Designing with L-Systems, Part 8
Non-Terminal T-Sequence Generation

The examples of L-Systems for generating t-sequence expressions in previous articles all have been terminal [1, 2]. While terminal L-Systems provide useful models, they do not exploit the power of L-Systems to generate successively more complex and detailed patterns — in this case, t-sequence expressions.

Consider this L-System:

seed: S
rules: S → pal(T)
T → motif(U,V)
U → hor(T)

It generates t-sequences expressions that are palindromes containing motifs along paths with horizontal reflection. But since T and U are defined in terms of themselves, exactly what is going on is hardly clear. The first few generations are:

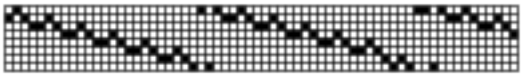
pal(T)
pal(motif(U,V))
pal(motif(hor(T),V))
pal(motif(hor(motif(U,V)),V))
…

The first three of these generations are comprehensible, but the last one is too intricate to comprehend; it is necessary to try examples and see what results.

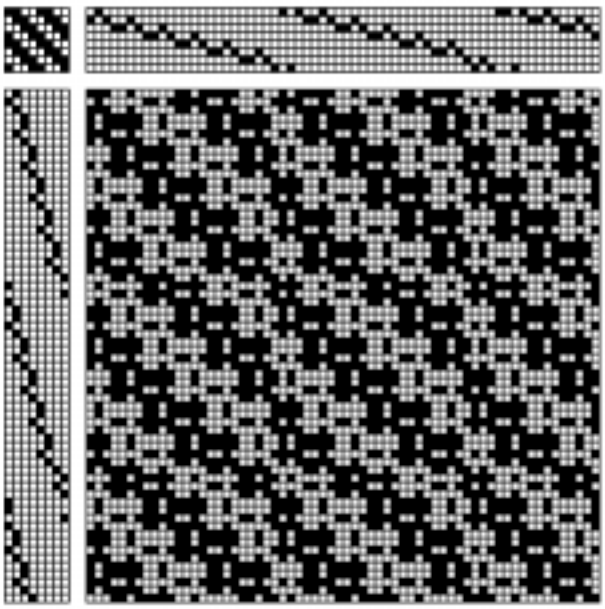
Suppose U and V have the values

U := [1,2,3,2]
V := [1,2,3,4,5,6]

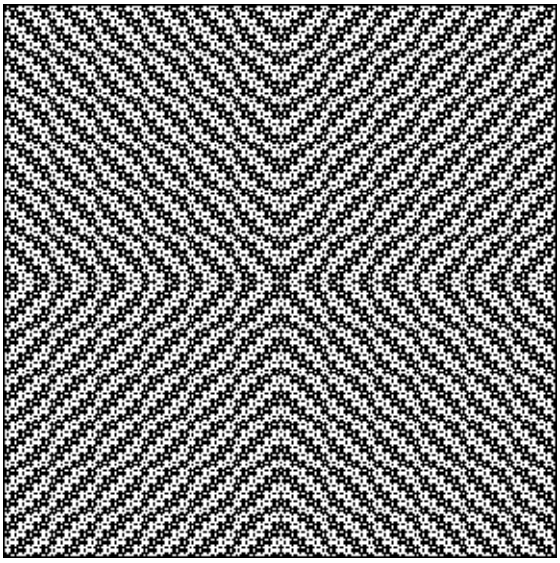
The resulting sequence starts like this:



A partial draft based on this sequence is:



and the weave is:



Here is another example of a non-terminating L-System:

seed: S
rules: S → pal(T)
T → coll(U,V)
U → pal(S)
V → ver(T)

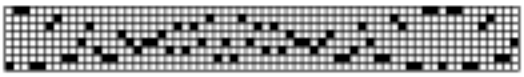
The first few generations are:

pal(T)
pal(coll(U,V))
pal(coll(pal(S),vert(T)))
pal(coll(pal(pal(T)),vert(coll(U,V))))
…

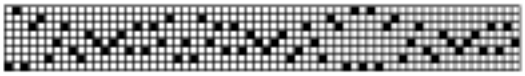
Suppose T, U, and V have the values

T := [1,2,3,4,5,6,7,8]
U := [1,1,2,2,3,3,4,4,5,5]
V := [8,7,6,5,4,3,2,1]

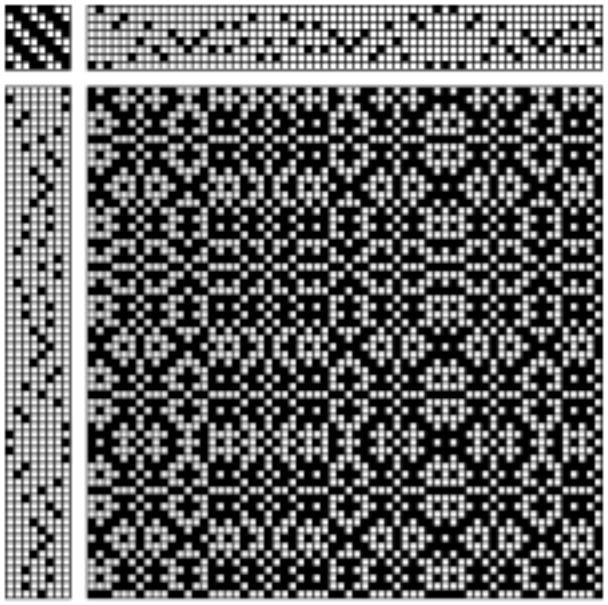
The resulting sequence starts like this:



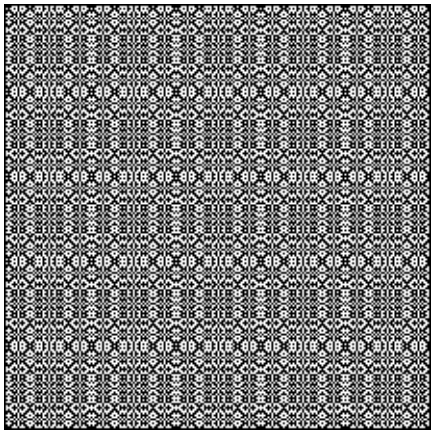
Here is a case of a defective sequence with adjacent duplicate values. This is not surprising, since the expression from which it was created is not comprehensible and hence the results unpredictable. Removing adjacent duplicates produces this:



A partial draft based on this sequence is:



and the weave is:



Of course other values for the variables give different results, sometimes very different results.

There are four problems with using L-Systems in the manner described above:

- It is difficult to design useful L-Systems.
- It is difficult to predict the results.
- It is difficult to assign useful values to undefined variables.
- The sequences produced quickly become impossibly long.

The last problem is the easiest to handle: Simply truncate long sequences so they are of a manageable length. Note the t-sequence expressions cannot be truncated; trying to do so results in invalid expressions.

The other three problems are essentially intractable if any degree of complexity it to be obtained. The best approach to problems like this is to try many alternatives, extract the useful results, and learn from the process.

For this to be practical, it is necesary to use computer programs.

References

1. *Designing with L-Systems, Part 6: Generating T-Sequence Expressions,* 2004:
http://cs.arizona.edu/patterns/weaving/webdocs/gre_ls06.pdf
1. *Designing with L-Systems, Part 7: T-sequence Models*, 2004:
http://cs.arizona.edu/patterns/weaving/webdocs/gre_ls07.pdf

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