Growing Music: Musical Interpretations of L-Systems

Peter Worth and Susan Stepney

Department of Computer Science, University of York, York YO10 5DD, UK

Abstract. L-systems are parallel generative grammars, used to model plant development, with the results usually interpreted graphically. Music can also be represented by grammars, and it is possible to interpret L-systems musically. We search for simultaneous 'pleasing' graphical and musical renderings of L-systems.

1 Introduction

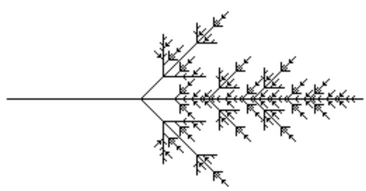
L-systems are parallel generative grammars [12], originally defined to model plant development. Starting from an axiom string, or 'seed', the grammar rules are applied in parallel to each element of the string, for several iterations or generations. For example, consider the following L-system [12]:

ω: X $\qquad$ p_1: X → F[+X][-X]FX $\qquad$ p_2: F → FF

Starting from the axiom ω, successive generation strings are:

0: X
1: F[+X][-X]FX
2: FF[+F[+X][-X]FX][-F[+X][-X]FX]FFF[+X][-X]FX

and so on. The resulting string is typically rendered graphically, by interpreting the elements as turtle graphics commands [10]. For example, interpreting F as 'forward distance *d*, drawing a line', ± as 'turn through ± δ degrees', [ ] as 'start/end branch', and X as null, then after 5 generations the example L-system renders as a 'leaf':



Non-graphical renderings can be considered. Here we consider musical renderings, and ask: "is it possible to have simultaneous 'pleasing' graphical and musical renderings of L-systems?"

2 Musical Grammars

The idea of generating music algorithmically is not new. The earliest recorded work was by the Italian monk Guido D'arezzo in 1026. Demand for his Gregorian chants

F. Rothlauf et al. (Eds.): EvoWorkshops 2005, LNCS 3449, pp. 545–550, 2005.

was so high that he devised a system to systematically create them from liturgical texts. Mozart, Haydn, and C.P.E. Bach had an interest in generative music; Mozart invented *Musikalisches Würfelspiel* (musical dice game), which involved using dice to decide which of a set of pre-defined musical phrases came next in the piece [9].

Heinrich Schenker's (1868–1935) work on the analysis of tonal master works provides an insight into the formal organisation of music. He broke pieces down into their background, middleground, and foreground [4]. These are structural levels, each of which intuitively fits the idea of description by a formal grammar. The *fundamental line* (*urlinie*) gives the tonal progression of the piece which is generally part or all of a scale. This low level structure can be embellished by expanding the components into more complicated sections until the foreground is reached.

Music as grammar has been widely investigated, eg [3][6][13]. [1] describes a context sensitive grammar for generating European melodies; these are structured around a *kernel*, the sequence of all the notes in a scale between arbitrarily chosen first and last notes, and the melody is the way the notes move around the kernel.

[11] maps the turtle drawing into musical score, by using a lookup table to map *y* co-ordinates to notes, and line lengths to note durations. [15] maps branching angles to changes in pitch, and distance between branches to duration. [14] maps the turtle's 3D movement, orientation, line length, thickness, colour, programmably into pitch, duration, volume, and timbre. [7] uses L-system grammars directly to represent pitch, duration and timbre, without going via a graphical rendering. This gives a better separation of concerns than deriving the music from the graphical rendering, and we follow that approach here.

3 Plants to Music : Finding a Rendering

First, we experiment with existing L-systems that produce pleasing-looking 'plants', and try to discover pleasing musical renderings of these.

(All the examples below are taken from [12], unless otherwise stated, and all the musical examples discussed here can be listened to at the website http://www-users.cs.york.ac.uk/~susan/bib/ss/nonstd/eurogp05.htm).

Music is essentially sequential in time: we do not want a temporal branching interpretation. We define a *sequential rendering*: interpret [/] as 'push/pop current state *except* the time'; F as 'play a note of duration 1'; a sequence of *n* Fs as 'play a single note of duration *n*'. So a note is broken by a change in pitch, by a new branch, or by the current branch ending. The *sequential rendering* of the 4-generation 'leaf' L-system is rhythmically interesting, and makes sense melodically:

Although the sequential rendering produces pleasant results, it can be improved to capture a Schenkerian background/middleground/foreground hierarchy. Jonas [4] uses the term "the flowerings of diminution" in describing the sonata form. This suggests an interpretation in which only the ends of the plant (leaves, flowers) are "heard". The middleground and background (the stem and branches) are not actually heard in Schenkerian analysis: they just give the structure from which the foreground appears.

In our *Schenkerian rendering*, interpret F as 'increase note duration by a quarter note', ± as 'move up/down one note in the chosen scale', [as 'push current state and set note duration to 0',] as 'play note according to current state, and pop', and X as null.

Under this rendering, the 'leaf' now plays as:

Despite not appearing to fit into a 4/4 framework, this melody sounds very musical, with a quite distinctive tune, even with a metronome beating 4/4 time behind it.

4 Stochastic L-Systems

Plants are all different: stochastic L-systems are used to generate plants from the same "family" but with different details. A musical rendering should similarly generate a variety of pieces in the same "style". Consider the following simple stochastic L-system (where the subscript on the arrow gives the probability that rule is chosen).

ω: F p_1: F $\rightarrow_{1/3}$ F[+F]F[-F]F p_2: F $\rightarrow_{1/3}$ F[+F]F p_3: F $\rightarrow_{1/3}$ F[-F]F

The Schenkerian renderings of three different productions of this stochastic L-system (3 iterations deep) are:

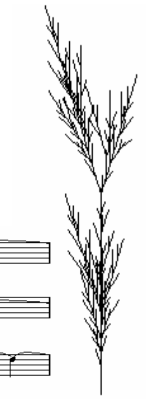
These sound 'random' but well structured, and not overly complex (as one would expect from the fairly simple nature of the rules). They do sound 'similar' to each other, but different enough to be used perhaps at different points in the same piece of music, or when combined.

5 Context-Sensitive L-Systems

Context sensitivity in L-systems gives more power as parts of the string or plant can grow differently depending on what is around them. This could be useful in music since a generated piece could build to a climax or break down at certain points. In a context-sensitive Lsystem the production rule is applied to symbol only if it appears in a specific context (between other symbols). The notation A<B>C means the string B with A to the left, and C to the right.

Consider the following context-sensitive L-system, from [2].

ω:	F1F1F1		
p_1:	$0 < 0 > 0 \rightarrow 0$	p_6:	$1 < 0 > 1 \rightarrow$ 1F1
p_2:	$0 < 0 > 1 \rightarrow$ 1[+F1F1]	p_7:	$1 < 1 > 0 \rightarrow 0$
p_3:	$0 < 1 > 0 \rightarrow 1$	p_8:	$1 < 1 > 1 \rightarrow 0$
p_4:	$0 < 1 > 1 \rightarrow 1$	p_9:	$+ \rightarrow -$
p_5:	$1 < 0 > 0 \rightarrow 0$	p_{10}:	$- \rightarrow +$



This melody, and others derived similarly, sound fairly ‘random’ (despite being deterministic); they are reminiscent of jazz solos. They do not fit well into 4/4 score notation because many of the notes are offbeat, but this just adds to their “freeform” sound. Yet the tunes always return to a main motif or phrase, that is sometimes transposed or played at a different point in the bar. For example, in the score above, the series of notes in the 1st bar is repeated in the 9th bar, but very offbeat (moved forward a quarter of a beat) and raised by 2 semitones. This kind of repetition mirrors how music is normally composed or improvised.

6 Music to Plants

Previously we started from existing L-system ‘plants’, and tried interpreting them as music. Here we take the opposite approach, of starting from musical grammar notations, and trying to produce L-system versions.

We combine the ideas of Jones [4] and Baroni *et al* [1] to write a formal grammar that generates music by recursively splitting up an event space (initially one long note) into 2 or 3 shorter, different notes. After a number of recursions we have a melody that is the length of the initial event space. Insertion rules [1] provide tonal information (we add an ‘identity’ insertion that does nothing), and halving note duration rules provide the rhythm. These insertion rules were initially written for analysis; adding probabilities as in stochastic musical grammars [5] allows them to be used for production. Variations of the rules of insertion and the rhythm grammar are given below.

We interpret d as ‘halve the duration’. We get the following grammar:

Identity:	F $\rightarrow_{1/2}$ F		
Repetition:	F $\rightarrow_{1/26}$ [dFF]		
Appogiatura1:	F $\rightarrow_{1/26}$ [d-F+F]	Appogiatura2:	F $\rightarrow_{1/26}$ [+F-F]
Neighbour note1:	FF $\rightarrow_{1/26}$ [Fd+F-F]	Neighbour note2:	FF $\rightarrow_{1/26}$ [Fd-F+F]
Skip1:	F+F $\rightarrow_{1/26}$ [Fd++F-F]	Skip2:	F+F $\rightarrow_{1/26}$ [Fd+++F--F]
Skip3:	F+F $\rightarrow_{1/26}$ [Fd-F++F]	Skip4:	F+F $\rightarrow_{1/26}$ [Fd--F+++F]
Skip5:	F-F $\rightarrow_{1/26}$ [-Fd++F-F]	Skip6:	F-F $\rightarrow_{1/26}$ [-Fd+++F--F]
Skip7:	F-F $\rightarrow_{1/26}$ [-Fd-F++F]	Skip8:	F-F $\rightarrow_{1/26}$ [-Fd--F+++F]

Starting from the axiom F++F++F+++F---F--F--F, using the *sequential rendering* and the classical turtle graphical rendering, after 4 iterations we get

The tune is pleasant. The graphical rendering (to its left) looks *somewhat* plantlike, but is not very aesthetically appealing. Starting from the musical grammars, it is unclear how to add the necessary branching instructions to get pleasing-looking plants.

7 Conclusions and Further Work

We present two musical renderings that produce pleasant sounds from classic ‘plant’ L-systems. The sequential rendering is relatively naïve, yet works well. The Schenkerian rendering is inspired by an analogy between the musical theory concepts of fore/middle/background and the components of a plant, and produces very pleasant pieces.

These examples have been evaluated to a depth of 3 or 4 iterations only. There seems to be enough information in a typical L-system to create only a short melody and still be interesting. At longer derivations, the melodies begin to get dull: the same bit of music is repeating continually, albeit normally transposed in some way. Stochastic L-systems may help, by enforcing some kind of structure on the score but giving varied melodies. The context-sensitive L-systems seem to offer the best potential for creating longer pieces of music, since identical parts of the string in different places can grow differently, so the piece can actually “go somewhere” rather than repeat the same pattern.

Starting from musical grammars and producing L-systems from them works well musically. However, the attempt to get simultaneously pleasing graphics starting from a musical grammar has been less successful: the branching necessary for graphics is not an intrinsic part of existing musical theory, and it is not clear how to add it in. More work on the Schenkerian rendering from a music theory point of view may be valuable here.

More powerful L-systems, such as parametric L-systems, could be used to generate more complex and realistic music. One exciting possibility is the use of L-systems with environmental inputs [8]. These have been developed to model environmental effects on plant growth (sun, shade, etc), but might be applicable to music generation, to allow two L-systems growing their music together as different "instruments" to react to each other.

References

[1] M. Baroni, R. Dalmonte, C. Jacobini. Theory and Analysis of European Melody. In A. Marsdon, A. Pople, eds, *Computer Representations and Models in Music*, 187-206. Academic Press, 1992

[2] P Hogeweg, B. Hesper. A Model Study On Biomorphological Description. *Pattern Recognition* **6** 165-179, 1974

[3] S. R. Holtzman. Using Generative Grammars for Music Composition. *Computer Music Journal* **5**(1):51-64, 1981

[4] O. Jonas. Introduction to the Theory of Heinrich Schenker. Longman 1972

[5] K. Jones. Compositional Applications of Stochastic Processes. *Computer Music Journal* **5**(2):45-61, 1981

[6] F. Lerdahl, R. Jackendoff. *A Generative Theory of Tonal Music*. MIT Press, 1983

[7] J. McCormack. Grammar-Based Music Composition. In Stocker *et al*, eds. *Complex Systems 96: from local interactions to global phenomena*, 321-336. IOS Press, 1996

[8] R. Měch, P. Prusinkiewicz: Visual models of plants interacting with their environment. *Proc. SIGGRAPH 1996*, 397-410. ACM SIGGRAPH 1996

[9] A. Moroni, J. Manzolli, F. Von Zuben, R. Gudwin. Vox Populi: Evolutionary Computation for Music Evolution. In P. J. Bentley, D. W. Corne, eds, *Creative Evolutionary Systems.* Academic Press 2002

[10] S. A. Papert. *Mindstorms*. Harvester Press 1980

[11] P. Prusinkiewicz. Score Generation with L-Systems. *Proc. Intl. Computer Music Conf '86*, 455-457,1986

[12] P. Prusinkiewicz, A. Lindenmayer. *The Algorithmic Beauty of Plants*. Springer 1990

[13] C. Roads. Grammars as Representations for Music. *Computer Music Journal* **3**(1):48-55, 1979

[14] D. Sharp. LMUSe, 2001. http://www.geocities.com/Athens/Academy/8764/ lmuse/

[15] F. Soddell, J. Soddell. Microbes and Music. *PRICAI 2000*, 767-777. LNAI 1886, Springer 2000