Graph, Geometry, and Language

A design triangle

21

Weiling He
Georgia Insitute of Technology, USA

Abstract

The syntactical concern of architectural form focuses on its structure and related measure and meaning. It is essentially a process of deriving architectural form of its specific geometry and language so that the topological structure is revealed. In this study, the idea of the graph, one of the syntactical concerns, is taken as the starting point. Geometry and language, in relation to graphs, are taken into account later on. In simple terms, this study follows this direction, as opposed to the original syntactical research. As a result, the triangular relationship between graph, geometry and language is framed. In the triangle composed of graph, geometry and language, the relationship among the three components is defined as the constraints and possibilities that each of the components provides to the forms of the others. A sequential discussion is carried on so that the trilateral relationship is revealed step by step.

Keywords:
Graphs, design language, underlying geometry,interim geometry, language

Weiling He,
College of Architecture, Georgia Institute of Technology
Atlanta GA 30332, USA
weiing.he@arch.gatech.edu

Introduction

Design involves a three-way interaction between 1) abstract, topological-like, spatial relationships, 2) geometrical frameworks, and 3) specific languages that govern the bodily shape of the building. Within the framework of space-syntax, abstract relationships are typically expressed as graphs. The interpretation of geometrical constraints depends upon the assumptions made regarding the mapping of layouts into graphs. The mapping is typically mediated by linear, convex or visual-polygon based representations of layouts. Languages, in the sense in which the word is used here, are about the precise dispositions of physical boundaries so as to generate the linear, axial or visual polygon fields that are subsequently analyzed as graphs. This paper proposes to discuss, in a rather elementary way the three way interaction of graph, geometry and language, in order to demonstrate how it sets up a system of constraints. From a broader point of view, the exercise is of interest because it offers a preliminary insight of how we may model not so much design formulation – the creation of formal design aims – but rather design discovery – the exploration of possibility in a structured manner subject to formal aims and mathematical constraints – as part of design intentionality.

For the purposes of this argument, geometry will be approached from the point of view of the s-partition[1] . Because the s-partition is closely associated with the disposition of physical boundaries, it provides an unambiguous manner for linking an abstract graph to a

[1] According to Peponis et al, an s-partition is the pattern of s-lines that are the extended segments of extendible surfaces in a space.

specific built shape. We will be asking how to generate layout shapes whose s-partition complies with a previously defined graph. Having dealt with this – the relation between graph and geometrical realization – we will then explore two languages for shaping the layout: cellular on the one hand, and linear on the other. In fact, the cellular language is the abstraction of the classical cellular plan while the linear language is the abstraction of the modernist open plan. The reason why these two languages are chosen is to capture the modernist debate on the open plan. That is, whether or not it makes any difference to their compositional outcomes when architect reject cellular plan. In the cellular language every s-space[2] is fully enclosed by walls, all corners are built, and openings take the form of traditional doors. In the linear language walls are either free standing or intersect as t-junctions. No cross-junctions are allowed. Consequently, some of the corners of s-spaces are defined as intersections of s-lines, of walls or as intersections of a s-line and a wall. Also, co-linearity is not allowed in linear language, which can trace back to some extreme examples of open plans. It will be shown that some graphs can be realized in one language only – the cellular – while others can be realized in both. In this manner, the pattern of constraints arising from the interaction of graph, geometry and language will be heuristically illustrated. However, this paper also discusses some of the theoretical and methodological issues involved with the systematic exploration of the subject. It does so by dealing with a particular underlying geometry, that of two-dimensional grids.

Let us do two simple experiments. In the first experiment, suppose we have a graph of a ring, made up of four nodes, to be realized in the following geometry made up of four elementary rectangles (Fig.1-1). Also suppose that we have those two kinds of languages as

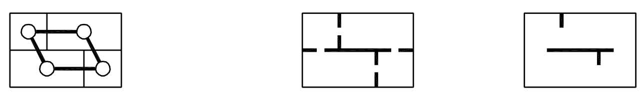


Figure 1-1. A Graph Realizable Both in Cellular Language and in Linear Language.

we mentioned above, the cellular language and the linear language, to apply. By an easy heuristic exploration, we come up with two designs in both languages that realize the graph in the geometry. In the second experiment, suppose we change to realize a four-node graph, but intend to add a diagonal link. This graph can easily be articulated in cellular language. However, it is impossible for linear language in that specific geometry. All possible designs that we can get violate the definition of non-colinearity of the linear language (Fig.1-2). These two experiments show that some combination of the graph, geometry and language is realizable while some are not. There is a triangular system of constraints, set between the graph, geometry and language, that controls this realizability. One question rises at this point, how could we create a framework for systematically exploring solutions and for confirming whether a graph is realizable on a geometry and according to a language? This question will be explored step by step in the following parts.

[2] An s-space is the elementary convex polygon defined by s-lines and walls in s-partition.

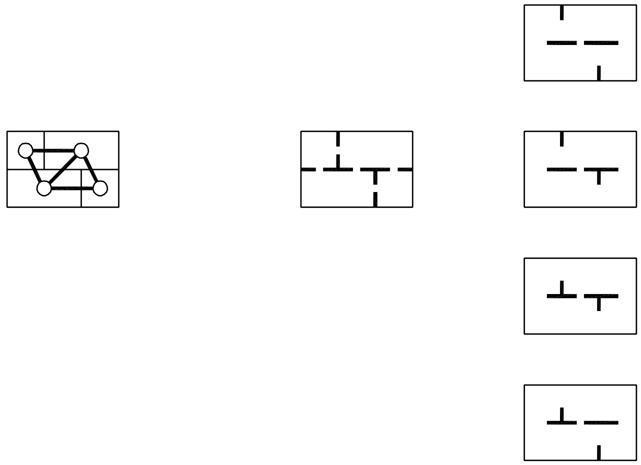


Figure 1-2. A Graph Realizable in Cellular Language but Not in Linear Language.

1. Graph

A graph is the starting point of the exploration of the interaction between the graph, geometry and language. It gives us two kinds of information: the number of nodes and the connectivity of the nodes. In the architectural interpretation, a node in a graph is a space in the final design so that the status of connectivity among those nodes is the potential connectivity of the spaces in the final design. The realizability of a graph can only be defined on the condition that the spatial limitation, within which this graph is supposed to be realized, is defined earlier. For instance, a graph may be realizable in a three dimensional space but not in a two dimensional plan. A graph can also be realizable in a triangular arrangement of space instead of a rectangular one. We will discuss the realizability of graphs in relation to our definition of geometry in this study, the orthogonal grid.

2. Graph and Geometry

J. P. Steadman (1983) explores the relationship between graphs and geometry and summaries the research of himself and others in his book, *Architectural Morphology*. He discusses a systematic way in which plans satisfying the size constraints can be developed from a given graph. Although our study does not possess the same mathematical rigor as Steadman's, some key issues related to the triangular relationship between graphs, geometry and language will be discussed.

In our study, the geometry taken as a point of departure in the previous two examples, will be treated as an "interim geometry." An "interim geometry" is defined as a pattern of quadrilaterals on an underlying grid, which represents both the walls and their extensions that are drawn to create the s-partition, without distinguishing them. As some parts of the linear segments that make up the geometry are subsequently interpreted as walls, to give rise to the built shape, no s-line should arise which is not already present in the "interim geometry". This is the fundamental constraint. "Interim geometry" also involves Steadman's idea of "dimensionless gratings."

The relationship between the interim geometry and the graph can be illustrated by the relationship between a more fundamental geometry and a more fundamental graph. A two-dimensional plan, which we are interested in, can be mapped as a grid made up of elementary units. Correspondingly, the grid of 2-D space can be translated as a graph, in which each of the nodes refers to a unit in the grid and each of the links between nodes refers to an edge shared by units in the grid. This basic graph can be transformed into others with more complicated structures in two ways: merging adjacent nodes or cutting links. In the first case, the original links between those adjacent nodes are eliminated, and other links between those adjacent nodes and other nodes are summarized as the links to the one node after the merging. To represent this transformation in the underlying geometry, we are in fact merging the units into a bigger rectangular shape. By definition, only rectangular spaces are allowed in the geometry. Any merging of the nodes resulting in non-convex shapes in the geometry is illegal. Thus, we will not end up with an "L" shape or a "T" shape or any other non-rectangle in our geometry (Fig.2). When the process of merging nodes is completed, we get both a desired graph embedded in an interim geometry. In this process, the underlying geometry acts as the bridge between the desired graph and the interim geometry. In the second case, a link can only be cut by the body of a wall if both nodes related by the link remain. We will discuss this case in the application of language in the next section.

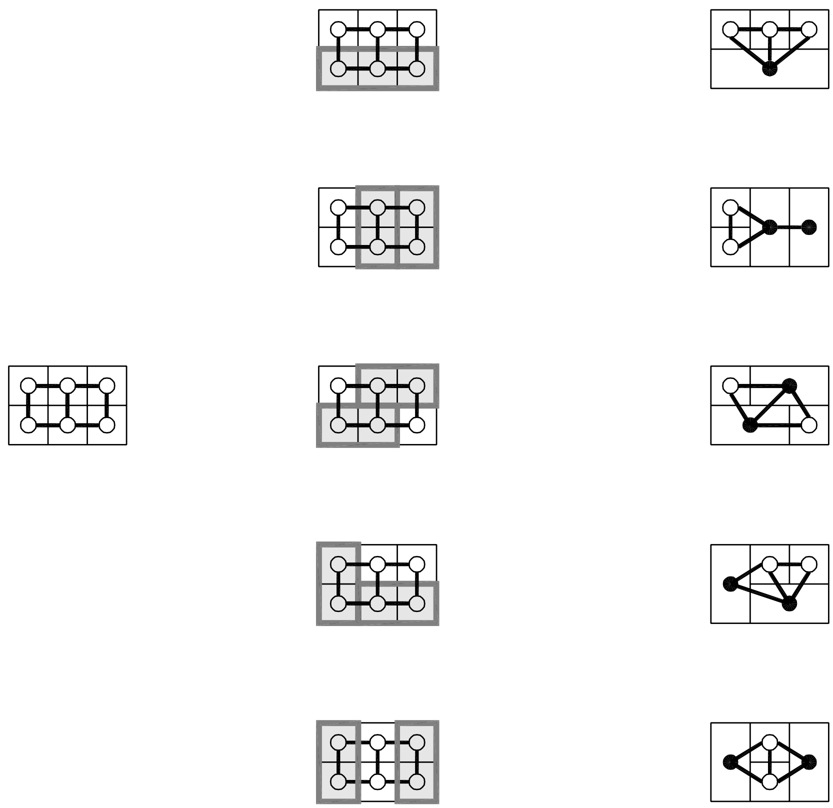


Figure 2. Generating Interim Geometry by Merging Nodes.

2.1 A Realizable Graph

Having achieved the above understanding, we obtain the basis to examine the realizability of a graph. Let us define several variables first. A basic graph composed of the total number of nodes as $m \times n$ can be derived from an $m \times n$ grid. L_m is the maximum number of links that an $m \times n$ grid can provide. L_m' is the maximum number of links in the geometry after merging. N is the number of nodes after merging. K is the number of links that are eliminated by merging the nodes. Thus, L_m, the maximum number of links that an $m \times n$ grid can provide, is shown as below.

$$L_m = 2mn - (m + n) \quad (1)$$

A certain number of nodes can only fit a maximum number of links if this graph is to be realized in a rectangular plan by our definition. Diagonal links are not possible, which is easy to explain in real-life architecture since two units cannot be connected by their shared angle.

Once a node is merged, a link, between this node and the one that it merges to, is eliminated. This one to one relation in number makes it possible to know the maximum number of the links in the geometry, L_m', on condition that the number of the nodes in the basic grid (*m by n*) and the number of spaces in the geometry after merging, N, is given.

$$K = mn - N \tag{2}$$

$$L_m' = 2mn - (m + n) - K \tag{3}$$

$$L_m' = mn - (m + n) + N \tag{4}$$

In other words, once a graph is given, we can tell, by fitting its number of nodes and number of links into the above theorem, if this graph is realizable in a rectangular plan by our definition.

2.2 The Size of the Underlying Geometry

Back to the starting point, given a graph, how would we know in which underlying geometry to look for the interim geometry? That is, what are m and n if N is known?

It is easy to understand that the minimum value of mn needs to be equal to N. Asking for the maximum value of mn without any further constraints is meaningless since any enormously big grid can always be merged as one node. However, there exists a maximum value of mn if one requirement is added: none of the dividing lines in the original grid is totally eliminated after the merging of the units. This maximum value has been proposed by Steadman as the theorem:

$$m + n - 1 = N \tag{5}$$

Providing m and n are integers,

$$mn = \frac{(m + n)^2 - (m - n)^2}{4}$$

$$mn_{max} = \frac{(N + 1)^2}{4} \tag{6}$$

Obviously, the size of underlying grid ranges from N to mn_{max} (when m is equal to n) and the grid can be any two factors whose product are equal to any value of this range. It is important to note that all the values discussed above are integers.

2.3 From Underlying Geometry to Interim Geometry

We will borrow Bloch's method (1976) of dissection as a systematical way of dividing the underlying geometry along the grid-line by enumerating all the possible ways. Further, we will define the constraint that applying four-way dissection has less priority than other dissections. By doing this, we prevent those duplicated cases from emerging in a bigger grid, which are isomorphic forms of the dissections in some smaller grid. For example, if we allow four-way dissection in a 3 by 3 grid to generate five spaces at the end, we will come up with those dissections, as shown in figure (Fig.3 see over), which we can get by dissecting a 2 by 3 grid. According to the definition of this constraint, there is no need to apply any four-way dissection in order to achieve the desired number of spaces.

We also propose a pinwheel dissection, which is more suitable to apply in our underlying geometry, as opposed to the additive method mentioned by Steadman (Fig.4 see over). The first step is to locate the core. If the underlying geometry is big enough, we may have a core

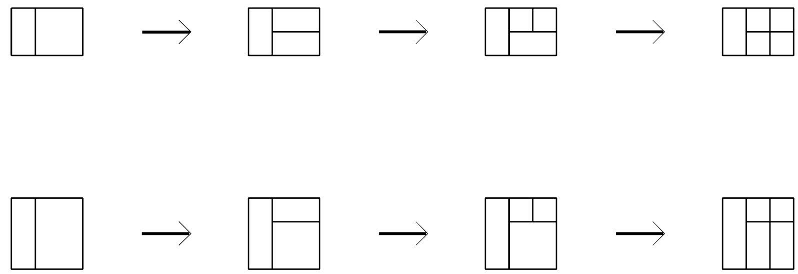


Figure 3. Duplicated Cases.

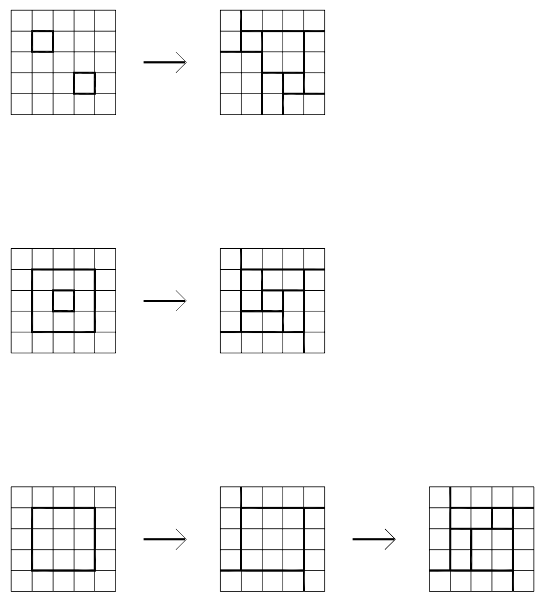


Figure 4. Three Examples of Developing Pinwheel Structure.

of more than one unit of the grid or we may have more than one pinwheel core. A pinwheel is realizable in a grid on condition that the core of the pinwheel is located at least one row off each side of the grid. The second step is to extend the four sides of pinwheel core until they reach the edge of the grid or the extensions or bodies of other pinwheel cores. Thus, in the case of multiple pinwheels, we need to enumerate all the cases with priority of extending different pinwheels. The third step following this is the regular dissection until the number of spaces is satisfied. The dissection can be applied in any leftover spaces off the pinwheel, or in any pinwheel core if the core is big enough.

2.4 Referring to the Original Graph

According to the two kinds of information that a graph provides, a geometry is related to the graph in a two-fold manner. On the one hand, the number of nodes in a graph determines the size of the underlying geometry and thus its substratum, the interim geometry. On the other hand, connectivity shown in a graph determines the legality of the geometry that has already satisfied the size requirement. Thus, in the move from the original graph to the interim geometry, we seem to explore possibilities when following the move from a graph to an underlying geometry, and then to an interim geometry. In fact, there are constraints, as determinant as possibilities, shown in this process, which put the relationship between graph and geometry into clearer light.

First of all, some graphs are not realizable in some geometries because of the embedded constraints. For example, a *1* by *n* underlying grid cannot realize any graph with a ring structure, nor can it realize a one-to-multiple connection. The only possible graph embedded in the *1* by *n* underlying grid is a sequence. As the overall configuration of the grid grows more square-like, the more possible graphs can be fit in. (There must be a limit.)

Second, the interim geometry is embedded with graphs that have the same number of nodes but different patterns of links. In the case of fitting a five-node graph into geometry, only the 3 by 3 is capable of expressing all the possible graphs made up of five nodes. Others, such as a 2 by 3 and a 2 by 4, are not. Only certain graphs can be realized in these two underlying geometries. Even though we trace the children of these graphs the possible graphs realizable in a 2 by 3 or a 2 by 4 grid are only a part of the complete matrix. The geometry enforces constraints on its realizable graphs.

3. Geometry and Language

In *Architectural Morphology*, Steadman reports on research that attends to examine all possible embeddings of a graph within a grating which is the dimensionless representation of a floor plan. Similarly, in the previous section, we explored the possible underlying geometries and the interim geometries when a certain graph is given. In this section, we will impose two languages on interim geometries to see how the definition of language constrain the realizability of a graph in certain geometry and how the given geometry limit the possibilities of articulating a language.

3.1 The Constraints of Geometry

As soon as the interim geometry is defined, the location of spaces in the final design is defined as a result. However, the structure of spaces has not been determined completely yet. Here, we move to a new phase in the design triangle, the articulation of language from the interim geometry. According to our earlier definition, language in this study is expressed as the arrangement of walls. What matters in this step is not merely the look of the plan owing to different wall languages, but also the permeability derived from various arrangements of walls even though they are based on the same underlying and interim geometry. In this way, the structure of spaces in the final design is defined completely.

The use of certain language in the interim geometry is constrained by both the configuration of the geometry – the s-space – and the definition of language. We have three basic cases to deal with in a rectangular geometry, 1) a dividing line without any junction; 2) a three-way junction; and 3) a four-way junction. Examining the configuration of the interim geometry, we find joints of those rectangles implicate the arrangement of walls if this configuration is to be preserved. In the first case, the way to develop a wall is to place at least one wall on the non-junction dividing line. In the second case, a three-way junction can only be possible if there are walls to define each of the junction lines and, at the same time, if the body of one of the walls blocks the body or extension of the other. The third case requires that either the bodies of the walls form a cross or the extensions of the walls form a cross. No other possibility exists.

According to the above examination, we find it possible that geometry limits the possible location of walls in some language, such as the linear one, more strictly than in some other language, such as the cellular one. Take a combination of two three-way junctions in interim geometry for example. In order to follow the way to preserve both of the junctions, in cellular language, one needs either to place a wall which blocks the bodies of the other two walls, or to place two walls that block the extensions respectively. However, in linear language, the only way to realize the three-way junction is to place a wall to block the extensions of the other two walls (Fig.5). After this move, in linear language, the permeability is eliminated in between the two three-way junctions while, in cellular language, the permeability may or may not be eliminated.

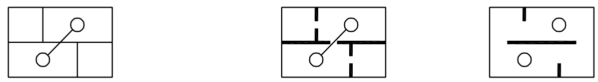


Figure 5. Translating Three-way Junction into Cellular Language and Linear Language.

3.2 Unrealizable Geometry

It is not only the case that geometry constrains the articulation of language, but also the case that language limits the choices of geometry to which the language applies. In this determinant loop, the definition of language plays an important role.

According to our definition, co-linearity is not allowed in linear language. Thus, any geometry with co-linear partition cannot be realized. That is because co-linearity will be the result if we try to preserve the original geometry after applying the language. Compared to the linear language, the cellular language seems to possess more flexibility. However, it depends on the definition of cellular language. Suppose the cellular language is defined in a strict classical sense, with symmetry required for the overall plan. As a pre-requirement, only symmetrical geometry is proper for this strict cellular language. Further, if proportion is taken into account as a crucial measure, only geometries embedded with certain proportions as well as being symmetrical are legal. On the other hand, if we push the definition of linear language to the extreme that symmetry is not allowed, a new constraint will be placed on the choice of geometry. The realizability of design depends intensely on the definition of the language.

4. Language and Graph

It is easy to discuss the graph-geometry and the geometry-language relationship in the triangle of the graph, geometry and language since one sees the direct move from one to the other in these relationships. Geometry seems like the mediator in between the graph and language. One might ask at this point, what can we gain if this mediator is removed? Is there any way in which language reflects itself in the graph, or graph in language?

4.1 The Graph embedded in Language

The cellular language seems to impose few constraints on graphs if the definition of the cellular language is as we proposed in the first section. The connections between two spaces can be as many as possible as long as an equal number of doors are created. Or, the connection can be zero if no door is there. However, it is a totally different case in linear language (Fig.6). Since co-linearity is not allowed, the maximum number of connections between two

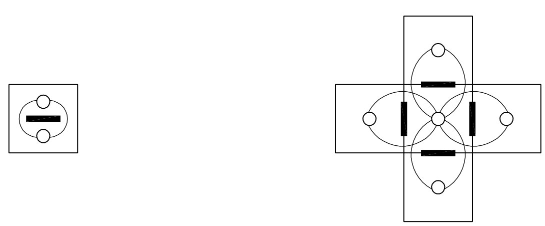


Figure 6. Maximum Graph Embedded in Linear Language.

spaces is two if the wall is located in such a way that two openings are left at each end of the wall. Expanding this measure of one unit, we can easily find that, in linear language, no space can have more than eight connections. That is shown as follows. A rectangular space (as we defined earlier we will only discuss rectangular plans in this study) has four edges. Suppose this space has the maximum number of connections on each edge, the maximum total number is eight. Owing to these constraints, some graphs are impossible if certain language is used although these graphs could occur in certain geometries.

4.2 The Graph Lattice

Given a graph with a certain number of nodes, there exists a maximum number of links that can be realized in this graph if certain constraints are defined. Starting with the graph being in this state of maximum links, variations of it with the same number of nodes but fewer links can be generated. Figure 7-1 shows the graphs made up of five nodes. Those numbers shown on each of the levels indicate the number of links each group of nodes has. The only restriction is that, in a graph, all five nodes need to be connected to form a linked whole. That is, one should be able to travel from one node to any other node through the links in the graph. Thus, there needs to be at least four links to connect the five nodes. That is why our graph group stops at the level of four links. As it is shown here, it is probable, with both five links and four links, that some graphs cannot satisfy the restriction of connecting each of the five nodes into a linked whole. Another point is, although we did not count those isomorphic forms caused from symmetrical transformations, we still have those isomorphic forms resulted from topological transformations.

By placing these generated graphs on different levels based on the derivability of one from the other, a lattice can be derived. Figure7-2 shows certain links of the lattice. Since, in the design triangle, graphs constrain geometry and language, the parent-child relationship shown in the lattice not only articulates the relationship between graphs but also indicates the relationship embedded in geometry and that embedded in language. To put it in more detail,

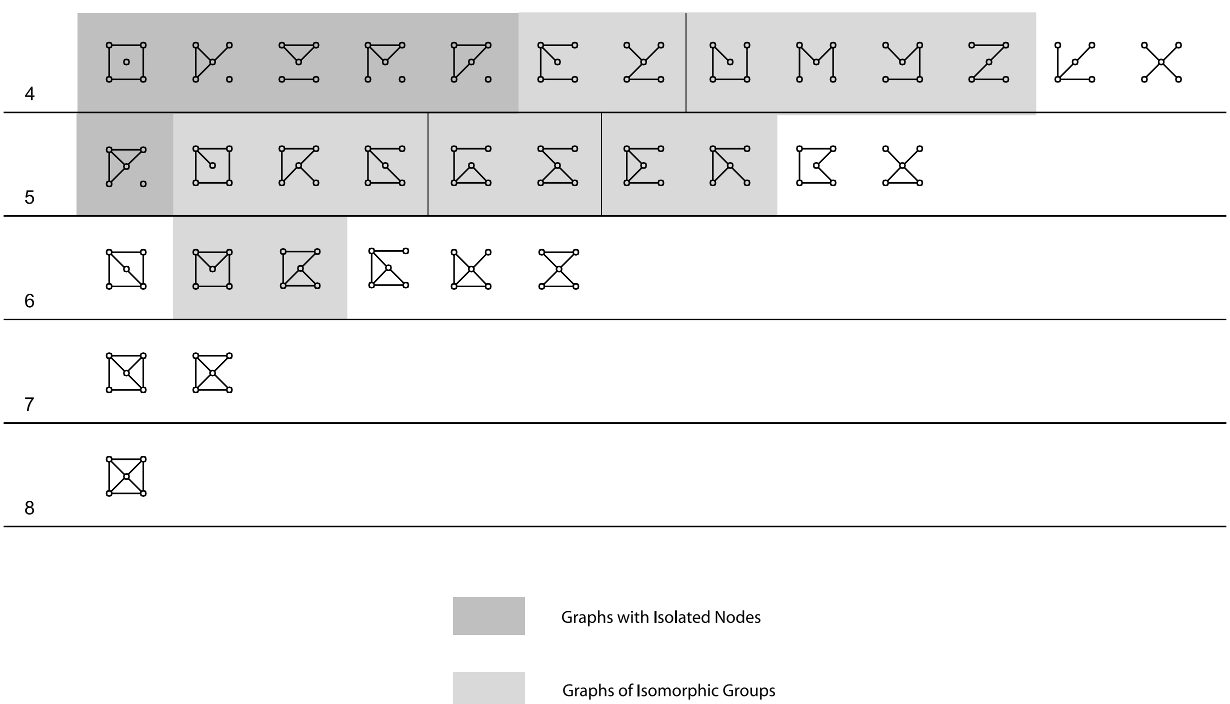
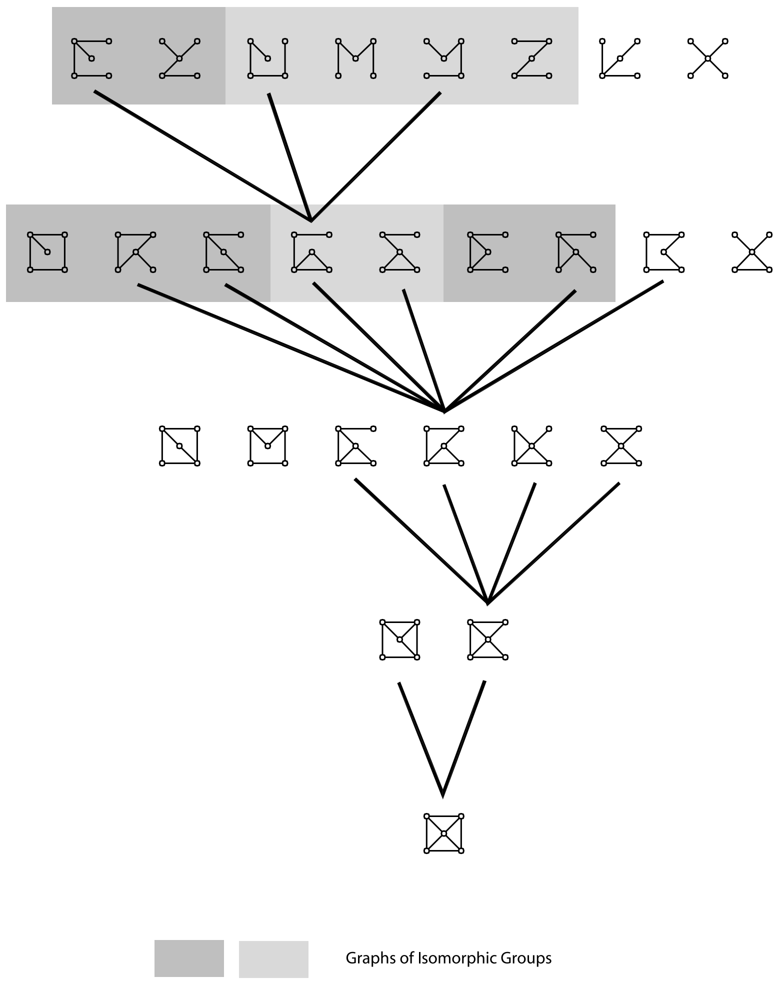


Figure 7-1. Graphs Made up of Five Nodes and Their Levels Defined by the Number of Links.



the interim geometry provides an embedded graph with the maximum number of links. When a certain language, for example the linear language, is applied, some links are eliminated. In Steadman's vocabulary, this is a move from an adjacency graph to a permeability graph. After the permeability graph is determined we can search, through the parent-child relation shown in the graph lattice, for all the other graphs which are descendants of this maximum one. Since to eliminate a link is always possible in both cellular and linear language, it is true that all those children graphs can be realized (Fig.8). Further, some branch of the graph lattice does not have any shared members with other branches, which means that a parent graph might never have certain child graphs even though the parent graph possessed more links than the child graphs. These graphs belong to different families. The parent-child relationship shown in the graph lattice shows the syntactical relationship between the designs in a certain language.

Figure 7-2. The Parent-child Relationship in the Graph Lattice.

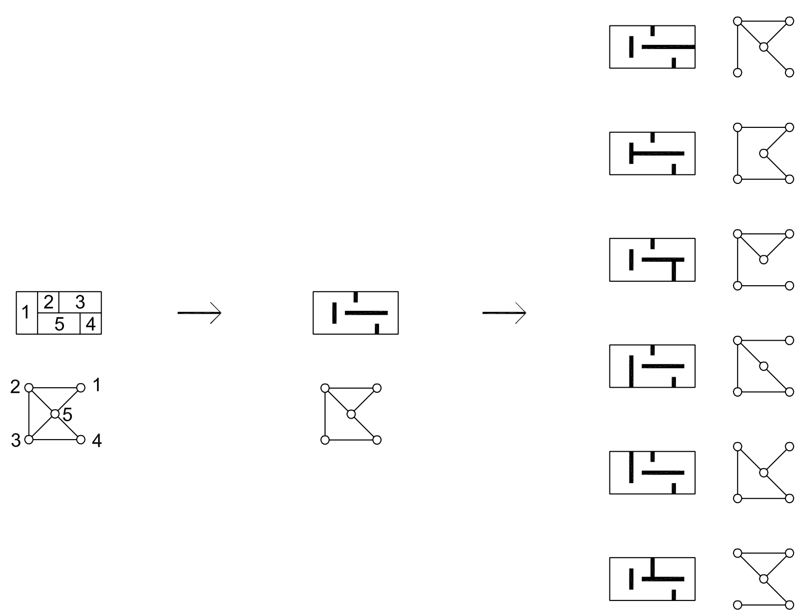


Figure 8. Designs of the Parent Graph and Children Graphs.

5. The Triangle as a Question

In the triangle, we move from the graph through geometry to language, and then back to the graph. This move explores the possibilities of realizing design intentionality in a structured manner. From the point of view of space syntax, the graph implies culture since those values shown in a graph, such as connectivity and depth, tell the function of a space in the whole structure. Geometry represents the idea of genotype, which is a basic layout of spaces. Language illustrates the intention of an architect. That is, what specific means he/she intends to choose to articulate the genotype of a certain culture. The triangular move is not a model of design formulation. Instead, it proposes the question of possibilities and constraints shown in the relationship between the graph, geometry and language.

Through the three-step move, space is defined more and more strictly. In the first step of generating a graph, the aspatial relationship between spaces are defined. That is, only the number of spaces and their status of links are certain. In the second step, deriving geometry from the graph, the adjacency of the space is determined with the constraints of convexity and orthogonality. Geometry constrains the possible ways in which a graph can be realized, and it also limits what kinds of graphs can be realized. Thus, some graphs are always impossible in certain geometries while some other graphs are only possible in their corresponding geometries. When we reach the third step, articulating language in geometry, the space and its structures are complete. From a syntactical point of view, the articulation of language is more about determining the permeability of the space than about determining the style. It is worth noting that the definition of the language also has its impact on the geometry. Even though some geometry has its maximum potential permeability, due to the definition of language this maximum case may not be possible at the end. As a result, we find many impossible triangles composed of a certain graph, geometry and language. Although every instant move, from graph to geometry and from geometry to language, is possible not every overall relationship is possible. That is, no design can be embedded in a specific graph in a certain geometry with a certain language at the same time. The issue here is not about the design process but about speculating on the relationship between the graph, the geometry and the language of design. Thus, in figure 9 we do not show the process of applying a specific graph into designs. Instead, we explore those cases where certain five-node graphs can be embedded in certain geometries in certain languages. The left part of the lattice shows a systematic way of searching for all the possible geometries that can embed graphs with five nodes. The column on the right of the figure shows some possible designs in linear language, rather than all the possibilities. After retrieving and examining the five-node graphs from the final designs we find that the relationship between graph and geometry is not exclusively a one-to-one relationship. It may be one-to-many or even one-to-null. That is also the case for the relationship between the graph and the final design in a certain language.

Syntactic representations of layouts involve two steps: first, the translation of a plan into a set of discrete 1D or 2D elements, and second the analysis of the graph representing the relationships between these elements. This paper demonstrates that the reverse process, going from an underlying graph to an architectural plan, has to likewise be conceptualized in two steps. First the graph is mapped into a robust arrangement of syntactic-geometrical elements. Second an actual plan is constructed whose analysis would retrospectively produce the same syntactic-geometric elements – where the s-partition has been used as the underlying syntactic arrangement and s-spaces have been treated as the syntactic elements. The second step can be subjected to rigorous rules, a "language" of design. This schematization has only heuristic value. In reality, the manner in which the syntactic-geometric elements are drawn and the possible languages for deriving plans have to be considered in conjunction. For example, if the s-partition is chosen as the syntactic-geometric basis, then only certain distinctions between design languages can be usefully explored. So, the main aim of the paper is to illustrate some of the dilemmas that have to be tackled if the graph-descriptions associated with space syntax are to be interpreted as plan generators. The underlying question of turning graphs into geometrical arrangements has been well familiar to Steadman and March. The potential contribution of syntactic theory resides in theorizing the appropriate intermediate representations between graph and competed plan.

Acknowledgement

Many thanks to Dr. John Peponis, whose inspiration and help go far beyond what is presented in this paper. Thanks to Dr. Athanassios Economou, who helped add an element of rigor to this study. Also, thanks to Lord, Aeck, Sargent Architects, whose support makes my studies at Georgia Tech possible.

Bibliography

Bloch C J, Krishnamurti R, 1978, "The counting of rectangular dissections" *Environment and Planning B* **5** 207-214.

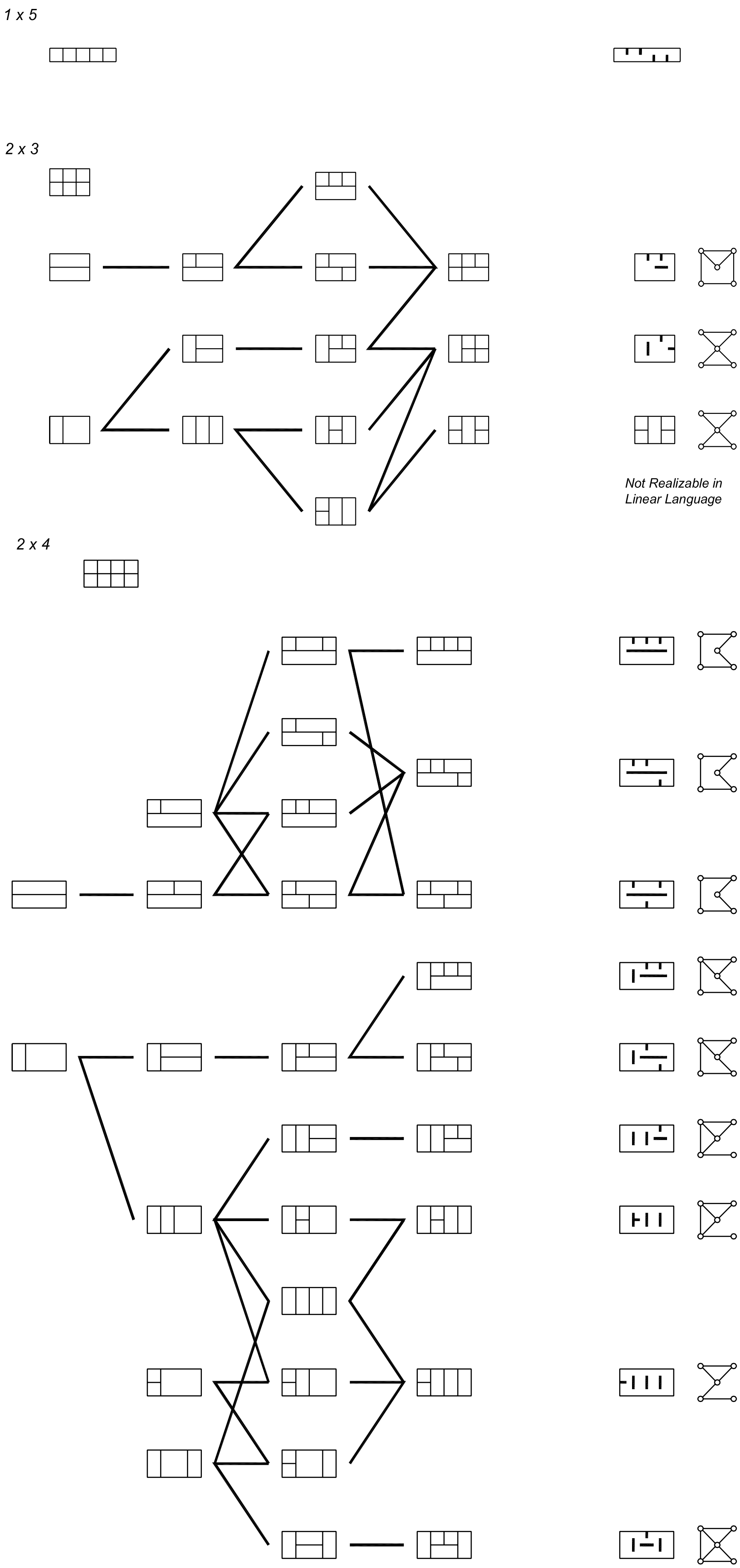
Hillier B and Hanson J, 1984, *The Social Logic of Space*, Cambridge [Cambridgeshire] ; New York : Cambridge University Press.

March L, and Steadman P, 1971, *The geometry of environment: an introduction to spatial organization in design*, London, RIBA Publications.

Rashid, M, 1998, *On the Configurational Studies of Building Plans from the Viewpoint of A Situated Observer: A Partial Theory of Configuration for Plans Not Involving Curves*, Georgia Tech Doctoral Thesis.

Steadman J P, 1983, *Architectural Morphology: An Introduction to the geometry of building plans*, Poin Limited.

Peponis J, et al, 1997, "On the description of shape and spatial configuration inside buildings: convex partitions and their local properties" *Environment and Planning B: Planning and Design* 24 761-781.

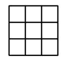


Appendix

This group of figures illustrates a systematical way to explore the possible designs of a five-node graph in certain geometries in linear language. In the first step, according to the theorem we proposed in the paper, we come up with four kinds of grid: 1 by 5, 2 by 3, 2 by 4 and 3 by 3. In the second step, dissection and pinwheel dissection are applied so that certain interim geometries are defined. In the third step, linear language is applied to each of the interim geometries. Not all possible designs in linear languages with five-node graphs are explored, however, examples are given to each interim geometry. Also, to simplify the problem, we only allow, at the most, one connection between two spaces.

(To be Continued)

3 x 3



21.14