Making Music from L-Systems

Steven Schwarz

New Music DC 2022
October 15-16, 2022

It all started with a Facebook post…

An L-System consists of

an alphabet of symbols used to form strings

a set of replacement rules of the form

symbol ⇒ string

an axiom symbol

symbols: variables and constants

variables are symbols that occur on the left in a replacement rule

constants are all other symbols

the replacement for a variable may contain a mix of variables and constants *generation 0 string:*

the axiom

generation n+1 string:

the generation n string, with each variable replaced using its replacement rule *an L-system can have a set of encoding rules of the form*

symbol ⇒ string

symbol *is a variable*

string *can only contain constants*

an L-system is designed for visualization when it has encoding rules, and its constants are drawing commands *logo turtle*

current drawing state:

location, direction, step size, turning angle

drawing commands:

`f`	*take a visible step*
`x`	*take an invisible step*
`+`	*turn right*
`−`	*turn left*

other commands exist (e.g., to change the step size) *special L-System constraints allow a correspondence*

graphics ⇔ *music*

initial direction of 0°
constant turning angle of 30°
only integer step sizes

the graphics ⇔ music correspondence

direction	⇔ *12-tone pitch*
step size	⇔ CPN *note/rest*
visible step	⇔ *note*
invisible step	⇔ *rest*

Example L-System

axiom:

F

replacement rules:

C ⇒ (F−−−−−C+F+++C)

F ⇒ (F−−−C−F++++C)

encoding rules:

C ⇒ f

F ⇒ f

Example L-System (cont.)

`F` *represents an abstract forward step*

`C` *represents an abstract step taken backwards (with the direction of all turns reversed)*

hint: read the formula for `F` *backward, switching* `C` *to* `F` *and vice versa, and switching* + *to* − *and vice versa; this gives the formula for* `C`.

Example L-System (cont.)

instance of an L-System derived from a generator for a fractal line drawing

uses only two of the four variables

- B *backward step*
- C *backward step, turns reversed*
- F *forward step*
- G *forward step, turns reversed*

Example L-System (cont.)

in terms of the turtle, as it moves on the screen:

- B *turtle goes backward above the screen*
- C *turtle goes backward below the screen*
- F *turtle goes forward above the screen*
- G *turtle goes forward below the screen*

Example L-System (cont.)

generation 0: `F`

encoding: `f`

generation 1: `(F---C-F++++C)`

encoding: `(f---f-f++++f)`

Example L-System (cont.)

generation 2:

```
((F---C-F++++C) ---
 (F----C+F+++C) -
 (F---C-F++++C) ++++
 (F----C+F+++C))
```

encoding:

```
((f---f-f++++f) ---
 (f----f+f+++f) -
 (f---f-f++++f) ++++
 (f----f+f+++f))
```

Example L-System (cont.)

generation 3: 64 steps *enough?*

generation 4: 256 steps *just about right!*

generation 5: 1024 steps *too many?*

When I make music from an L-System

strings generation 0 through generation n-1 form an n-voice, nested, canon by augmentation

each note/rest in each generation takes the same time as the entire melody that replaces it in the next generation

the axiom note lasts throughout the piece

When I make video from an L-System

the canon just described is synchronized to an animation of the final generation fractal line drawing as it is being made by the turtle

note:

Only the final generation is animated, while all generations go into the canon. The reasons for this are complicated and have to do with how our senses of vision and hearing react differently to superposed stimuli.

How I use color in animations

colors for notes are chosen randomly

all rests are colored gray

each note/rest in generation 1 gets a new color

each note/rest in the final generation gets its color from its generation 1 ancestor note/rest

How I use color in animations (cont.)

each step is colored white as it is first made (whether note or rest)

as each new step is taken, the previous step is recolored from white to the color it is meant to have

(makes it easier to follow the animation)

How I use color in animations (cont.)

the starting (resp. ending) location is marked by a green (resp. red) dot

if these locations are equal (or nearly so), the green dot will be drawn larger than the red dot, so both can be seen

everything happens against a black background

My Composition Workflow…

generator L-Systems

our example, with a more interesting generator:

axiom:	`F`
replacement rules:	`B` ⇒ `(B++++F-G---C)`
	`C` ⇒ `(C----G+F+++B)`
	`F` ⇒ `(G---C-B++++F)`
	`G` ⇒ `(F+++B+C----G)`
encoding rules:	`B` ⇒ `f`
	`C` ⇒ `f`
	`F` ⇒ `f`
	`G` ⇒ `f`

generator L-Systems (cont.)

example…

Further types of L-System

branching L-Systems…

Sturmian word L-Systems…

single-vertex flat-fold origami L-Systems…

one-dimensional L-Systems used to sample points on strange attractors…

branching L-Systems

new turtle drawing commands: [*and*]

keep a stack of saved drawing states

[*push current drawing state onto the stack;*
current drawing state continues to evolve

] *pop saved drawing state off the top of the stack;*
replace current drawing state with it

follow one branch and later return to follow another

branching L-Systems (cont.)

example…

Sturmian word L-Systems

a Sturmian word is an infinite sequence of 0s and 1s with no repeating finite sub-sequence and minimal complexity consistent with being non-repeating — i.e., they are the simplest transcendental objects

- *each Sturmian word can be represented by an L-System with variables* `0` *and* `1`

- *the axiom is* `0` *or* `1`*, according to the first letter in the Sturmian word*

Sturmian word example

The Fibonacci word is Sturmian. It is generation ∞ of this L-System with no encoding rules:

axiom:

```
0
```

replacement rules:

```
0 ⇒ 01
1 ⇒ 0
```

generation 0: 0	*ratio of 1s to 0s*	0
generation 1: 01	*ratio of 1s to 0s*	1
generation 2: 010	*ratio of 1s to 0s*	0.333_
generation 3: 01001	*ratio of 1s to 0s*	0.666_
generation 4: 01001010	*ratio of 1s to 0s*	0.6
generation 5: 0100101001001	*ratio of 1s to 0s*	0.625
generation 6: 010010100100101001010	*ratio of 1s to 0s*	0.615384…
generation ∞: the complete Fibonacci word	*ratio of 1s to 0s*	0.618033…

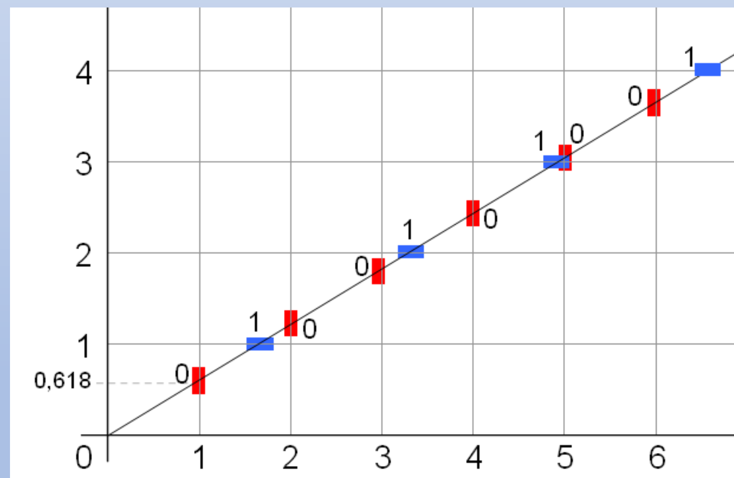
Sturmian word example (cont.)

this limiting ratio is known as the Golden Ratio –

it is the ratio of height to width for the Golden Rectangle) –

the exact value is: $((\sqrt{5}) / 2) - 1$

recover the Fibonacci word as the "cutting sequence" shown in this picture:



Sturmian word L-Systems (cont.)

the L-System that represents a Sturmian word will have rules replacing `0` *and* `1` *by sequences of* `0`*s and* `1`*s of different lengths*

the Sturmian word itself is the generation ∞ *string for the L-System that represents it*

Sturmian word L-Systems (cont.)

a new drawing command useful for encoding the axiom: |

- *step serial number added to current drawing state*
- *starts out 0*
- *increases by 1 each time a step is taken*
- *| turns right (resp. left) if the most recent step's serial number is even (resp. odd)*

Sturmian word example (cont.)

why is | useful?

suppress the last two digits of each generation of the Fibonacci word – always leaves a palindrome.

| helps to highlight this mirror symmetry

generation 2: 0 (10)

generation 3: 0<u>1</u>0 (01)

generation 4: 01<u>00</u>10 (10)

generation 5: 01001<u>0</u>10010 (01)

generation 6: 010010100<u>1</u>001010010 (10)

Sturmian word L-Systems (cont.)

example…

single-vertex flat-fold origami L-Systems

origami done with a circular piece of paper

- *creases **must** join the center to some point on the edge*
- *when finished, the origami **must** be folded flat*

each such origami is determined by the pattern of angles between adjacent creases

one can always find an ordering of adjacent creases beginning with two mountain folds, and then alternating valley and mountain folds

single-vertex flat-fold origami L-Systems (cont.)

each such origami leads to an L-System with variables `M` *and* `V` *and constants* `f`*, +, and −*

- `M` *is replaced by a sequence beginning with two* `M`*s, then alternating* `V` *with* `M`
- `V` *is replaced by a "reverse" sequence that alternates* `M` *with* `V`*, but ends with two* `V`*s (gotten by looking at the origami from the other side of the paper)*
- *+s and −s express the angles between adjacent creases*
- *these angles alternate between groups of + and groups of −*

single-vertex flat-fold origami L-Systems *(cont.)*

example…

sampling points on a strange attractor

when iterating a plane map: there are four possibilities

a fixed point	*boring!*
a limit cycle	*boring!*
instability	*boring!*
chaos	*beautiful!*

iterating a chaotic plane map yields a strange attractor

the attractor typically contains millions of discrete points, but looks like it has other structure

sampling points on a strange attractor (cont.)

create music by using a one-dimensional L-System to organize the sampling of points from a strange attractor

- *there will be only a few hundred to a few thousand sample points*
- *join adjacent sample points with a line segment*
- *quantize the slope of the segment to the nearest multiple of 30° to form a 12-tone pitch*
- *the duration of each pitch will come from the L-System's step size (not the length of the line segment)*

sampling points on a strange attractor (cont.)

L-Systems used for sampling can make use of a rich set of drawing commands:

`f`	*a visible step*
`x`	*an invisible step*
`m`	*a step visible in all but the final generation*
`e`	*a placeholder step (invisible; 0 used as step size)*

plus: a command to change the step size (always an integer)

but: no way for the turtle to turn

sampling points on a strange attractor (cont.)

animation: show the line segment joining adjacent samples while the corresponding musical tone/rest lasts

only show the sampling from the final generation of the sampling L-System, but use the canon from all generations

show segments in semi-transparent white in front of a colored image of the strange attractor

change the color of the strange attractor in sync with each sample made in generation 1

at the end, show all sampling segments shrouding the attractor

sampling points on a strange attractor (cont.)

examples…