L-System Design, Part 1: Introduction

Previous articles on L-Systems [1-10] described how they work and showed a variety of ways they could be used in design. This series of articles addresses issues in designing original L-Systems.

Creating an L-System for a particular purpose is neither easy nor intuitive, but, when successful, the results can be more than worth the effort. And as is the case with much such things, with practice and experience, the process becomes easier.

When designing L-Systems, it is important to keep in mind their basic properties and their inherent problems.

The Fractal Nature of L-Systems

L-Systems fundamentally are fractal generators. Although it is possible to design L-Systems that produce simple, easily understood patterns, the L-System mechanism by its nature is fractal.

This fractal nature comes from three sources:

- parallel operation on symbols
- symbols defined in terms of themselves
- repeated application of the rules (iteration)

The ways symbols can be defined in terms of themselves is of central importance. A simple example is

seed: A
rules: A → ABA
B → BBA

Here both A and B are defined in terms of themselves and each other. Repeated applications of the rules produces increasingly long and intricate combinations of the two symbols:

A
ABA
ABABBAABA
ABABBAABABBABBAABAABABBAABA
…

Although the rules are simple, the patterns that develop are nonetheless complex and not easy to characterize.

The Seed

The seed, with which generation begins is not particularly important. In most of the examples given in the series of articles on L-Systems, the seed is a single symbol. The seed can be a string of symbols, but an L-System with such a seed can always be replaced by an L-System whose seed is a single symbol.

Consider this example:

seed: ABCBA
rules: A → BC
B → AB
C → CB

A new symbol can be added as the seed and a new rule can be added replacing it by the original seed:

seed: D
rules: D → ABCBA
A → BC
B → AB
C → CB

The only difference between these two L-Systems is an additional initial generation in the second. Note that D only appears once.

Alphabet

The alphabet of the symbols used in an L-System really only matters as to the number of symbols. Symbols are arbitrary. They may be chosen for mnemonic value, but until the interpretation of a string generated by an L-System, they have no meaning.

For example,

seed: A
rules: A → ABA
B → BBA

and

seed: 3
rules: 3 → 3X3
X → XX3

are equivalent.

Generation Length

An inherent property of L-Systems is increase in length of successive generations. In fact, this limited early work on L-Systems at a time when computer memory was very limited.

It is possible to design L-Systems in which generation length does not increase. An example is

seed: A
rules: A → B
B → A

which generates

A
B
A
B
…

Such L-Systems are both contrived and trivial.

When a rule specifies replacement by more than one symbol, generation length increases. This problem is addressed in a subsequent article.

Symbol Relationships

The way that symbols are defined in terms of themselves and each other has many effects on L-System generation.

If not all symbols appear in all rules, there many be successive generations that have essentially different characteristics.

A simple and trivial example is

seed: A
rules: A → BB
B → CC
C → AA

for which the generations are

A
BB
CCCC
AAAAAAAA
BBBBBBBBBBBBBBBB
…

A subsequent article will address the issue of symbol interaction in more detail.

What is Possible

L-Systems are one of many kinds of formal grammars [11]. Different kinds of grammars have different "expressive power". The issue of expressive power is of both theoretical and practical importance.

Expressive power, roughly speaking, is a measure of what kinds of patterns a formal grammar can produce. Expressive power is measured more in terms of what patterns can be excluded than what may appear.

For example, for almost all kinds of formal grammars there are specific grammars than can produce palindromes, but they inevitably produce other patterns as well. That is non-palindromes cannot be excluded by grammars of most kinds. (They can be by L-Systems.)

What kinds of patterns L-Systems can produce will be the subject of a future article.

Interpretation

The power of interpretation of symbols in the strings produced by L-Systems has been described in other articles [1-7].

Although interpretation falls outside the scope of L-Systems proper, it is a power design tool and its possible use needs to be kept in mind when L-Systems are designed.

An L-System intended to produce a profile draft may not require any interpretation other than the think of the symbols as blocks. On the other hand, an L-System designed to draw a pattern may require interpretation of symbols as navigation and drawing actions [1].

But interpretation can be used to change L-System strings in arbitrary ways, including reordering them, deleting symbols, and so forth.

References

1. *Designing with L-Systems, Part 2: A Side Trip to Graphics*, Ralph E. Griswold, 2004:
   http://www.cs.arizona.edu/patterns/weaving/webdocs/gre_ls02.pdf

2. *Designing with L-Systems, Part 3: Back to Basics — T-Sequence Design*, Ralph E. Griswold, 2004:
   http://www.cs.arizona.edu/patterns/weaving/webdocs/gre_ls03.pdf

3. *Designing with L-Systems, Part 4: Articulated L-Systems*, Ralph E. Griswold, 2004:
   http://www.cs.arizona.edu/patterns/weaving/webdocs/gre_ls04.pdf

4. *Designing with L-Systems, Part 6: Generating T-Sequence Expressions*, Ralph E. Griswold, 2004:
   http://www.cs.arizona.edu/patterns/weaving/webdocs/gre_ls06.pdf

5. *Designing with L-Systems, Part 7: T-Sequence Models*, Ralph E. Griswold, 2004:
   http://www.cs.arizona.edu/patterns/weaving/webdocs/gre_ls07.pdf

6. *Designing with L-Systems, Part 9: Devious Interpretation*, Ralph E. Griswold, 2004:
   http://www.cs.arizona.edu/patterns/weaving/webdocs/gre_ls09.pdf

7. *Designing with L-Systems, Part 10: Profile Drafts* Ralph E. Griswold, 2004:
   http://www.cs.arizona.edu/patterns/weaving/webdocs/gre_ls10.pdf

Ralph E. Griswold
Department of Computer Science
The University of Arizona
Tucson, Arizona