Martin Supper
Department of Music
The University of the Arts Berlin
Fasanenstra§e 1B
D-10623 Berlin, Germany
supper@hdk-berlin.de

A Few Remarks on Algorithmic Composition

Computer music is music that cannot be created without the use of computers. There are many examples of computer music that use the term in a less narrow sense, however. The performance of baroque music on digital equipment employs computers, but the very same compositions might be performed on other more suitable instruments. This article focuses on the use of computers for generating algorithmic compositions. I will define different categories of algorithmic compositions and discuss which of these deserve the designation *computer music*.

If one considers various methods of algorithmic composition, considerable importance can be allotted to differentiating between construction and resultant form, i.e., between compositional idea, its realization in the musical score, and its auditory perception. Computer-aided algorithmic composing consists of constructing or selecting algorithms to generate a composition. These constructions are then evaluated and assessed according to whether the resultant forms seem to work musically. To put it another way, computer-driven differentiation between algorithm and resultant—that is, the received form—is a peculiarity of computer-aided algorithmic composition. Only rarely is the process of generating the composition, and thus the construction of the algorithm, intentionally made visible.

The theory of algorithmic composition usually distinguishes between score synthesis (the computer-aided working out of a composition, usually for traditional acoustic instruments) and sound synthesis (the computer-aided working out of a synthetic sound that can only be heard through loudspeakers). This distinction has its roots in the traditional differentiation between score and instrument, but a computer-generated continuum between two different sounds, however, is both score and sound synthesis. In both types of synthesis, the appearance of events in time is structured, both globally (form) as well as locally (sound, timbre).

Computer Music Journal, 25:1, pp. 48–53, Spring 2001

The present text is concerned for the most part with various methods of computer-aided score synthesis for orchestral instruments. It does not deal with the wide area of computer-aided interactive composition in real time, nor does it deal with algorithmic sound synthesis.

The term *score synthesis* is closely related to computer music, computer-aided composition, CAO (*composition assistée par ordinateur*, i.e., composition assisted by computer), and automatic composition or algorithmic composition. Algorithmic composition does not necessarily require the application of a computer. Algorithmic processes have also been shown to be at work in certain procedures in analog electro-acoustic music studios, where they were called *semi-automatic* and *automatic* composing (Stockhausen 1971). On the other hand, Arvo Pärt does not use a computer and calls those of his pieces in which a pattern is extended, shortened, or otherwise permutated according to an algorithm "computer music"(La Motte-Haber 1996a, p. 158; La Motte-Haber 1996b, p. 23). This illustrates how closely related both terms are.

The selection or construction of algorithms for musical applications can be divided into three categories:

1. Modeling traditional, non-algorithmic compositional procedures
2. Modeling new, original compositional procedures, different from those known before
3. Selecting algorithms from extra-musical disciplines

I will discuss each of these categories in terms of individual procedures and compositions. The third category receives greater weight, because, in my opinion, it contains the more recent developments in computer music.

Modeling Traditional, Non-algorithmic Procedures

The composer Lejaren A. Hiller's approach to form is characterized by a traditional attitude. In his early computer-generated compositions, which number among the earliest experiments in score synthesis, he simply borrows traditional forms and uses the computer to make the composing easier. But he works so skillfully with the computer that new dimensions are revealed. Hiller's first three experiments produced not just single pieces, but whole classes of compositions equivalent to the selected form. His fourth experiment generated new, unpredicted forms for the first time, which directly changed his own formal understanding. From these experiments arose the *Illiac Suite* for string quartet (1955–1956), by Hiller and Leonard M. Isaacson. The suite's four movements document the four experiments Mr. Hiller and Mr. Leonard carried out to find out how a composition might be generated by computer. The first one ("Monody, Two-Part, and Four-Part Writing") programmed 16 different rules in three categories: what is allowable, what is forbidden, and what is required. These included the rules for simple polyphony. Within this rule-based structure, individual musical events were determined by chance procedures. The results were simple cantus firmus melodies of varying lengths. The second experiment ("Four-Part First-Species Counterpoint") extended the insights of the first to generate different musical styles. The third ("Experimental Music") was programmed with serial structures and techniques. The methods of the fourth experiment ("Markov Chain Music") influenced later works that used computer-aided score synthesis. Hiller and Isaacson worked here not with rules of composition, but with sequences of dependent random parameters.

The listener can hear traditional musical forms, but the algorithms used by Hiller and Isaacson to generate them remain hidden. The listener cannot distinguish whether the piece has been generated by means of algorithmic or traditional composition.

Other important representatives of early score synthesis are Gottfried Michael Koenig and Iannis Xenakis. Koenig began to develop his composition program *Projekt 1 (PR1)* in 1963. In 1966, he began work on *Projekt 2 (PR2)*. Both programs are rooted "in serial thought . . . which attempts to structure symbolic spaces" (Koenig 1993). *Projekt 1* describes a general serial compositional strategy that deliveries various results with the help of chance generators. With *Projekt 2* Koenig developed a composition that must be built step by step: individual dimensions for the different parameters of the compositional structure (instruments, division of time, harmonic/melodic area, volume); rules for selecting these dimensions (specifically or at random, singly or in groups, with or without temporal development); and rules of combination. Using *Projekt 2*, one must justify one's actions at every step and analyze one's intentions before beginning to compose.

Modeling New Procedures

The aforementioned compositional algorithms by Gottfried Michael Koenig are intended to solve general musical problems and to be able to be used by other composers. The algorithms of the composer Clarenz Barlow, on the other hand, have the character of a private language, and each was developed for a single piece only. Here the algorithms are not simply compositional tools, but are themselves—at an abstract level—an aspect of the music's concrete manifestation. Barlow claims that he would obtain the same results without the help of a computer and defends himself against the term "computer music." He notes that "One doesn't talk about `pencil music'" in focusing on the way a piece has been composed.

His numerical examination of tonality and meter has found its expression in the piano piece *oğluotobüsişletmesi*. Several sound layers run simultaneously in 14 different tempi. Quarter-tone intervals can be heard in the change from one scale to the next, but not within each scale. In *oğluotobüsişletmesi*, the metric and harmonic intensity, rhythmic and melodic uniformity, chordal density, density of attack, dynamics, and articulation were calculated according to a composed meta-structure: the algorithms (Barlow

1980; Wilson 1985). The actual piece is only one of many possible realizations.

Selecting Algorithms from Extra-musical Disciplines

The use of extra-musical algorithms is varied. Popular are *L-systems*, developed by the biologist Aristid Lindenmeyer. Lindenmeyer's L-systems are based on grammars by the linguist Noam Chomsky. But while Chomsky formalizes the production of sentences in a language, Lindenmeyer models the growth processes of plants (Prusinkiewicz and Lindenmayer 1990).

The use of L-systems is interesting to composers for two reasons. First, the often very complex growth processes can be formulated by the construction of very simple derivatives that develop an aesthetically challenging opposition. Second, nonpredictable, mainly self-similar resultant forms arise which cannot be achieved by usual compositional methods. The buzzword "self-similar," made popular by chaos theory, plays an enormous role in the discussion of computer-aided composing. In fact, a tendency can be observed to regard self-similarity alone as the solution to formal problems. However, the use of compositional rules derived from L-systems is usually not enough to generate a structurally coherent composition.

The following replacement rules are a simple example of an L-system:

a → b
b → ab

Here, a is replaced by b, and b by ab. The symbol beginning the derivation is called the axiom. If one starts with a, the following sequence results:

a
b
ab
bab
abbab
bababbab
abbabbababbab
bababbababbabbababbab

The length of the chains of symbols extends rapidly and can be calculated in this example as a Fibonacci sequence: the length of each line is the sum of the lengths of the two preceding it.

L-systems are a popular choice for compositional algorithms because they generate self-similar structures, i.e., similarities arise between the microscopic and macroscopic aspects of a possible composition. These similarities can clearly be seen in the following complex example using ten different replacement rules:

0 → 10
1 → 32
2 → 3(4)
3 → 3
4 → 56
5 → 37
6 → 58
7 → 3(9)
8 → 50
9 → 39
) →)
(→ (

If we begin with the axiom 4, the following sequence emerges:

1 4
2 56
3 3758
4 33(9)3750
5 33(39)33(9)3710
6 33(339)33(39)33(9)3210
...
10
$33(3^{6}9)33(3^{5}9)33(3^{4}9)33(33(9)3750)33(3758)33(56)33(4)3210$

Even a non-mathematician will be able to recognize self-similarity as the essential feature of the resulting sequences. Artists and musicians have been interested in this concept since the 1980s. L-systems are used to generate musical structures, although of course the musical parameters represented by the individual symbols in a sequence remain perfectly free. The simplest of these attempts are those in which the symbols are taken to represent pitch and duration (Prusinkiewicz 1986).

Figure 1. Cells, for saxophone and ensemble (1993–1994) by Hanspeter Kyburz. Third movement, fourth section, alto saxophone part.

Table 1. Rules used in producing the musical excerpt shown in Figure 1

Axiom	*Rule*
1(a)	if (a < 4), then apply Axiom 1(a + 1)
1(a)	if (a = 4), then apply Axioms 1(1) 2(1)
2(a)	if (a < 3), then apply Axiom 2(a + 1)
2(a)	if (a = 3), then apply Axioms 2(1) 3(1)
3(a)	if (a < 2), then apply Axiom 3(a + 1)
3(a)	if (a = 2), then apply Axiom 3(1)
4	4 → 4

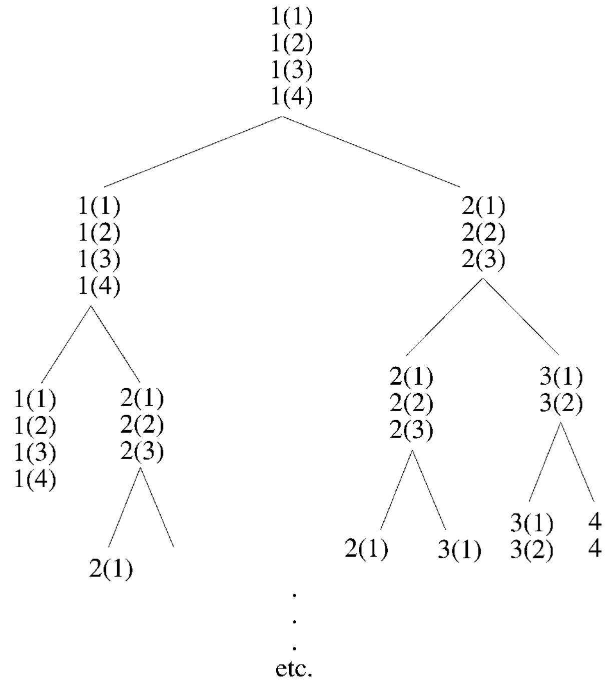
Since attending a conference on chaos theory in 1989, the composer Hanspeter Kyburz has been investigating the use of L-systems for compositional processes. The piece *Cells*, for saxophone and ensemble (1993–1994), was one of the first results. Kyburz at first composed small musical objects—small motifs—which are stored independently of one another in the computer and can be used as variables in the application of L-systems. The rules of derivation constructed by the composer define when a particular musical object is called. These rules define the compositional procedure, and they are not known to the listener, nor can they be heard. However, the process-oriented syntax of the L-systems produces the audible, self-similar structure of the composition. This method of composition might actually be perceived by the listener. When using L-systems, the composer has to define when the derivation should stop, as they tend to spool on indefinitely (Supper 1997, pp. 106–108).

Figure 1 shows an excerpt from the score. It demonstrates the formal concept—in this case self-similar structures—automatically determined by the use of L-systems.

The rules for producing the excerpt are shown in Table 1.

Here, *a* is a variable which is replaced by the figures 1, 2, 3, or 4. The tree divides, for example, if in 1(a) the variable *a* is replaced by 4, or if in 2(a) it is allotted the parameter 3. The structure of the tree arising from these rules is shown in Figure 2.

Figure 2. Tree structure arising from the rules shown in Table 1.



Kyburz uses the first 13 generations. They are marked in the score as G*1, G2 . . . G13*. I have also indicated the current derivative, for example 1(2) or 3(1). These markings relate directly to the above-mentioned musical objects.

Cellular automata are other algorithms from extra-musical disciplines used for algorithmic composition. They have been developed to simulate dynamic systems, such as the movements of fluids. For example, in fluid dynamics, a fluid particle would be represented by a *cell*. Individual particles influence each other by knocking against one another or changing places, etc., thereby defining the movement of the fluid as a whole. The easily comprehensible, elementary effects within a cell's immediate area can have an unexpected effect on the system as a whole, because each cell can simultaneously influence and be influenced by several of its neighbors. The behavior of dynamic systems has been described by cellular automata since the early 1980s.

The ideas behind cellular automata have encouraged various musicians to use them as models for compositional processes. Beyls notes, "As a composer I am interested in models of evolution and growth rather than in theories for structural design" (Beyls 1989). For Xenakis, they are a tool for constructing complex structures with minimum means. The developmental processes of cellular automata define the dynamic course of orchestral clusters in *Horos* (1986) (Hoffmann 1994).

The application of extra-musical algorithms simulating natural phenomena raises the question whether composers secretly see algorithmic composition as a way of generating natural forms naturally—forms which are taken to justify themselves by their naturalness alone. In any case, the modeling of natural processes requires the use of computers owing to the large number of mathematical calculations needed. The computer is not only used as a compositional facilitator—it becomes a necessity.

Acknowledgment

Thanks to M. J. E. Turnbull for translating this article.

References

Barlow, C. 1980. "Bus Journey to Parametron (all about *oğluotobüsisletmesi*)." *Feedback Papers* 21–23.

Beyls, P. 1989. "The Musical Universe of Cellular Automata." *Proceedings of the 1989 International Computer Music Conference.* San Francisco: International Computer Music Association, pp. 34–41.

Hoffmann, P. 1994. "Zelluläre Automaten als kompositorische Modelle. Sind Chaos und Komplexität musikalische Phänomene?" In *Arbeitsprozesse in Physik und Musik.* Frankfurt am Main/Berlin: Peter Lang Verlag/Akademie der Künste Berlin, pp. 7–18.

Koenig, G. M. 1993. *Ästhetische Praxis. Texte zur Musik*, Band 3: 1968–1991. Saarbrücken: Pfau.

La Motte-Haber, H. de. 1996a. "Alles ist Zahl. Formen pythagoreischen Denkens in der Musik." In T. Ott, and H. von Loesch, eds. *Musik befragt—Musik vermittelt,* Augsburg: Wi§ner, pp. 153–163.

La Motte-Haber, H. de. 1996b. "Struktur und Programm. Analytische Betrachtungen zur Komposition Summa von Arvo Pärt." In W. Gratzer, ed. *Nähe und Distanz. Nachgedachte Musik der Gegenwart 1.* Hofheim/Ts.: Wolke, pp. 14-25.

Prusinkiewicz, P. 1986. "Score Generation with L-Systems." *Proceedings of the 1986 International Computer Music Conference*. San Francisco: International Computer Music Association, pp. 455–457.

Prusinkiewicz, P., and A. Lindenmayer. 1990. *The Algorithmic Beauty of Plant*s. New York: Springer-Verlag.

Stockhausen, K. 1971. "Elektronische Musik und Automatik [1965]." In D. Schnebel, ed. *Texte zur Musik 1963–1970*, Band 3. Köln: DuMont, pp. 232–241.

Supper, M. 1997. *Elektroakustische Musik und Computermusik. Geschichte—Ästhetik—Methoden—Systeme*. Darmstadt: Wissenschaftliche Buchgesellschaft.

Wilson, P.N. 1985. "Formalisierte Musik—naturwissenschaftliches Denken in der Musik. Gedanken zu den stochastischen Kompositionstheorien von Iannis Xenakis und Clarence Barlow." *Neuland* (5):52–59.