Programming the Logic of Space: A new approach to space syntax

Pedro Afonso Fernandes, *[0000-0001-5762-5157], Universidade Católica Portuguesa, Católica Lisbon School of Business & Economics, Católica Lisbon Research Unit in Business & Economics (CUBE), Lisbon, Portugal. paf@ucp.pt*

Abstract - *Space syntax is a set of theories and techniques for analysing urban settlements and buildings. It is a relevant approach for spatial studies concerned with the scientific understanding of and intervention in public open spaces. It requires a previous convex/line/segment map of the area in question designed with the aid of CAD or GIS. Such preparatory task may be laborious and more difficult for social researchers than for architects or urban planners. A new approach to syntactic analyses is proposed here; it requires only setting the connections between axial lines or convex spaces in a computer program. We performed a settlement (alpha) analysis of the Areeiro neighbourhood in Lisbon by computing connectivity, control, depth, integration, ringiness and other syntactic measures, using Prolog predicates. Prolog is a logic programming language used in Artificial Intelligence and a powerful tool to describe the patterns of urban systems as socially knowable due to its declarative nature. In fact, a Prolog program represents a certain amount of knowledge, in this case, of an urban settlement such Areeiro, which is used to answer queries about the social and economic consequences of the spatial design. This approach revealed the intelligibility but also the weaknesses of Areeiro, a remarkable milestone for Portuguese urbanism from the mid-20th century.*

Keywords - Space syntax, alpha-analysis of urban settlements, logic programming, Prolog

Resumo - *A sintaxe espacial (space syntax) é um conjunto de teorias e técnicas de análise do ambiente construído. É uma abordagem de base científica relevante para os estudos urbanos e para a intervenção no espaço público. Como passo prévio, as análises sintáticas requerem um mapa convexo, axial ou de segmentos desenhado com recurso a um sistema CAD ou SIG. Esta tarefa preparatória pode ser exigente ou difícil, nomeadamente, para investigadores das ciências sociais, sem formação em arquitetura ou urbanismo. Neste capítulo é proposta uma nova abordagem sintática que apenas requer a declaração, num programa de computador, das conexões entre linhas axiais ou espaços convexos. Desta forma, foi possível realizar uma análise do espaço urbano (análise alfa) do bairro do Areeiro, em Lisboa, mediante o cálculo de medidas sintáticas como a conectividade, o controlo, a profundidade, a integração ou o coeficiente anelar (ringiness) com recurso a Prolog, uma linguagem de Programação em Lógica utilizada em Inteligência Artificial. Fruto a sua natureza declarativa, a linguagem Prolog é uma poderosa ferramenta na descrição dos sistemas urbanos* *enquanto entidades socialmente cognoscíveis. De facto, um programa Prolog é um repositório (base) de conhecimento, por exemplo, sobre um espaço como o bairro do Areeiro que depois é utilizado para responder a questões concretas sobre as consequências sociais e económicas de determinado desenho urbano com recurso a lógica. Esta abordagem revelou a inteligibilidade, mas também os pontos fracos do Areeiro, uma realização, em todo o caso, notável do urbanismo português de meados do século XX.*

Palavras-chave: Sintaxe espacial, análise alfa do espaço urbano, programação em lógica, Prolog

Smaniotto Costa, C., Aragão, N. (Eds.): Understanding and transforming the territory, Culture & Territory 05, pp 93-109, 2023
https://doi.org/10.60543/ecati/ejn5-2w03

INTRODUCTION

Space syntax is a set of techniques for analysing urban settlements and buildings, as well of theories linking space and society founded on Architecture, Engineering, Mathematics, Sociology, Anthropology, Ethnography, Linguistic, Psychology, Biology and Computer Science. It was developed originally by Bill Hillier, Julienne Hanson and colleagues at the Bartlett School of Architecture and Planning, University College of London (UCL), since the 1970s. Their innovative approach was condensed in three landmark books: *The Social Logic of Space* (Hillier and Hanson, 1984), *Decoding Homes and Houses* (Hanson, 1998) and *Space is the machine: a configurational theory of architecture* (Hillier, 2007).

An important contribution of space syntax is the study of urban form using *discrete systems*, that is, computer-aided recursive techniques based on *elementary generators* such as a pair of open and closed cells. Another key contribution is the concept of *depth* in urban systems and its relationship with social phenomena like pedestrian movements or crime incidence, with special focus on different uses of open spaces by inhabitants and strangers, men and women, old and young, adults and children, and so on (Hillier, 2007: 146-152). Anyway, space syntax has influenced and contributed to several projects, namely, the redesign of Trafalgar Square or the location of the Millennium Bridge, both in London. Thus, space syntax is a relevant and universal scientific approach for spatial studies, namely those concerned with the understanding of and intervention in public open spaces.

Space syntax is mainly concerned with the study of the relations between *convex spaces*. Convexity exists when straight lines can be drawn from any point in a space to any other point in it without going outside of the space itself (Hillier and Hanson, 1984: 97-98). In fact, convex spaces rather that concave stimulate social interaction in the sense that everyone sees everybody within that kind of space. Thus, the starting point of a syntactic analysis is typically a map of the fattest convex and open (or permeable) spaces that cover the settlement (or building) in question. Eventually, this framework can be represented by an axial map with the smallest set of straight lines that pass through all convex spaces or by a map of segments. Then, several syntactic measures can be computed to explore the connectivity between spaces and lines or the angular deviation between segments in order to evaluate the degree of asymmetry (or integration) and distributeness of each convex space/line/segment or of the whole system (Heitor and Pinelo Silva, 2015: 154-164).

Space syntax analyses are typically performed with Depthmap (Turner, 2004) and similar software (Heitor and Pinelo Silva, 2015: 167-169). In these packages, the input is the convex/axial/segment map with the spatial configuration of the complex in question, previously designed with Computer-Aided Design (CAD) or Geographic Information System (GIS) software. This preparatory task may be laborious and difficult, namely for researchers from social and economic sciences, which as a rule limits the dissemination of space syntax techniques beyond architects and urban planners. Here, we propose a new approach to perform syntactic analyses that do not require a previously designed map, but only the declaration in a Prolog program of the connections between axial lines or convex spaces of a settlement or building, given a simple map/plant of it.

Besides the practical goal stated, this innovative approach aims to capture the essence of space syntax. Prolog is a *Logic Programming* language developed by Alain Colmerauer, Philippe Roussel, Robert Kowalski and colleagues between Montreal, Marseilles and Edinburgh in the 1970s (Colmerauer and Roussel, 1993). It uses logic to represent knowledge and uses deduction to solve problems by deriving logical consequences (Kowalski, 1988: 38). A corollary of using logic to represent knowledge is that such knowledge can be understood declaratively. This kind of reasoning is embodied in space syntax when it represents spatial arrangements as a *field of knowables*, that is, as a system governed by a simple and abstract underlying system of concepts (Hillier and Hanson, 1984: 66). In fact, clump, concentric, estate and other syntactic processes defined in the book *The Social Logic of Space* are recursions of simple relations between open and closed primary cells (Hillier and Hanson, 1984: 78). Prolog is recursive by construction and uses relational clausal logic in a straightforward way to declare binary concepts like *containment* or *adjacency* used to define the elementary generators of space syntax processes. Thus, the social (and economic) logic of space can, and ought to be, represented by a logic and declarative programming language such as Prolog.

Here, the application of Prolog to space syntax is illustrated by the case of the Areeiro urban settlement in Lisbon, Portugal. This remarkable borough was planned in 1938 by João Guilherme Faria da Costa [1906-1971] for 9,000 inhabitants with 2,680 dwellings and it was constructed in two phases starting in 1940 and 1948 (Lamas, 1992: 284). It had resulted from mayor Duarte Pacheco's aim to expand Lisbon to the north-east on expropriated land with a strong public control. Both Areeiro and Alvalade, also planned by Faria da Costa (in 1945), are good examples of settlements that combine traditional urban forms with modern concerns, namely, a hierarchy of streets or the implementation of public facilities (schools, pools, churches, gardens) in neighbourhood units.

This chapter is organised as follows: in the theorical framework, we briefly describe the key concepts of space syntax with a focus on configuration, depth and elementary generators. In the methodological section, we show how syntactic measures can be computed with Prolog in an elegant and straightforward way, drawing on the Areeiro neighbourhood as case study. Then, we present the main results of the syntactic analysis of this neighbourhood. Finally, some conclusions and future developments are pointed out.

FRAMEWORK

Based on graph theory and computer-aided simulations, *space syntax* aims to find and explain the relation between spatial configurations and social activities. *Configuration* is a concept addressed to the whole of a complex (settlement or building) rather than to its parts that captures how the relations between two spaces, say A and B, might be affected by a third space C (Hillier, 2007: 23-24). For instance, if A and B are adjacent or permeable, then they have a symmetric configuration in the sense that if A is neighbour of B, then B is neighbour of A, as illustrated by the left-hand side of Figure 1.

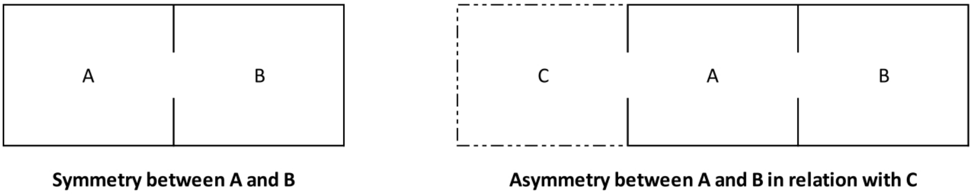


Fig. 1: Basic configurations. Adapted from Hillier (2007: 24)

However, if only A is connected with a third space C, as in the right-hand side of the same figure, A and B become asymmetrical in relation with C, because we have to go through A to get to B from C, but we do not have to go through B to get to A from C. Thus, *asymmetry* relates to *depth*, that is, to the number of spaces (or steps) to go from some space, say C, to another space, A (1 step), B (2 steps), and so on.

$$C \rightarrow A \rightarrow B \rightarrow \ldots$$

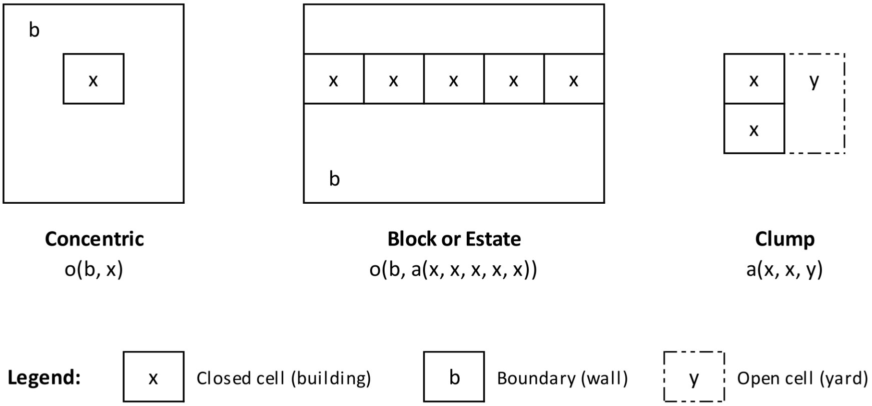
If we count the number of steps to go from some space to every other space in a complex, we can get a measure of its *total depth* (TD), or of its *mean depth* (MD), by dividing that total by the number of spaces in the complex minus one (Hillier and Hanson, 1984: 108) (Hanson, 1998: 27-28). In the previous example, the total depth of C is 3 which is the sum of steps to reach A and B (1+2) from that origin. This is also the case of B, but the total depth of A is 2, noting that this (central) space is directly connected with both B and C spaces (1+1). Thus, the mean depth is 1 (2/2) for A, and 1.5 (3/2) for B and C, noting that the number of spaces minus one in this simple complex is 2 (3-1).

This simple illustration suggests that syntactic measures can be computed for every (convex) space in a settlement or building. Then, we may find that some spaces have lower depth than all other spaces (A in the same example), and others have greater depth (B and C). The former are the most integrated spaces, where social life and/or economic activities such as retail trade might be concentrated in cities (Heitor and Pinelo Silva, 2015: 162), or the living room in most houses (Hillier and Hanson, 1984: 155-158) (Hanson, 1998: 104-105) (Hillier, 2007: 25-27). The latter are typically the most segregated, quite removed or remote spaces in a town, building or house. Thus, *integration* is inversely related with *depth*. It is a *global* measure in the sense that it considers the configuration of one space in relation with all other spaces (or with spaces at some radius of *n* steps from the original space). Besides, *local* measures such *control* are based on the relations between each space and only the spaces directly connected to it (Turner, 2004: 14).

As an intelligent critique of Modernism, space syntax stresses the deep tree-like configuration of modern estates (Hillier and Hanson, 1984: 129-132, 262-263). A *tree* is a special kind of graph which contain a *root* such that there is a unique path from the root to any other node (Flach, 1994: 83). Thus, trees are necessarily non-cyclic or *acyclic*, that is, they do not contain paths from a node to itself. The absence of 'rings' in trees give them a *non-distributed* rather than a *distributed* configuration, where there is more than one independent route from one space to another, including one passing through a third space (Hillier and Hanson, 1984: 148). *Distributedness* can be illustrated by a recursive discrete process where a primary cell or syntactic type is 'glued' to cells of the same type by the space 'between' them, meaning that the global structure of the settlement is distributed amongst all primary cells (Hillier and Hanson, 1984: 11).

The analysis of urban settlements as discrete systems is founded on the premise that urban forms, in general, can be structured or generated from some elementary types. Hillier and Hanson (1984: 55-62) showed how the 'beady ring' structure observed in some French rural villages can be generated with the aid of a computer by applying recursively and randomly a very simple syntax, called 'clump' and denoted by Xy because it is composed by a closed cell or elementary building X adjacent (or permeable) to an open cell y. Other elementary generators (Hillier and Hanson, 1984: 78) are the concentric type, where X is put inside another closed cell or boundary, here denoted by (b o X) where 'o' represents the relation of containing, and the block or estate syntax where several adjacent buildings are enclosed inside a boundary, namely, (b o XXXXX).

The clump syntax, visible in many traditional villages and towns, is associated with symmetric and distributed urban forms, while the concentric and estate types are the staple of modern, typically asymmetric and non-distributed settlements. For instance, the Portela estate, located in Sacavém – Loures, near Lisbon, can be generated on a computer using Prolog from the elementary forms described in Figure 2 (Fernandes, 2021).



Concentric
o(b, x)

Block or Estate
o(b, a(x, x, x, x, x))

Clump
a(x, x, y)

Fig. 2: Elementary generators

METHOD

A Prolog *program* is a knowledge base (or database) composed by a collection of facts and rules which describe a set of relations of interest (Blackburn et al., 2006). *Facts* express unconditional truths (Flach, 1994: 5), namely the connections between lines in an axial map (the longest and fewest lines that cover the street grid) or between spaces in a convex map (Hillier and Hanson, 1984: 91-92) (Hillier, 2007: 98).

In Areeiro's axial map (Figure 3), from left to right and from north to south, we know that Rua Oliveira Martins is directly connected with four lines: Avenida São João de Deus (the northern limit of the settlement), Avenida Sacadura Cabral (through a pedestrian passage/tunnel), Rua Edison and Avenida João XXI. These facts can be declared in a Prolog program (text file), e.g. `areeiro.pl`, in the following fashion:[1]

```
connected(rua_oliveira_martins,av_sao_joao_de_deus,1).
connected(rua_oliveira_martins,av_sacadura_cabral,1).
connected(rua_oliveira_martins,rua_edison,1).
connected(rua_oliveira_martins,av_joao_xxi,1).
```

These four clauses or *atoms* belong to the same *predicate* because they have the same *name* ('connected') and the same *arity*, that is, the same number of arguments (three) enclosed in parentheses and separated by commas (Flach, 1994: 25, 31). Here, the third argument is the topological distance between two adjacent lines, that is, one. In some applications, it might be greater than one in order to incorporate the metric distance or other cost (e.g. stairs or ramp) between two convex spaces, but here we adopt the topological distance introduced in *The Social Logic of Space* (Hillier and Hanson, 1984: 103). Clauses of the same predicate

---

[1] The complete areeiro.pl program is publicly available on the SWISH webpage: https://swish.swi-prolog.org/p/areeiro.pl

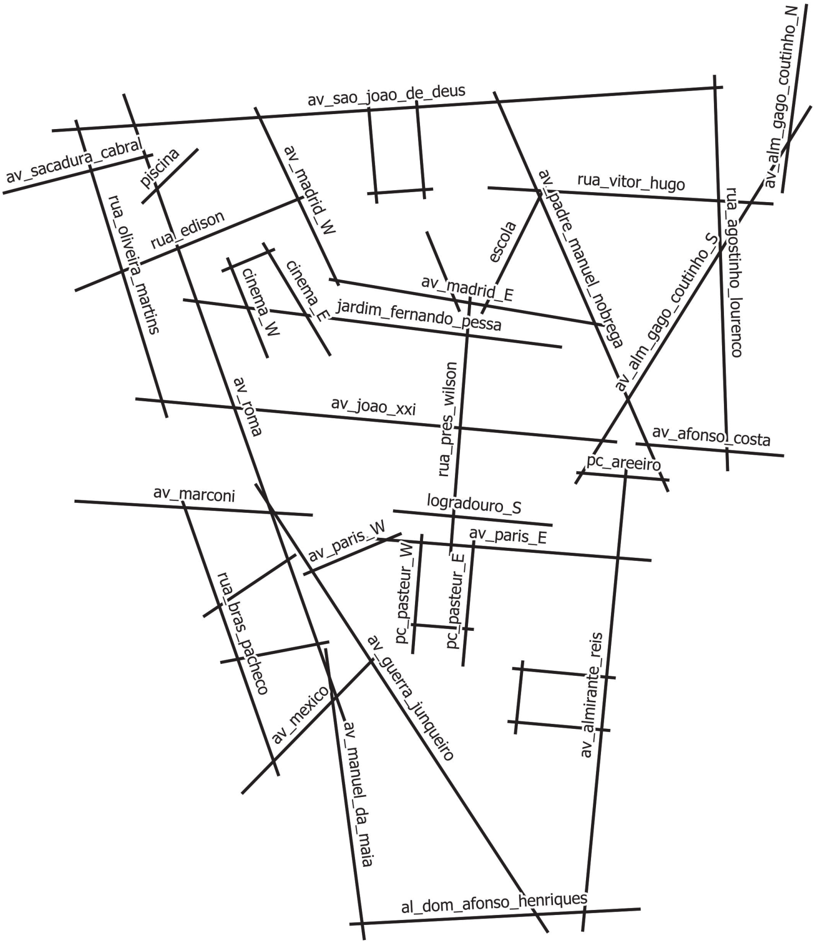


Fig. 3: Axial map of Areeiro

must be declared jointly in a Prolog program, so we must add the remaining connections pertaining to the Areeiro's axial map to our knowledge base in order to proceed, namely:[2]

```
connected(av_sao_joao_de_deus,av_roma,1).
connected(av_sao_joao_de_deus,av_madrid_W,1).
connected(av_sao_joao_de_deus,pc_afranio_peixoto_W,1).
...
connected(pc_joao_do_rio_S,pc_joao_do_rio_W,1).
```

Then, after consulting the program `areeiro.pl` with a distribution such SWI-Prolog (Wielemaker et al., 2012) readily available online on the SWISH platform

---

[2] The complete list of connections is available at the path (URL) indicated in the previous footnote.

(https://swish.swi-prolog.org/p/areeiro.pl), we may ask it about the *Y* lines connected, for instance, with Rua Oliveira Martins by posing the query:

```
?-connected(rua_oliveira_martins,Y,1).
```

where the prefix '?-' indicates that this is a query rather than a fact. In Prolog, a *variable* such *Y* is written as a sequence of letters and digits, beginning with a capital letter or underscore. An answer to the previous query, e.g. Avenida São João de Deus, will be written `{Y -> av_sao_joao_de_deus}` following the notation of Flach (1994: 5). This means that Prolog found a value for the variable *Y*, that is, a *solution* given the knowledge base previously loaded and consulted. If asked, Prolog will try to find more values to *Y* given the knowledge base, namely, `{Y -> av_sacadura_cabral}`, `{Y -> rua_edison}` and `{Y -> av_joao_xxi}`. These are the axial lines 1-step away from Rua Oliveira Martins, that is, at depth 1 from this root.

For practical purposes, we must create a pair of *rules*, that is, of conditional truths that can only be drawn when their premises are known to be true (Flach, 1994: 5), which isolates the lines that are adjacent with some axis *X* in the sense that it can be connected with *Y* or, conversely, *Y* can be connected with *X*:

```
adjacent(X,Y,1):-connected(X,Y,1).
adjacent(X,Y,1):-connected(Y,X,1).
```

where the symbol '`:-`' should be read as 'if'. These rules mean: 'for any values of *X* and *Y*, *X* and *Y* are adjacent if *X* is directly connected with *Y* or *Y* is directly connected with *X* with a topological distance of 1'. Now, we can compute the *axial connectivity* (AC) of a line, say Rua Oliveira Martins, which is the number of other lines it intersects (Hillier and Hanson, 1984: 103) using the SWI-Prolog built-in predicate `aggregate_all/3` with the template count:

```
connectivity(X,Y,AC):-
aggregate_all(count,adjacent(X,Y,1),AC).
```

For instance, the following query returns `{AC -> 4}` because Rua Oliveira Martins is directly connected with four axial lines, as said:

```
?-connectivity(rua_oliveira_martins,Y,AC).
```

Connectivity is the basic syntactic measure in the sense that the others are based on it (Heitor and Pinelo Silva, 2015: 158). In particular, the measure of *control* (E) proposed by Hillier and Hanson (1984: 109) and implemented below, sums up the space given *G* to some space *X* by its immediate (adjacent) neighbours *Z*, where *G* is the reciprocal of the axial connectivity (AC) of each *Z*, and the comma '`,`' between predicates should be read as 'and':

```
control(X,Y,E):-aggregate_all(sum(G),(adjacent(X,Z,1),
connectivity(Z,Y,AC), G is 1/AC),E).
```

This specific rule illustrates how useful and pedagogical Prolog can be in describing the syntactic concepts. In fact, it suggests the direct relation of control with the original line's connectivity, that is, with the number of its adjacent neighbours, as well as the inverse relation with the connectivity of the last ones. In fact, spaces with a strong control, that is, with an E greater than 1, are typically connected with several spaces, namely, 'cul-de-sacs' or streets with few connections. So, in order to be controlling, a line must see many spaces, but these spaces should each see relatively little (Turner, 2004: 16). In the Areeiro settlement, the case of Avenida de Roma is remarkable with `{E -> 4.593}`, which can be computed by formulating the query:

```
?-control(av_roma,Y,E).
```

Control is a *local* measure, since it only considers relations between a space (line) and its immediate neighbours (Hillier and Hanson, 1984: 109). Another local measure, *controllability*, picks out areas that may be easily visually dominated (Turner, 2004: 16). It is simply the ratio of connectivity to the total number of spaces 1- or 2-steps away from the original space. In order to compute this measure as well as relative asymmetry we must define a predicate for *depth*. Obviously, axial connectivity is related with the concept of depth 1 in the sense that it is the number of lines 1-step away from the original line. In fact, we can define a rule such that *Y* is 1-deep from *X* if *X* and *Y* are adjacent:

```
depth(X,Y,1):-adjacent(X,Y,1).
```

Similarly, the lines 2-steps away from the original line are 1-step away from the lines adjacent to the root. The following rule explores this recursive nature of depth, that is, the spaces at depth 2 from *X* are the spaces *Y* at depth 1 from the spaces *Z* adjacent of *X*:

```
depth(X,Y,2):-adjacent(X,Z,1), depth(Z,Y,1), dif(X,Y).
```

where the SWI-Prolog built-in predicate `dif/2` introduces a constraint such that every 2-deep line *Y* must be different from the original line *X* in order to avoid backward relations. The following illustrative query returns a list, indicated by square brackets '`[Y]`', with the lines 2-steps away from Rua Oliveira Martins:

```
?-distinct([Y],(depth(rua_oliveira_martins,Y,2))).
```

where the built-in predicate `distinct/2` eliminates duplicates, that is, 2-deep lines such as Avenida de Roma that can be accessed through two or more lines directly connected with the original space, in this example Rua Oliveira Martins. Depth 3 can be defined in the same recursive way by isolating the spaces 2-steps away from the spaces *Z* directly connected with the original space *X*, depth 4 by isolating the deep 3 spaces, and so on:

```
depth(X,Y,3):-adjacent(X,Z,1),depth(Z,Y,2),dif(X,Y).
depth(X,Y,4):-adjacent(X,Z,1),depth(Z,Y,3),dif(X,Y).
depth(X,Y,5):-adjacent(X,Z,1),depth(Z,Y,4),dif(X,Y).
depth(X,Y,6):-adjacent(X,Z,1),depth(Z,Y,5),dif(X,Y).
depth(X,Y,7):-adjacent(X,Z,1),depth(Z,Y,6),dif(X,Y).
depth(X,Y,8):-adjacent(X,Z,1),depth(Z,Y,7),dif(X,Y).
depth(X,Y,9):-adjacent(X,Z,1),depth(Z,Y,8),dif(X,Y).
```

It is important to note that a space or line may be accessible at several steps from the same origin. For instance, Rua Edison is either 1-step away from Rua Oliveira Martins or 3-steps away because we can go through Avenida João XXI and Avenida de Roma to get there. This occurs in every axial map with 'rings' or 'islands', such Areeiro. So, we must create a new rule to set the minimum number of steps D to go from X to Y once again using the Prolog's predicate `distinct/2`:

```
graph(X,Y,D):-distinct([Y],depth(X,Y,D)).
```

We named this predicate '`graph`' because it analytically generates the *justified graph*, or j-graph, for a root X. The resulting j-graph is a 'picture' with the depth of all spaces in a complex from a point in it (Hillier, 2007: 22-23, 72-73). For instance, we can analytically describe the j-graph of Rua Oliveira Martins with the following simple query:

```
?-graph(rua_oliveira_martins,Y,D).
```

to get:

```
{Y -> av_sao_joao_de_deus, D -> 1}
{Y -> av_sacadura_cabral, D -> 1}
{Y -> rua_edison, D -> 1}
{Y -> av_joao_xxi, D -> 1}
{Y -> av_roma, D -> 2}
{Y -> av_madrid_W, D -> 2}
...
{Y -> pc_joao_do_rio_W, D -> 6}
```

The predicates `graph/3` and `connectivity/3` are the 'building blocks' to compute several measures, namely, the above-mentioned *controllability* (F):

```
controllability(X,Y,Z,F):-connectivity(X,Y,AC), aggregate
_all(count, graph(X,Z,2), D2), F is AC/(AC+D2).
```

Another measure that can be computed with `graph/3` is *mean depth* (MD) referred to in the previous section. It is calculated by summing up the depth values D from a root X and dividing by the number of spaces (lines) in the complex minus one, as explained above (Hillier and Hanson, 1984: 108). The following implementation introduces two preliminary or auxiliary predicates that compute the *total depth*

(TD) and the number of *nodes* (N) in the complex except the root:

```
totdepth(X,Y,TD):-aggregate_all(sum(D),graph(X,Y,D),TD).
nodes(X,Y,D,N):-aggregate_all(count,graph(X,Y,D),N).
meandepth(X,Y,D,MD):-totdepth(X,Y,TD), nodes(X,Y,D,N), MD
is TD/N.
```

Mean depth is an important syntactic measure and its reciprocal can be used as a simple measure of the *integration* of each space in a complex, as suggested by Heitor and Pinelo Silva (2015: 162):

```
integration(X,Y,D,I):-meandepth(X,Y,D,MD), I is 1/MD.
```

Nevertheless, the value of total or even mean depth can be affected by the number of nodes in a graph. Thus, Hillier and Hanson (1984: 108) proposed a normalisation of MD which eliminates the bias due to the number of nodes, that is:

```
asymmetry(X,Y,D,RA):-totdepth(X,Y,TD), nodes(X,Y,D,N), RA
is 2*(TD/N-1)/(N-1).
```

This will give a value between 0 and 1, with low values indicating a space which tends to integrate the whole complex or system, and high values a space which tends to be segregated from the system. Thus, *relative asymmetry* (RA), also denoted by 'i-value' (Hillier, 2007: 77), is a normalised measure of integration.[3]

The RA measure can be used to compare different systems with (approximately) the same number of spaces. However, if the systems differ considerably in size, a second normalisation must be applied because a small system always looks more integrated than a large one (Turner, 2004: 14). To deal with this empirical problem, Hillier and Hanson (1984: 109-113) proposed a second standardised measure, the *real relative asymmetry* (RRA), which is the RA value of the space divided by the RA value of the root of a graph shaped like a 'diamond' where there are *k* nodes at middle level, *k*/2 at one level above and below the middle level, *k*/4 at one level above and below the *k*/2 level, and so on, until there is one node at the root and deepest nodes (Teklenburg et al., 1993: 350-351). The following implementation uses the formula proposed by Krüger and Vieira (2012: 200) to estimate the RA value of the root of that diamond (*d-value*):

```
dvalue(K,DV):-DV is 2*(K*(log((K+2)/3)/log(2)-1)+1)/
((K-1)*(K-2)).
rra(X,Y,D,RRA):-asymmetry(X,Y,D,RA), nodes(X,Y,D,N),
dvalue(N+1,DV), RRA is RA/DV.
integrationHH(X,Y,D,IHH):-rra(X,Y,D,RRA), IHH is 1/RRA.
```

Both RA and RRA are measures of *asymmetry*. Besides, *distributedness* can be

---

[3] The mentioned program areeiro.pl, available on the SWISH webpage, has additional predicates to compute radius-r integration and other local measures.

evaluated using either control (or controllability) or *relative ringiness* (RR). The RR of some space is the number of independent rings that pass through that space over the maximum that can pass through it, which will be the total number of nodes in the system minus one (Hillier and Hanson, 1984: 153-154). Detecting the number of rings that pass through one particular space is a very difficult task to be performed by a computer: for instance, the software Depthmap (Turner, 2004) does not compute RR. As a programming language concerned with Artificial Intelligence, Prolog can do the task relatively well in most cases. In fact, the following code will produce accurate estimates for RR, namely where the number of independent rings is smaller than axial connectivity:

```
graph2(X,Y,D):-distinct([Y],(depth(X,Y,D),D>1)).
ring(X,Y,Z,D):-adjacent(X,Y,1), graph2(Y,X,D), connectivity
(X,Z,AC), AC > 1.
rings(X,Y,Z,D,R):-
aggregate_all(count,(ring(X,Y,Z,D)),C), R is max(C-1,0).
ringiness(X,Y,Z,RR):-rings(X,Y,Z,D,R),nodes(X,Y,D,N), RR
is R/N.
```

RESULTS

The key syntactic measures computed with Prolog for the *integration core*, that is, for the 25% most integrating spaces of Areeiro are presented in Table 1[4]. Typically, greater connectivity is associated with the most integrated (or less segregated) spaces, namely, Avenida de Roma (directly connected with 12 lines), Avenida São João de Deus or Avenida Padre Manuel da Nóbrega (both connected with 7 lines), but

Axial lines	**AC**	**E**	**F**	**TD**	**MD**	**I**	**IHH**	**RA**	**RRA**	**RR**
av_roma	12	4.593	0.414	101	2.349	0.426	2.247	0.064	0.445	0.233
av_joao_xxi	4	0.700	0.174	109	2.535	0.394	1.974	0.073	0.507	0.070
av_sao_joao_de_deus	7	2.060	0.280	111	2.581	0.387	1.916	0.075	0.522	0.140
rua_pres_wilson	5	1.900	0.278	112	2.605	0.384	1.888	0.076	0.530	0.070
av_padre_manuel_nobrega	7	1.926	0.304	115	2.674	0.374	1.810	0.080	0.553	0.140
jardim_fernando_pessa	4	1.283	0.211	116	2.698	0.371	1.785	0.081	0.560	0.070
av_guerra_junqueiro	5	1.500	0.250	119	2.767	0.361	1.715	0.084	0.583	0.093
av_alm_gago_coutinho_S	6	2.226	0.333	121	2.814	0.355	1.671	0.086	0.599	0.093
av_manuel_da_maia	3	0.667	0.177	126	2.930	0.341	1.570	0.092	0.637	0.047
av_madrid_E	5	2.010	0.294	127	2.953	0.339	1.551	0.093	0.645	0.070
av_almirante_reis	5	1.867	0.357	127	2.953	0.339	1.551	0.093	0.645	0.093

AC - Axial Connectivity; E - Control; F - Controllability; TD - Total Depth; MD - Mean Depth; I - Integration Heitor & Pinelo Silva (1/TD);
IHH - Integration Hillier & Hanson (1/RRA); RA - Relative Asymmetry; RRA - Real Relative Asymmetry (RA/d-value); RR - Relative Ringiness

Table 1: Syntactic measures for Areeiro's integration core computed with Prolog

---

[4] The same values are obtained using DepthmapX 0.8.0, except for controllability, with Prolog producing slightly different estimates. Relative ringiness (RR) is not calculated by Depthmap.

exceptions can occur, such Rua Presidente Wilson/Rua Cervantes (connected with only 5 lines), Jardim Fernando Pessa (4 lines) or Avenida Manuel da Maia (3 lines).

Where connectivity and integration happen together, complexes are said to be *intelligible* (Hillier, 2007: 94-95). This kind of relation is not readily evident in Areeiro, where the integration core includes secondary and/or less connected lines, as stated above. This is a direct consequence of the openness of city blocks derived from installing public gardens such Jardim Fernando Pessa, schools, pools and other facilities in the backyard of the main streets, which gave them an unexpected centrality. In fact, Areeiro was the first borough in Lisbon with neighbourhood units and open (U-shaped) blocks of flats (Salgado and Lourenço, 2006: 62). This modern attitude originated asymmetries, namely, segregated 'main' streets and a strange paradox: Praça do Areeiro, the 'regulator square' of the settlement projected by architect Luís Cristino da Silva [1896-1976] between 1938 and 1949 (Becker et al., 1997: 188-189), is a moderately integrated space with an i-value (RA) of 0.097 and a reciprocal of mean depth (I) of 0.328. Thus, it is not included even in the core of the 25% most integrated spaces.

Nevertheless, Areeiro is an intelligible settlement in the sense that the linear correlation between connectivity and integration is high (0.794). This close relation between what can be seen from each space and what cannot be seen (Hillier, 2007: 94) is suggested by the agglomeration of points around the line in Figure 4. The outlier located at the north-east end of the figure corresponds to Avenida de Roma. Its prominence is extended also to relative ringiness (RR) with a high value (0.233, see Table 1). This result suggests that one in every four possible rings in Areeiro go through that avenue. Thus, Avenida de Roma is not only the most connected and integrated space but also the distributive axis of Areeiro (and Alvalade). This latter role of Avenida de Roma is also suggested by the high values of control (E, 4.593) and controllability (F, 0.414).

Besides, the Areeiro settlement validates well-known facts about space syntax, namely the effect of grid deformation on integration found by Hillier and Hanson (1984: 124-125) for the Barnsbury area in London. In fact, the 'deformed' axis of Avenida Guerra Junqueiro / Praça de Londres is one of the most integrated spaces in the Areeiro borough (see Table 1), concentrating retail trade, coffee shops and other economic and social activities (e.g. church). This centrality resulted also from a good relationship with the larger-scale system where this axis is embedded, namely, with the pre-existing southern boundary of the settlement (Alameda Dom Afonso Henriques) and with Alvalade, the northern settlement developed after Areeiro, which is also structured by the most integrated and distributed space of the Areeiro neighbourhood, that is, Avenida de Roma.

A good integration resulting from grid deformation is also found in Avenida Padre Manuel da Nóbrega, another shopping street near Areeiro square. In fact, retail trade

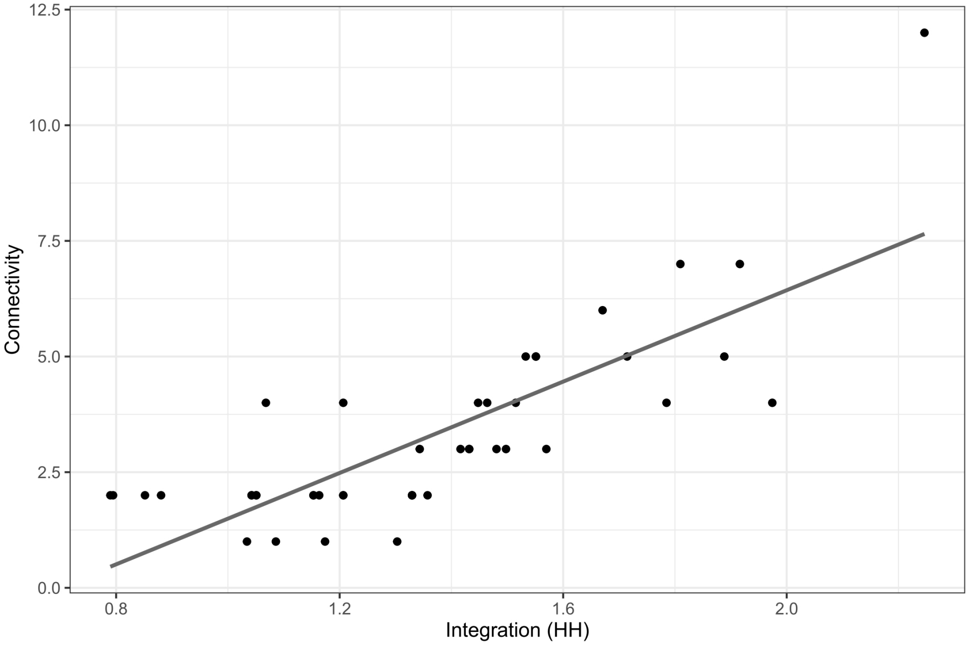


Fig. 4: Relation between connectivity and integration in Areeiro (intelligibility)

is typically located at the most integrated axes as suggested by Figure 5, with a reasonable correlation coefficient of 0.644. However, some exceptions do exist in Areeiro, for example Avenida São João de Deus and Avenida Manuel da Maia.

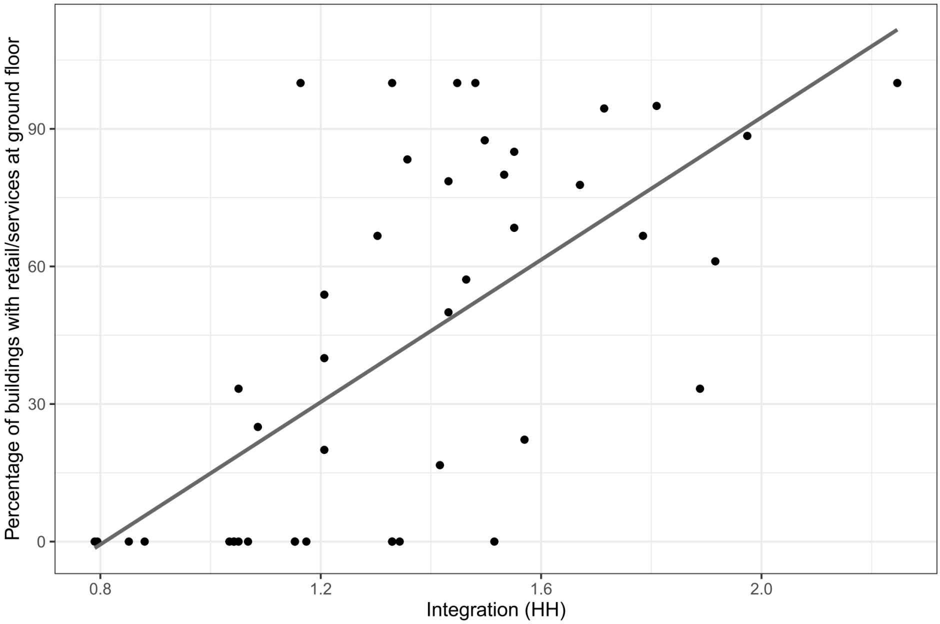


Fig. 5: Relation between incidence of retail/services at ground floor level (% of buildings) and integration in Areeiro

In fact, these integrators have moderated incidence of retail/services on the ground floor of their buildings (61.1% and 33.3%, respectively) because they were built on one side only: the other (northern) side of Avenida São João de Deus is bordered by a railway, and Avenida Manuel da Maia is bordered at full extension from the west by the backyard of a public building (that of Statistics Portugal), which limits their attractiveness as shopping streets.

CONCLUSION

The latter result suggests that space syntax does not show the whole story, because syntactic measures do not account for the facade's orientation to the sun, the sides' built-up or even the current uses of axes or convex spaces. In fact, the location (or concentration) of social and economic activities may privilege sunny facades, 'corridor' streets built on both sides, flat layouts instead of climbing streets, the presence of green spaces and other amenities, as well as urban areas with previously installed activities in order to explore economies of agglomeration and movement. This kind of factors can be critical to evaluate the quality and liveability of open spaces in addition to distributedness, integration, intelligibility and other proprieties captured by space syntax methods.

Nevertheless, the contribution of space syntax towards understanding how configuration can structure and influence social relations is undeniable. Empirical studies have suggested that integration (or relative asymmetry) is the key feature to predict human co-presence and movement in cities (Hanson, 1998: 1). Space syntax methods can also detect layout problems, namely, 'dead areas', or how (public sector) housing estates connect to their surroundings (Hillier et al., 1989: 5). Finally, space syntax theories based on discrete systems show how complex structures can be generated from simple patterns given basic rules (Hillier and Hanson, 1984: 59-61).

In fact, space syntax was founded on the premise that even complex spatial arrangements, just like natural languages, are systems governed by simple concepts and rules. Prolog was founded on the same idea. In fact, Prolog was born as a research project to process natural languages using grammars. Thus, space syntax and Prolog are similar in reasoning, and both represent knowledge in a declarative way. Indeed, the 'problem of knowability' is paramount not only to space syntax but also to Artificial Intelligence (and Prolog), as stressed by Hillier and Hanson (1984: 46) in their seminal work.

The present research suggests that Prolog can be used to good effect to describe an urban settlement such as Areeiro as a 'field of knowables' or an 'intelligent system'. It is an appropriate tool for syntactic analyses of small or medium-sized complexes following the 'case study' approach introduced by Hillier and Hanson (1984) and Hanson (1998). Definitely, Prolog is not the tool to perform deep analyses of whole cities or metropolitan areas where Depthmap (or another similar package) should be privileged.

Thus, future developments of this research will cover restricted modern and/or public housing estates as well as buildings with traditional architecture in line with the anthropological tradition of space syntax. Anyway, the inclusion of economic reasoning in space syntax theories and methods, complementing the contributions of other sciences, is another line of development that may use Logic or Dynamic Programming.

REFERENCES

Becker, A., Tostões, A., & Wang, W. (1997). Catálogo da Exposição Arquitectura do Século XX: Portugal. Frankfurt am Main / Lisboa: Deutsches Architektur-Museum / Centro Cultural de Belém.

Blackburn, P., Bos, J., & Striegnitz, K. (2006). Learn Prolog Now! College Publications. Retrieved from: http://www.learnprolognow.org on 27/09/2020.

Colmerauer, A., & Roussel, P. (1993). The birth of Prolog. ACM SIGPLAN Notices, 3. DOI: 10.1145/155360.155362.

Fernandes, P. A. (2021). A syntactic analysis of the Portela Urbanization using Prolog [Conference poster]. Optimistic Suburbia 2: Middle-Class Large Housing Complexes, ISCTE-IUL, Lisboa. DOI: 10.13140/RG.2.2.33746.68801.

Flach, P. (1994). Simply Logical: Intelligent Reasoning by Example. John Wiley and Sons. Retrieved from: https://book.simply-logical.space/ on 27/09/2020.

Hanson, J. (1998). Decoding Homes and Houses. Cambridge: Cambridge University Press.

Heitor, T., & Pinelo Silva, J. (2015). A Sintaxe Espacial e o Ambiente Construído - Análise Morfológica. In V. Oliveira, T. Marat-Mendes, & P. Pinho (Eds.), O Estudo da Forma Urbana em Portugal (pp. 147–189), Porto: Universidade do Porto.

Hillier, B. (2007). Space is the machine: a configurational theory of architecture. London: Space Syntax.

Hillier, B., & Hanson, J. (1984). The Social Logic of Space. Cambridge: Cambridge University Press.

Hillier, B., Jones, L., Penn, A., Jianming, X., & Grajewski, T. (1989). The Architecture of the Maiden Lane Estate: a second opinion. London: Unit for Architectural Studies, Bartlett School of Architecture and Planning, University College of London. Retrieved from: https://discovery.ucl.ac.uk/id/eprint/1751/ on 23/03/2021.

Kowalski, R. A. (1988). The early years of Logic Programming. Communications of the Association for Computing Machinery, 31(1): 38–43. DOI: 10.1145/35043.35046.

Krüger, M., & Vieira, A. P. (2012). Scaling relative asymmetry in space syntax analysis. Journal of Space Syntax, 3(2): 194–203.

Lamas, J. M. R. G. (1992). Morfologia Urbana e Desenho da Cidade. Lisboa: Fundação Calouste Gulbenkian / Junta Nacional de Investigação Científica e Tecnológica.

Salgado, M., & Lourenço, N. (2006). Atlas Urbanístico de Lisboa. Lisboa: Argumentum.

Teklenburg, J. A. F., Timmermans, H. J. P., & van Wagenberg, A. F. (1993). Space Syntax: Standardised Integration Measures and Some Simulations. Environment and Planning B: Urban Analytics and City Science, 20(3): 347–357. DOI: 10.1068/b200347.

Turner, A. (2004). Depthmap 4 - A Researcher's Handbook. London: Bartlett School of Graduate Studies, University College of London. Retrieved from: https://discovery.ucl.ac.uk/id/eprint/2651/ on 01/10/2020.

Wielemaker, J., Schrijvers, T., Triska, M., & Lager, T. (2012). SWI-Prolog. Theory and Practice of Logic Programming, 12(1–2): 67–96. DOI: 10.1017/S1471068411000494.