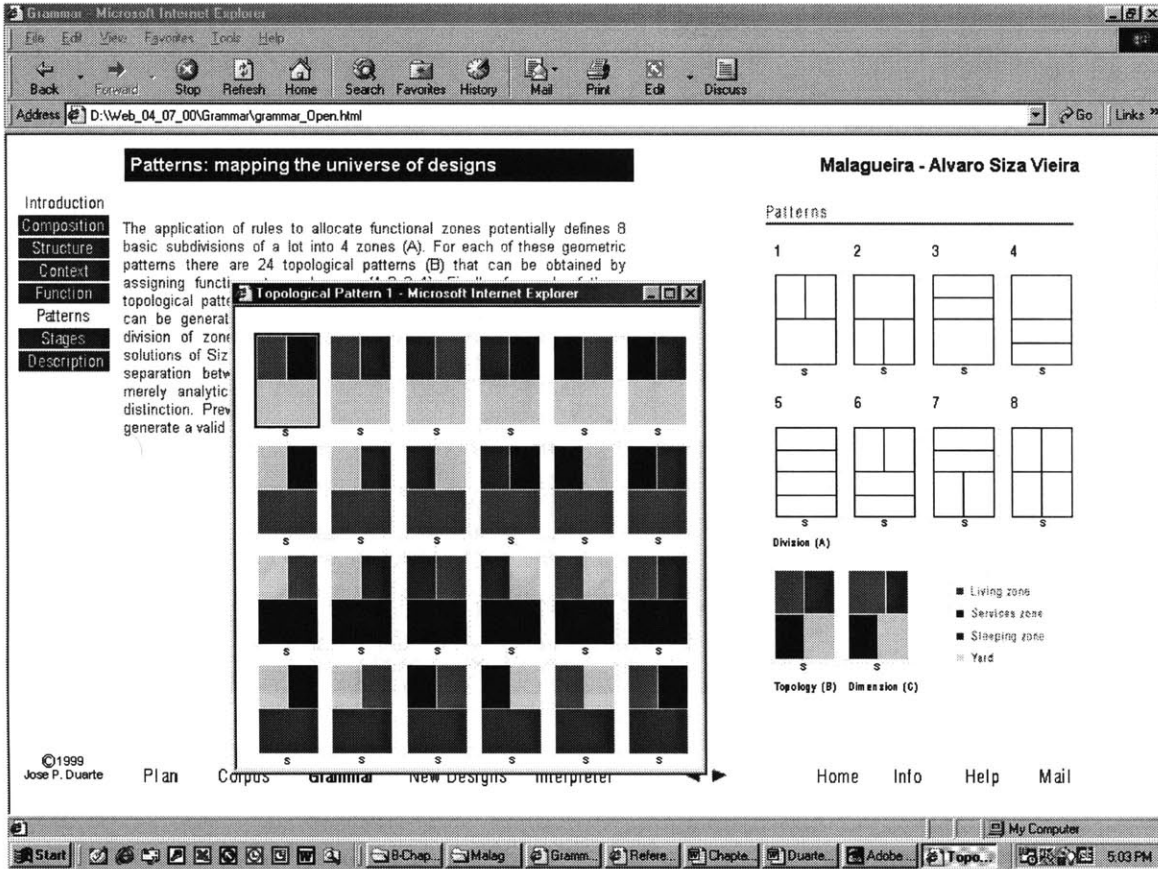


Figure 8.8 – One of the pages in the Corpus section of the Malagueira Web site.

Navigation in the site requires the use of a frames compatible browser and a true color monitor, and it is optimized for a 1024 by 768 dpi resolution. The pages are written in the HTML language and JavaScript functions were used to allow an interactive behavior through event handling routines. Nested frames were used to achieve an effective interface in which the uploading time is reduced to a minimum. The use of JavaScript and nested frames became necessary because the large amount of visual information required high-quality images.



Figure 8.9 – One of the pages in the New designs section of the Malagueira Web site.

The main navigation system is based on JavaScript encoded keywords-bars placed on two fixed frames. The frame on the bottom of each page contains the main "*Site Bar*", and the frame on the left contains the "*Section Bar*". The "*Site Bar*" contains a set of buttons on the left that is used to change between the different sections of the site, and another set of buttons on the right that give access to the subsidiary pages referred to above. The "*Section Bar*" is used to navigate from page to page within each section. Whenever a page contains a set of subsection pages, the left frame is split into top and bottom frames, which contain the "*Section Bar*" and the "*Subsection Bar*," respectively.

The screen area used to display the contents of each page has a title at the top and it may have a single area or left and right-framed areas, depending on the specific contents displayed. To create links among text pages, named anchors coupled with JavaScript functions were used to identify options and to immediately scroll the page and display the identified item. For the pages that contain mainly images, visual metaphors like timelines and colored boxes are used to allow the user to explore its contents. This interactivity can result from the use of *image maps* or JavaScript functions applied to areas sensitive to cursor passage. By clicking on some icons or images the user launches spare frames containing additional or detailed information about whatever is depicted in the icon or image. Finally, if the user has a Cosmo Player plug-in installed, 3D models in the VRML format can be explored in a virtual environment. The Web site is accessible at <http://destec.mit.edu/malag/> .

#### **8.4 The envisioned housing design framework**

The web site described above is part of a larger framework proposed for the design of customized housing. The other components of this framework are the use of rapid prototyping techniques and virtual reality environments. The web site simultaneously provides the user with a catalog of existing designs and a computer program to generate new designs on-line. The catalog provides prospective dwellers with a way of understanding the available housing solutions, and a way of structuring their needs. The computer program allows a thorough exploration of the space of design solutions in search of a solution that matches user needs. The goal is to increase user satisfaction.

One of the advantages of high user satisfaction is to avoid post-construction changes to the dwelling. However, to decrease the likelihood of such changes it also is important that users have a way of assessing their houses before they are built. As they are not

designers, their ability to understand and visualize designs just by looking at traditional representations, such as plans, sections, and elevations, is rather limited. It is, therefore, necessary to provide them with representations that they can understand. In the envisioned framework, this is achieved by the use of rapid prototyping techniques and a virtual reality environments.

In brief, rapid prototyping is a technique that automatically produces a physical model from a CAD file. Among these techniques, there are techniques that create the model through an additive process. One of this is the Deposition Model (FDM) developed by Stratasys, which builds the model from bits of a fused material that solidify once deposited. Another technique is stereolithography, which builds the model by using a laser to solidify a polymer liquid at given points in the space. In 3D printing, developed at MIT, the model is built by using a glue to agglomerate a powder. Some of these techniques were used in the experiments described in Chapter 6 to produce physical models of new houses, which were then shown to clients. One can imagine techniques like these being used to explain to clients how their future house will look like. For more information on rapid prototyping, please see Burns (1993).

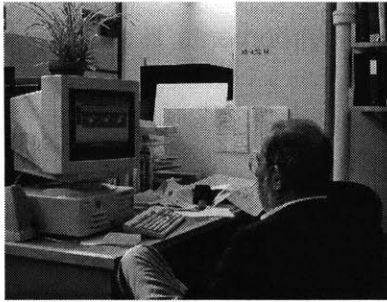
The term virtual reality often is abusively used. In the appropriate sense, the term means to create an environment that simulates reality to an extent that human observers are led to believe that they are in a real environment. There are several virtual reality techniques available with varying degrees of immersion. The simplest technique, uses glasses to give the observer the impression of looking at a 3-D environment while looking at a flat computer screen through such glasses. On the opposite extreme, is the system known as CAVE, in which the observer is fully immersed in the environment through a head-mounted device and aptic gloves. In the CAVE, observers have the



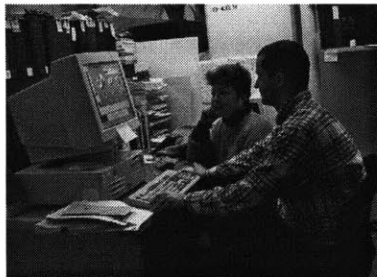
impression that they look around as they turn their heads, or that they touch and open doorknobs, climb stairs, and so on. Our idea is to have environments like these to enable clients to visit their house before construction.

Siza says that user participation “promotes conflict and delays when it is not simulated or mystified,” and he concludes that “the difficulty is not to build homes but communities.” (Quoted in Fleck 1992.) Our goal in proposing the described framework is to enable community building by providing a tool that fosters the participation of community members in the design of their homes. According to Black, (quoted in Fleck 1992) user participation put a heavy burden on project management at Malagueira. Such a burden led some critics to say that the project was too expensive to be considered social housing. Another goal of the proposed framework is to simplify user participation to diminish the burden on project management. It is proposed to make the Malagueira Web site available from Évora city hall’s Web site. This would allow prospective dwellers to log in into the Internet and access a wide network of contacts and information on the project. While the public control of the Web site serves the social scope of the proposed framework, this is not the only way of using such a framework. Independent designers and housing companies could develop similar frameworks to advertise and sell their own customized designs. Figure 8.10 shows some of the ways in which interpreters, virtual reality and rapid prototyping can be used in the generation and assessment of designs.

### The interpreter



Use by the architect



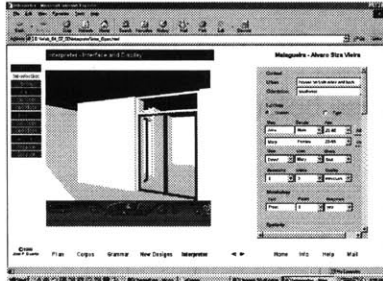
Use by the architect with the client

Use by the client alone

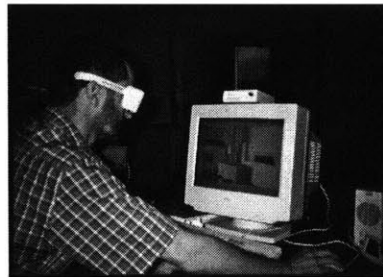


Use by a salesperson

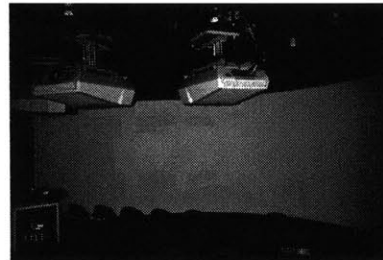
### Virtual Reality



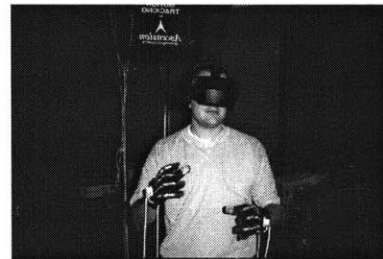
Digital walk-through



Desktop solution

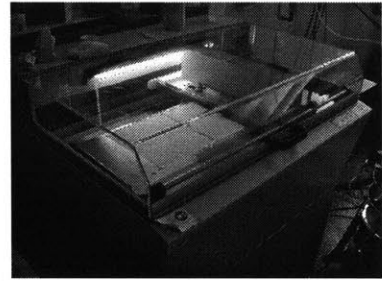


Partially immersed environment

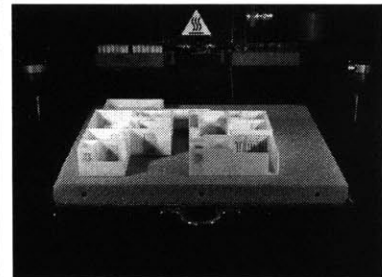


Immersed environment

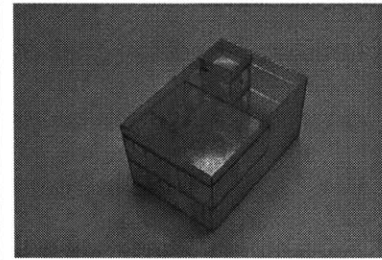
### Rapid prototyping



3D-printing



FDM (Fused deposition model)



Stereolithography



The client looking at the model

Figure 8.10 – The envisioned framework for the design of mass housing: the interpreter is used either by the designer or the client to input requirements and generate solutions (left column), virtual reality environments with different degrees of immersion (middle column), and various rapid prototyping techniques (right column) are, then, used to visit and assess the solution before construction.

## 8.5 Summary

A Web-based digital framework composed of a discursive grammar, a computer interpreter of the grammar, rapid prototyping techniques, and a virtual reality environment, is proposed as a tool for designing customized mass housing. This framework augments the designer's creativity by enhancing the ability to generate diverse designs in response to diverse user requirements. It also increases the ability to convey to clients how their future houses look like thereby avoiding post-construction changes and leading to greater user satisfaction. It has been described how such framework could be implemented.

## References

- Tapia M, 1999, "A visual implementation of a shape grammar system", *Environment and Planning B: Planning and Design* 26:1 59-73.
- REDDY G. and CAGAN J., 1995, "An improved Shape Annealing Algorithm for Truss Topology Generation", in *Journal of Mechanical Engineering, U.S.A.*, Volume 117, pages 315-321, June.
- SHEA K and CAGAN J, 1996, "Innovative Dome Design: Applying Geodesic Patterns with Shape Annealing", *Artificial Intelligence for Engineering Design Analysis, and Manufacturing*.
- Burns, Marshall, 1993, "Automated Fabrication." Prentice-Hall, New Jersey.
- FLECK, B., 1992, "Álvaro Siza", Relógio d'água editores; Lisboa, Portugal; page 79.



## **9. Conclusion**

This chapter summarizes the research presented in this dissertation, lists its contributions, and outlines paths for future work.

### **9.1 Summary**

The ultimate goal of this dissertation is a framework for the mass customization of housing that includes computer aided-design and production systems. Its focus is on the design part by proposing a mathematical model, called discursive grammar, for an interactive system to be used in the automatic exploration of criteria-matching housing solutions.

A discursive grammar includes a programming grammar that generates the design brief based on user and site information, and a designing grammar that generates a design solution that satisfies the requirements specified in the brief. The solution is achieved by using a set of heuristics to choose the rule that takes the evolving design closer to the specified goal. The heuristics are different at different stages of the derivation so as to provide an appropriate estimation of the distance to the goal, based on the currently available contextual information.

The model is illustrated with a specific grammar called PAHPA-Malagueira grammar. In this grammar, the programming grammar encodes the rules of the Portuguese housing guidelines known as *Programa Habitacional*, as well as the intelligence of a human designer using such guidelines to generate a design brief. The designing grammar codifies the rules followed by the architect Álvaro Siza in the design of patio houses at Malagueira, and they were developed after a proposed methodology comprising descriptive, analytic, synthetic, and goal tests. The heuristics were developed after a protocol analysis study in which subjects were asked to design criteria-matching houses.

The division of the problem into two programming and designing steps helped to overcome the ill-definition and over-constraining problems, which constituted two major hurdles in previous design systems. The programming grammar leads the user to increasingly clarify and de-constrain the problem by generating the design requirements that fit the current design context, which the user can change within allowable limits, until the brief is defined. If this is still over-constrained, the designing grammar includes rules to de-constrain it and to generate the feasible solution that is closer to design goal.

## **9.2 Contributions**

This dissertation makes several findings, as well as minor and major contributions, to the field of design grammars, in particular, and to architecture, in general, as described below.

### 9.2.1 Findings

The findings that emerged from the research presented in this dissertation are:

Hand-use oriented grammars are different from computer-oriented ones. Monitoring studies permitted to identify an important difference between developing a grammar for use by hand by designers and developing a grammar for computer-implementation. The difference is related to the way in which rules are formalized. Rules with a strong use of mathematical symbols become difficult to grasp by human designers and are more appropriate for computer implementation. On the other hand, rules with strong use of labels and weights put a bigger burden on the computer implementation.

Teaching grammars are different from designing grammars. Monitoring studies also showed that writing a grammar to teach designers how to design in the style of another designer is different from writing a grammar for designing in the style.

Skillful designers are better modeled by 'strong grammars' than 'weak grammars.' A protocol study of designers using the grammar showed the difference between grammars where the knowledge is more on the side of the generator, here called 'strong', and grammars where it is more on the side of the evaluator, here called 'weak'. Designers use of grammars falls in between, but skillful designers are better modeled by strong grammars than weak grammars. Results suggest that skillful designers use appropriate heuristics to traverse the space of design solutions towards the goal, thereby diminishing search time and increasing the likelihood of achieving a better solution.

'strong' grammars put a burden on the development time. As a consequence of being knowledge-intensive, 'strong' grammars require more time for acquiring and inferring such knowledge than 'weak' grammars do.

### **9.2.2 Minor contributions**

The minor contributions are:

A systematized methodology for developing shape grammars. This methodology comprises descriptive, analytic, and synthetic tests used in previous approaches, but also a goal test to verify a grammar's ability to generate criteria-matching designs. In addition, it involves two steps. First, to develop the exhaustive set of rules that could be derived from the compositional principles observed in the corpus designs and second, to limit such an exhaustive set whenever it seemed that it would oppose the designer's design principles.

Use of protocol studies to monitor designers using grammars. This dissertation is the first to monitor the use of grammars by human designers using protocol studies. These monitoring studies are important to better understand how grammars can be used in practice and how they should be developed in the future.

A shape grammar for the work of a living architect. This dissertation is the first grammar developed for the work of a living architect with the architect's full support. This permitted to confirm the ability of the grammar paradigm to codify a designer's style and to represent its implicit design knowledge. This will contribute to foster the use of



computational design systems on the behalf of other practicing architects, thereby leading to more research, and subsequently, to the development of practice-oriented applications.

A description grammar for real a world application. The dissertation presents the first application of the concept of description grammar to a real world situation by proposing a description grammar for the Portuguese housing guidelines. The development of such grammar, in turn, made it possible to encode design regulations into a coherent rule-based system.

An application of the parallel grammars paradigm to solve a real design problem. The dissertation is among the first to validate the use of the parallel grammar paradigm to model a real design problem, by using it to represent the multitude of viewpoints involved in the design of a house and to model their mutual dependency.

A general packing algorithm for space allocation. The dissertation proposes a new packing algorithm embodied into a shape grammar for Siza's Malagueira houses. It is, nevertheless, a general algorithm that can be applied to other design strategies based on the dissection of rectangles. The adjustment of the algorithm to specific styles can be done by controlling the label and description parts of the grammar rules.

The system's architecture of a Web-based interpreter for the grammar. The dissertation proposes a system's architecture for an interpreter of the PAHPA-Malagueira grammar that runs on the Web.

### 9.2.3 Major contributions

The major contributions of the dissertation are:

1. The outline of a system for the mass customization of housing. This system includes computer-aided design and production systems. The design system, which is the focus of the research, encompasses a Web-based interactive system for the exploration of design solutions, as well as virtual reality and rapid prototyping techniques for their visualization. The use of such a system will enable a move from mass production towards mass customization.
2. A mathematical model for the interactive design system. This model, called discursive grammar, is an extension of the shape grammar formalism and it includes a shape grammar, a description grammar, and a set of heuristics that allows the generation of criteria-matching designs. The model overcomes drawbacks of previous approaches by generating both syntactically and semantically correct designs using a heuristic search mechanism embedded into the rules, instead of an external stochastic mechanism. This potentially enables a decrease in the derivation time thereby making it reasonable to develop Web-based interpreters.
3. The prototype of a Web site for teaching the grammar in a visually coherent way. An interactive Web page in which the process of using the grammar is explained using interactive gadgets is proposed as a way to overcome the difficulties posed by technically oriented documents that have been traditionally used to describe shape grammars. The Web page is part of the mass-customization system mentioned

above and it includes a catalog of existing designs, a tool for teaching how to design new ones and, ultimately, a mechanism for generating new houses on line.

4. A rigorous method for understanding and teaching architectural styles. The dissertation uses as a case study the houses designed by Álvaro Siza at Malagueira and proposes a grammar for this style that can be used to teach the style with mathematical rigor. Similar methods might be applied to other styles leading to the explanation of general architectural qualities with the same rigor.

### **9.3 Future work**

This dissertation represents a major step towards the development of the proposed framework for customizing mass housing. Ideas for future work, thus, fall into two categories. The first category includes improvements to the current research, whereas the second incorporates other major steps towards the implementation of that framework. The improvements and the major steps are outlined below.

#### **9.3.1 Improvements**

The improvements are:

Develop the computer implementation. The dissertation proposes a system's architecture for the discursive grammar interpreter, provides the specifications of its user interface, and tests them by making a small, partial implementation of the system. The

next logical step is to complete such an implementation. This will enable extensive testing and refinement of the model, as well as its use in practice.

Add local 'optimization' rules to the grammar. The heuristics used to guide the derivation of the design towards the goal were modeled after protocol studies and attempt at finding the best shortcut to reach the goal without the need for 'optimization'. The computer implementation will help to determine whether there is the need to add rules for improving the design locally, at each step in the derivation, as designers did in the experiments.

Improve the teaching capabilities of the Web page. One of the goals of the development of the Web page was to teach the Malagueira grammar to designers so that they could use it to generate customized houses. Experimental results showed that the teaching capabilities of Web site should and could be improved. Future work should be developed with this aim, which can be undertaken by using the current site to develop more experimental work.

Undertake more monitoring studies. The undertaking of further monitoring studies, including protocol ones, is instrumental to understand how people use grammars in design. Such an understanding, will permit to tailor the design of grammars and grammar interpreters to their effective use by human designers. In addition it should help to identify other strategies used by human designers in the generation of goal-oriented designs.

### 9.3.2 Major steps

The proposed framework for customizing mass housing includes a computer tool for generating solutions, visualization techniques for their assessment by clients, and computer-aided manufacturing techniques for producing the house. Major steps for future work are related to each of these three components of the framework.

Use the grammar paradigm to develop new design systems for housing. The grammar models Siza's systematic approach to housing. Future work should aim at doing the reversal, that is, to use the grammar paradigm to help other designers to develop similar approaches. One possibility for developing new housing design systems is to start by changing the Malagueira rule set to create new grammars. This can proceed by altering the label and description part of the Malagueira rules, by deleting existing rules, or by introducing new ones.

Develop on-line visualization techniques. The visualization techniques considered in the current research consist of rapid prototyping and virtual reality. These techniques work well, but either required sophisticated equipment that the user is unlikely to have, or an effort on his or her behalf to navigate in the house if a simple 3D viewer is used. These drawbacks limit on-line clients' ability to assess solutions. Therefore, future research should aim at developing alternative techniques to overcome such drawbacks.

Decompose the houses into building parts for manufacturing. In the current grammar the knowledge about the building system is encoded at a very abstract level. Future work should put an effort on the encoding of more explicit knowledge so that the house might be decomposed into building parts for manufacturing. This requires the

development of appropriate building systems, the development of rules encoding such systems, and the development of algorithms to list the parts and to generate the required information for manufacturing them.

Once these steps have been completed, the full potential of the envisioned framework will be achieved. Then, it will be possible to customize mass housing at an affordable cost, which will constitute a major social contribution.