DOUNAS, T. 2013. Some notes on the incompleteness theorem and shape grammars. In Zhang, J. and Sun, C. (eds.) *Global design and local materialization: proceedings of the 15th International conference on Computer-aided architectural design features 2013 (CAAD Futures 2013), 3-5 July 2013, Shanghai, China*. Communications in computer and information science, 369. Cham: Springer [online], pages 368-375. Available from: https://doi.org/10.1007/978-3-642-38974-0_35

Some notes on the incompleteness theorem and shape grammars.

DOUNAS, T.

2013

Some notes on the incompleteness Theorem and Shape grammars

Dounas Theodoros

Theodoros.dounas@xjtlu.edu.cn
Xi'an Jiaotong Liverpool University

Abstract. This paper is False
The paper presents a critique of the Shape Grammar paradigm viewed through the lens of the incompleteness theorem of Gödel. Shape Grammars have been extensively researched through many lenses. Their productive systemic nature was the focus of the first papers along with more recent treatises in the field while their use in analysis of known building styles has been extensive and a proven mechanism for style analysis. It is surprising though that use of Shape Grammars in actual design in practice however has been minimal. The architectural community has not actively used the paradigm in the design of real buildings, probably because of the rigid analytical approach to style and rules, following from the academic analysis the paradigm has been subjected to. However I propose that there is another underlying reason, other than the rigid approach to construct a Shape Grammar. The nature of the concurrent application and creation of the rules lies close to the incompleteness theorem of Gödel, that uses a multitude of Turing Machines to prove that a from a set of True Axioms -A- we will never be able to determine if all sentences are true, without having to invent new axioms, outside the initial set -A-, thus unproven in terms of their true or false nature. Negation of this possibility drives us to the conclusion that true Design can never be feature -complete and thus can never be placed in a trusted framework that we all agree or believe it to be the complete truth.

Keywords: Incompleteness Theorem, Incomputability of Shape Grammars

1 Introduction

The paper is concerned with the application of shape grammars in the practice of architectural design. Specifically it tries to address the issue of predictability of shape studies and the application of rules, the creation of rules and timing there of, in contrast with architectural design in its applied forms. It first develops a limited typology of shape grammars based on precedent work [19] with the addition of two examples, then a critique of the types of shape grammars, developing then the discussion of analytical versus design grammars, feature parity between Turing machines and Shape Grammars and concludes with a discussion on the incompleteness theorem by Gödel and its possible application in the Shape Grammar paradigm.

2 Typology of Shape Grammars

Shape grammars are formal systems, consisting of an initial set of shapes, an initial set of rules and their termination shapes that end the application of the grammar. The classic definition of Shape Grammars is as follows: [17]

A shape grammar is a 4-tuple (Vt, Vm, R, I) where :

Vt is the initial set of shapes from where we are going to pick our starting shapes.

Vt* is a set that contains any number of elements formed from elements belonging to Vt, combined, rearranged and transformed in any number of ways

Vm is a finite set of shapes so that the common set between Vt and Vm is null (Vm + Vt*) =0 meaning that Vm does not contain any element belonging to Vt*.

R is a set consisting of pairs of shapes, which when taken together form rules of the U>V type. I is a set of shapes that consist of termination shapes, ie the shapes that when found by the user of the grammar terminate any further application of the rules. The use of the computer in the application of shape grammars encourages the generation of design alternatives, than in a "traditional" design process would happen "by hand". Apart from the philosophical question on whether a user simulates a computer when generating design alternatives by hand the application of the rules and the choice of the initial shapes give infinite combinations within the possibilities of the grammar, even accounting for emergent shapes that did not belong in the initial Vt set, however it is impossible to go beyond the initial set of possibilities the grammar allows. This impossibility stems for the fact that the enumeration of rules and the enumeration of shapes initially in the shape grammar already restrict the number of different possible designs or different alternatives, even if only from a purely numerical point of view.

However looking carefully at the mechanisms of creating a shape grammar one can make a different case. The literature suggests that a user has in her hands a shape grammar, complete with termination rules, before she starts to design. However the design process does not operate in such a manner. In real life the designer does not possess a well thought-out map before the design begins, but the map, ie the grammar, is created at the same time the design process is getting negotiated. Contrasting well constructed shape grammar mechanisms [5,7,8,12,21] with the output and interim stages of a design process initially results in the view that design in real life is a unorganized and naive process rather than the well constructed, actively reasoned, and feature complete process of shape grammars. By feature complete we ascertain the completeness of the design process with a clear starting point and a clear end, while at the same time the set of rules is identified and enumerated in full. This rigorous structure and the completeness of shape grammars originates from the 'Turing machine' algorithmic paradigm with which they have feature parity [10]. The 'Turing Machine' paradigm also presents processes with a clear start, clear specific rules for elements manipulation and clear termination rules and elements. This feature makes them an excellent metaphor for computing, but a misunderstood metaphor for design. From a computational point of view the classic design process looks like an infinite number of Turing Machines working together, but with a probabilistic, random factor built in when making choices to transcribe - Change- one element to another, with the rules being created at the time they are applied.

A complete typology of Shape Grammars cannot be presented in the confines of this paper, however a critique is presented bellow based on a precedent analysis conducted by [19] with the addition two specific cases, the grammatical basis of Chinese traditional architecture by [4] and the implementation of curved Shape Grammars by [11]. In our case the analysis of precedent case studies is extended here specifically through the classification of grammars in terms of dealing with emergence, since this is one of the core features of shape grammars that provide design robustness in the production of alternative designs. In the analysis of the grammatical basis of Chinese traditional architecture [4] document the building steps of a traditional Chinese house in Taiwan, based on Yingzhao Fashi - the manual of traditional Chinese architecture, and derive a grammar from the documentation of the building steps. Their method of building the grammar brings forward the main characteristics of the traditional Chinese architecture: Axial symmetry, in one axis or in parallel axes, Additive composition and a Top-Down Approach in classic architectural composition terms. Here the analysis is predictable to the last piece of timber used in the construction. Emergence does not happen, as the manual -or the grammar- is followed to the last iota. However one can make the case that emergence does happen, but the rigid system employed does not allow the designer to recognize emergent shapes and employ them since the compositional and construction path is laid before her from the beginning. The second case, the implementation of curved grammars by [11] is “stemming from Krishnamourti’s maximal line solution and using a solution based on the theorem of space curves: any two continuous functions of a real variable define a space curve. These functions serve as its curvature and torsion, with the variable acting as a natural distance parameter.” This analysis uses explicitly the embedding of shapes in Shape Grammars that is considered the generative power of the Shape Grammar Paradigm. A single part of a Curve can belong in more than one whole curves at the same time, allowing the designer the freedom to choose which curve to use when moving forward [Figure 1]

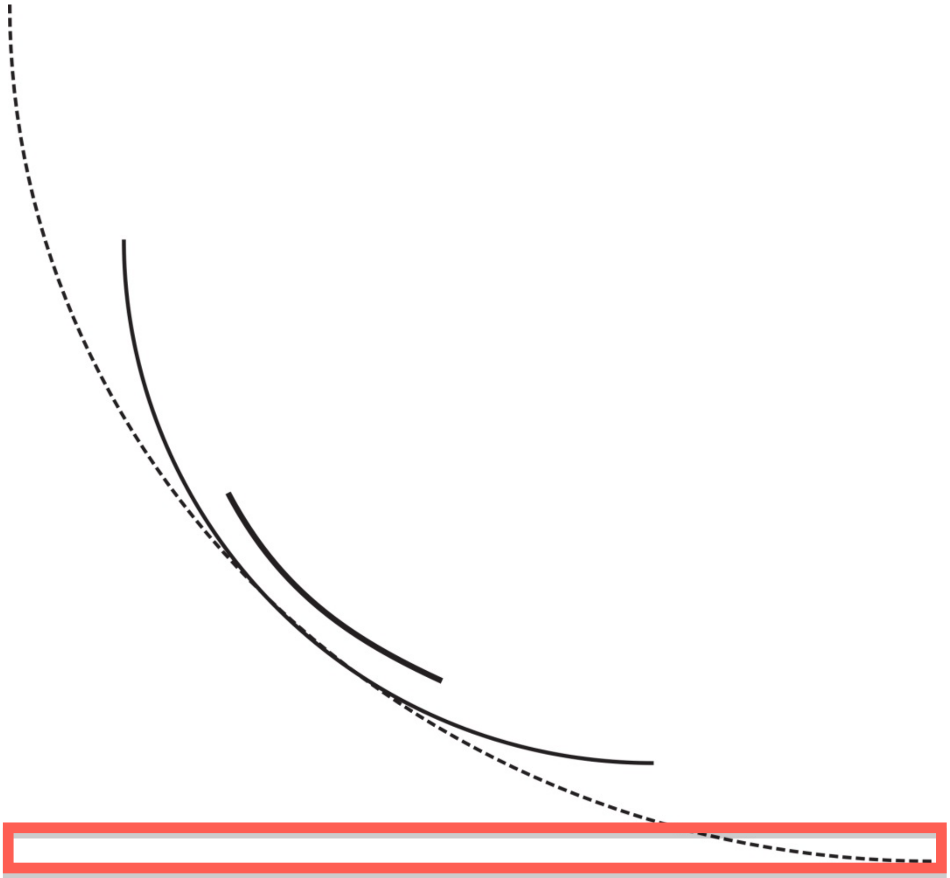

Fig. 1. : Embedding of the same part of a curve in two different curves.

This freedom is embedded in the shapes and is not a characteristic explicitly put forward by the designer, at the initial stages of building the grammar. In a sense it is a freedom that cannot be avoided, intrinsic of the Shape Grammar paradigm when applied to curves. However emergence can be dealt with by the designer directly by creating rule(s) of recognition of the curve. At the same time this shape grammar example intimately shows the creative power of shape grammars and their encapsulation for novelty, if used in a creative manner, full of exploration of different alternatives; in this case the application of different embedding rules.

3 Analysis vs Design

Shape grammars in analysis perform predictably. In analyzing a style the grammarist does not consider variations where the outcome of a termination rule lies outside the body of work that represents the style under analysis. The distinction between classical and non-classical computation in architecture [16] provides a clear example of the issue addressed in the present paper: In non-classical computation the computing algorithm is decoupled or even divorced from the parameters that formulated the problem. In other words, the explanation of the results, the issues that the observer sees are divorced from the computation mechanics, even if the representation is highly classical, ie architectural. In the same book [16] a classification of shape grammar constructs is provided: Analytical and Design. The analytical shape grammars are used to formulate a hypothesis of how a style is constructed. The analysis breaks down the style into two sets of parts and rules, to be used in reconstructing instances of the architectural style. Analytic shape grammars aim in describing rather than pre-scribing design. In their application the sets of geometric shapes and rules are given before design of the stylistic instances begin- however it would be interesting to observe the process of a shape grammarist in the process of deconstructing a style to the set of rules and initial proto-shapes that constitute any given analytical grammar. To understand this one can look at the differences in initial shapes and initial rules that the literature provides: Analysis of the same stylistic example, Palladian Grammars [18], has been conducted by two teams of researchers [2] with a span of 34 years between the two instances of analysis. From the two Palladian Grammars of Stiny and Mitchel and Benros et al, one cannot identify the 'correct' in terms of Palladian style grammar, even though the example by Benros et al is much more economical in number of rules.

Few examples of 'design' shape grammars exist: the Froebel block grammar from [16], The Malageira housing grammar by [6] or the animation parity grammars designed by the author [5]. In these examples the creation of the grammar is simple and straight forward. No explanation is needed as long as the grammar performs as the designer intended, pedagogically in the case of Stiny, creating new designs that belong into an established architectural language in the case of Duarte's discursive housing grammar or creating new buildings that respond to certain conditions in the case of the author. However one can propose that in these cases the design of the grammar happens simultaneously with the application of the grammar. Some prediction of unexpected results can be incorporated inside the rules of the grammar, when emergence is expected. Expectation of emergent shapes is though a contradiction and in most cases the grammarist will have to create rules as she goes along, ie creation and completion of the full set of rules for a style is dependent on the grammarist assuming the role of designer and grammarist at the same time, concurrently. Like a classic design process, initial rules are applied, emergence is noticed and at that mo-ment the designer of the grammar can decide to ignore (thereby nullifying the emergent shape) or exploit the emergent shape by creating, modifying or adapting rules so that the emergent shape is used in the initial set. The critical point in Emergence is its definition [13]. Surprise is the characteristic most designers subscribe to emergent behaviors or shapes in generative or production systems and one could attribute surprise to the limits of the knowledge of the designer rather than emergence itself [13].

4 Feature Parity

Feature parity between Turing Machines and Shape Grammars is proven by implementing a ‘Turing Machine’ inside a Shape Grammar, sacrificing in the process the graphic visual nature of the Shape Grammar in implementing a universal symbolic computing system. When we implement this, shapes are treated as symbols representing an algebraic unit, losing their underlying possibility for emergence and embedded shapes. Embedding provides some of the creative freedom to the designer using the grammar, in the sense that in the choice of rules to apply, the designer can choose to apply different rules in specific stage of the computation, negating the creation of a predetermined result

In this instance Shape grammars do have one to one mapping with Turing machines, but not feature parity. Parity would require that a shape grammar could be implemented as a ‘Turing machine’ at all times, however the creative process of the designer [15,16] is much more poised towards a cumulative, back-and-forth process of iteratively building on an initial idea that gets re-interpreted at every step. For feature parity to exist in the universe of ‘Turing machines’ the actions of the designer should be recordable, transferable to rules in a Turing machine. After the act of designing one can always go forward by backtracking on the design steps, rationalize or generalize from those and create a ‘Turing machine’ specification to fit the design outcome. Under these conditions feature parity would make possible the application from Mathematics and Logic Thinking of the Incompleteness theorem of Gödel.

5 Incompleteness Theorem in Shape Grammars

Turing in his original paper on “Turing Machines” described three states for the hypothetical “Turing Machine”; based on its internal state and the symbol currently being read, the machine could do one of three things in the next step: (i)write a new symbol, (ii)move backwards or forwards one square, or (iii)switch to a new state or halt. On the halting step there exists the logical paradox that Alan Turing proved to be “in-computable”: Can we decide whether any given software programme or “Turing machine” will ever halt? Turing solved this by supposing that there exists a programme P that can decide whether any given programme Q can halt. If we now modify programme P to produce a new programme P’ that takes any given programme Q as in-put and:

1. runs forever if Q halts given its own code for input
2. Halts if Q runs forever given its own code as input

Then all we need to do now is provide P’ with its own code as input. Therefore P’ will run forever if it halts or it will halt if it runs forever, which of course is a logical paradox, or rather an incomputable situation.

Moving from this, imagine another situation, where we start with the V(i) set containing all possible architectural designs. Algorithmically it is possible that we can describe a "Turing Machine" T(a) that will be able to produce all Designs in a given architectural style, or a given subset of architectural elements that have common features. Is there a "Universal Turing Machine" T(i) capable of simulating any T(a)? In effect is there a T(i) capable of simulating any Shape Grammar known or unknown, allowing us to provide a true formalized framework for explaining the production of all Designs? To this question Gödel's Incompleteness theorem can helps us find the answer.

In simple terms Gödel's Incompleteness theorem states that there exist systems of sentences (axioms) where even if we may know that an axiom, or rules of transformation to produce a certain design, in a system is true, we have no manner in proving that the axiom is true. Gödel's hypothesis was concerned with natural numbers axiomatic systems and number theory, negating Hilbert's program to find a set of axioms that would definitely prove that mathematics are a complete science. Connected with algorithms and their basic mechanism, ie. Turing Machines, Gödel in essence predicted that given specific problems to solve algorithmically, there is never a consistent manner to decide that a problem can be predictably solved, ie computed before we attempt the solution. Translating this in design and shape grammar terms one can say that Gödel's incompleteness theorem could mean that we would never know whether a shape grammar can predictably produce design results within a known specific design language. This may seem obvious to a practicing designer but it is not as a simple situation for the shape grammarist. Shape prediction in shape grammars has been restricted either in the technique of using multiple classes of rules when needed or in using embedding to bypass the issue of prediction altogether. By logically following through the repercussions of the incompleteness theorem of Gödel in Turing machines, to shape grammars, if and when feature parity exists, the shape grammarist acting as a logician can never truly predict the results of her shape rules. This may very well be the reason why some of the shape grammar examples use external to their inherent system references grammars or productive systems that complement their agility in design.

6 Conclusions

In critiquing the shape grammar paradigm the present paper attempts to clarify their inherent logical and algorithmic constraints. However the shape grammarist does not always retain the knowledge of these constraints in her mind when designing or applying rules to shapes. This situation of oblivion can very well be connected with the act of design where consistency, predictability or completeness are not always desired features in processes. We are aware that the issue of application of the Incompleteness theorem in shape grammars could provide more issues for clarification or strengthening of the paradigm in the future.

References

1. Ahmad, S et al (2004): Design Generation of The Central Asian Caravanserai, 1st ASCAAD International Conference, e-Design in Architecture, KFUPM, Dhahran, Saudi Arabia. pp 43-58
2. Benros, D., Duarte, J., Hanna, S: An alternative Palladian shape grammar: A subdivision grammar for Palladian villas, Proceedings of the 17th International Conference on Computer Aided Architectural Design Research in Asia / Chennai 25-28 April 2012, pp. 415–424 (2012)
3. Chase S C, (2002): A model for user interaction in grammar-based design systems" Automation in Construction 11 pp161-172
4. Chiou, S., Krishnamourti, R: The grammatical basis of Chinese traditional architecture, Languages of design 3, Formalisms for word, Image and sound, pp 5-31 (1995)
5. Colakoglou, B: An Informal Shape Grammars for Interpolations of Traditional Bosnian Hayat Houses in a Contemporary Context, Generative Art 2002, pp15.1-15. 15.9 (2002)
6. Dounas, T.,Kotsiopoulos, A: Dynamic (Shape) Grammars, CAADRIA 2008 [Proceedings of the 13th International Conference on Computer Aided Architectural Design Research in Asia] Chiang Mai (Thailand) 9-12 April 2008, pp. 22-28, (2008)
7. Duarte, J: A Discursive Grammar for Customizing Mass Housing, The case of Sizá's houses at Malagueira, Digital Design [21th eCAADe Conference Proceedings / ISBN 0-9541183-1-6] Graz (Austria) 17-20 September 2003, pp. 665-674 (2003)
8. Downing, F., Flemming, U: The bungalows of Buffalo, Environment and Planning B: Planning and Design, volume 8, pp. 269-293 (1981)
9. Flemming, U: More than the sum of parts: The grammar of queen Anne houses, Environment and Planning B: Planning and Design, 1987, volume 14, pp323-350 (1986)
10. Gips, J: Shape Grammars and their uses, Artificial perception, Shape perception and computer aesthetics. Birkhauser Verlag, Basel und Stuttgart, (1975)
11. Jowers, I., Earl, C: Implementation of curved shape grammars, Environment and Planning B: Planning and Design, vol 38, pp 616-635 (2011)
12. Koning, H. et al: The language of the prairie: Frank Lloyd Wright's prairie houses, Environment and Planning B: Planning and Design, volume 8, pp. 295-323 (1981)
13. Monedero, J: Can a Machine Design? A disturbing recreation of Turing's Test for the use of architects, in "Architectural Computing from Turing to 2000" [eCAADe Conference Proceedings / ISBN 0-9523687-5-7] Liverpool (UK) 15-17 September 1999, pp. 589-594 (1999)
14. Rao, Z et al: "Emergence in Entity Grammar systems" Kybernetes, Vol.38 Iss:10 pp1856-1861 (2009)
15. Schön, D: *The Reflective Practitioner* (Basic Books, New York) (1983)
16. Stiny, G: Shape, Talking about seeing and doing, The MIT press, Campridge, Massachusetts, London, England, (2006)
17. Stiny, G.,Gips,J.: Shape Grammars and the Generative specification of painting and sculpture. IFIP Congress 71, Area 7 (Sciences and Humanities: Models and Applications for the Arts) (1971)
18. Stiny, G., Mitchell, W. J.: The Palladian grammar , Environment and Planning B: Planning and Design, 5(1), 5-18 (1978)
19. Sumbdul, A., Chase, SC: Style Representation in Shape Grammars, *Environment and Planning B: Planning and Design* 2012, volume 39, pages 486 – 500 (2011)
20. Tapia, M: A visual implementation of a shape grammar system, Environment and Planning B: Planning and Design, volume 26, pp 59-73 (1999)
21. Yue, K et al: Determining the interior layout of buildings describable by Shape grammars, CAADRIA2008, pp117-124 (2008)