Grammars as Representations for Music

C. Roads

Introduction

In a recent lecture, the linguist/philosopher Noam Chomsky posed the question: "Properly speaking, is music a language?" (Chomsky: 1979). He spoke to the question immediately after posing it: it all depends on one's definitions, and ultimately it is an unnecessary question; one shouldn't be diverted by it. I take this as the starting point of this essay; being not too concerned as to whether "music is a language" I would like to rather pose several more concrete questions about the application of linguistic concepts to music studies, and in the writings to follow try to find answers by surveying the literature. Can techniques from linguistics be usefully applied to the study of music? More specifically, can the use of grammar representations be effective in describing and specifying music structures? Are grammars intrinsically normative, *i.e.*, are they fit only to describe traditional music? Lastly, what are the implications of the representation of music as a grammar?

The idea of viewing music in terms of a "musical grammar" is not new. However, recent years have seen a resurgence of this perspective. Analogously, a score can be seen as an example of a particular language, *i.e.*, as a set of semiotic expressions conforming to a specific music grammar which is inferred by the score.

Iconic, Symbolic, and Score Representations

For the purpose of this discussion, it is useful to establish certain working definitions and clarify certain concepts. First, it is necessary to distinguish between "iconic" (or "analog") and "symbolic" representations. As stated in (Sebeok: 1975, p. 242): "A sign is said to be iconic when there is a topological similarity between the signifier (the sign) and its denotata (*i.e.*, what it represents)." (My parenthetical remarks, C.R.) An example of such an iconic or analog representation is a sequence of numbers stored in a computer memory which correspond in value to the shape of an acoustic signal. In computer music, a common form of iconic or analog information is typically raw, uninterpreted data usually obtained by applying a transducer to the physical manifestation of sound. In this case, the patterns of the numbers mirror the patterns of the waveforms.

In contrast to iconic signs, the more formal "symbols" can be defined as follows: "A sign without either similarity or continguity but with only a conventional link between its signifier and denotata . . . is called a symbol." (Sebeok: 1975, p.247). Symbols are most often combined into formal languages. The symbols and their syntactic arrangement have functional meaning within the language, enabling them to be interpreted. Such symbols do not usually mirror the surface structure of a composition, rather, they represent the "background" interrelations (or "deep structure"). An example of a symbolic notation for traditional music can be found in the language of chords and chord sequences, *e.g.*, "V - I - VI - II_7 - VII^6 - III - I^6" or in the labeling of movements, *e.g.*, "AbbaA."

Standing between the iconic and the symbolic are most scores, which are intermediate representations. Graphic scores and tablature tend to be more iconic, while traditional stave notation contains more formal elements such as note heads and stems, dynamic terms (*ppp*, *pp*, *mp*, *etc*.), accidentals, and the like. In both cases, there is typically room for performance and analysis interpretation, wherein the background or macrostructure, only inferred by the score, is articulated, or explicated.

A grammar representation of music is a symbolic representation. A grammar usually represents a generic class of compositions. Specific compositions can be represented in terms of a grammar by means of a list of grammatical production rules, or by means of a "parse tree" which graphically depicts the syntactic structure of a composition. At the highest level, a composition is represented by a single symbol, *e.g.*, Σ called the "root token" from which other tokens representing successive sub-units of a composition emerge. "Token" is the term used to denote a symbol in a grammar or parse tree. These formations of tokens can be described by means of concepts and notation developed in formal language theory. A variety of tools exist which can be adapted for various musical purposes.

Limits and Powers of a Grammar Approach

One of the limits of a grammar approach is implied in the distinction made earlier between iconic and symbolic representation systems. Before a sound signal can be represented symbolically, it must be transformed from a continuous domain into the discrete, symbolic domain. As Saussure noted long ago, this discretization is not given in nature. In a computer-aided sound analysis system, considerable signal-processing will be necessary to obtain the pertinent data necessary to transform musical sound from the low-level acoustic domain through the intermediate level of score representation and finally into a high-level, symbolic notation (*cf.* Moorer: 1975). For compositional uses, this situation is slightly less restrictive, in that composers can coordinate symbolic and sonic representations by working in each domain separately and mapping the symbolic to the sonic. In this case it makes sense to treat the sonic domain as being composed of discrete "sonic objects" (the "objects sonores" of Schaeffer: 1966). Naturally, objects can be linked and blended for a continuous musical flow. As is the case in any representation system, the use of grammars relies on certain biases; the notation system for formal grammars is very useful for representing multi-leveled macrostructures where broad macro-sections encapsule substrata consisting of sound objects on the lowest level. Clearly the musical scope of such a representation system is very broad. However, while it may be "theoretically possible" to generate "any" structure with any one system, there is always a sharp pragmatic distinction between what can be done easily and what is difficult to do – between what can be done with some elegance and what requires ad hoc patchwork to be accomplished. Of course, to be of any use, the grammar model itself must embody a certain explanatory strength (for purposes of analysis) or expressive power (for purposes of composition), *i.e.*, a weak or trivial grammar model of a composition is probably less effective than a strong, non-grammar model.

One last limitation of a grammar representation is that it is purely structural (as described here). Other facets of compositions, such as sociological content, *etc.*, are left open.

Nevertheless, some words about the power and scope of a grammar representation are in order. There is a very broad range of music which can be described naturally using grammars. Theorists have constructed grammars for describing Bach Chorales as well as musique concrête. Any music which can be partitioned can be described by a grammar. With particular reference to computer music, any composition which uses editing, be it of files or tape, can be represented by a grammar. Furthermore, a great deal of study has already gone into the properties of abstract grammars; their features are well understood. A rich collection of software tools has developed around these grammars through computer science, including parsers, compilers, reverse-compilers, and compiler-compilers, or parser generators.

Lastly, it should be evident that grammars provide a unified, multi-leveled, generative model. Parse trees have already proven useful in describing known compositions, and grammars may be used to test both analytical and compositional hypotheses.

Introductory Notes on Formal Grammars

For brevity, only the most general properties of grammars will be cited here; more tutorial information on formal grammars is readily available. Either (Salomaa: 1973) or (Aho and Ullman: 1972) is an excellent source. Some main results with musical implications are summarized in (Roads: 1978).

A formal grammar is a structural description of a formal language. The grammar specifies valid terms within the language and the rules which determine valid constructions. For musical purposes, statements within music grammars are representations of music structures – as a collection of sounds conforming to some compositional syntax. The notation system for abstract grammars is comprised of a collection of *graphemes* or characters, *e.g.*, $a, b, \ldots 0, 1, \ldots n$. Out of one or more graphemes are composed individual *tokens* or symbols, *e.g.*, $aa, bb, a^2, \ldots$ This collection of tokens along with the null token $\emptyset$ make up the *vocabulary of tokens* V. Within the vocabulary of tokens is defined the *alphabet of terminals* T. The terminals are the lowest level tokens; they correspond to the lowest level of microstructure that the grammar describes (the "surface structure"). A *language* L is defined as the subset of finite concatenations of tokens in T. The terms *sentence*, *word*, and *string* are often used synonymously in formal language theory to denote sequences of terminal tokens in L. An *alphabet of non-terminals* N are defined in V which represent categories of macrostructures (or "deep structure"). A *sentential form* is simply a sequence of non- terminals. The formula "$\alpha \rightarrow \beta$" is a *production* or *rewriting* rule where the right side is a replacement of the left side; this constitutes an algorithm for generating sentences. A *derivation* is a complete sequence of productions leading from the highest non-terminal token to a terminal, *e.g.*, $\Sigma \rightarrow A1$, $A1 \rightarrow A2$, $A2 \rightarrow a3$ is a derivation from the root Σ to the terminal $a3$. Finally, a *generative grammar* G can be characterized as a quadruple: $G = (N, T, \Sigma, P)$ where N is the alphabet of non-terminals, T is the alphabet of terminals, Σ is the root token, and P is the collection of production rules of the form: $\alpha \rightarrow \beta$, where $\alpha, \beta \in (N \cup T)^*$, and $(N \cap T) = \emptyset$. ("$*$" denotes the powerset.)

A Brief Consideration of Grammar Types

It will be useful to distinguish some of the relevant properties of some major grammar types as a prelude to studying how various kinds of grammars have been or might be applied to music. In this section six major forms will be very briefly studied. Again, for more formal details, see (Salomaa: 1973). The following paragraphs synopsize a part of (Roads: 1978). The six major forms to be discussed here include: *free*, *context-sensitive*, *context-free*, *finite-state*, *transformational*, and *regulated* grammars.

Grammar Types

A *type 0* or *free* grammar is a theoretical construct which imposes no restrictions on the form of the production rules. This allows intermediate strings to either expand or contract in length, which could produce the curious case of a "phantom macrostructure", *i.e.*, a macrostructure to which there corresponds no microstructure! Further, a type 0 grammar by definition allows the production of truly infinite (recursively-enumerable) strings as well as null productions, both of which make no musical sense.

A *type 1* or *context-sensitive* grammar derives its name from the production form: $A \; \alpha \; B \rightarrow A \; \beta \; B$, where α produces β in the context of the tokens A and B. The length of α must be less than the length of β. A type 1 grammar is less open than a type 0 grammar; in particular, the production $\alpha \rightarrow \emptyset$ is illegal. Both for descriptive and compositional purposes, a case can be made for the usefulness of context-sensitive productions. However, certain complications appear when context-sensitive production forms are used within a grammar. These complications are of two kinds. First, since the well-formedness of strings generated by a type 1 grammar is undecidable (Chomsky: 1963), ambiguous productions are quite legal, which means that there is no way of recovering a unique phrase-structure (*i.e.*, derivation) from sentences generated by a type 1 grammar. This complexity occurs since either side of a type 1 production rule may be a string of tokens, and this makes them difficult for use in music analysis applications. Type 2 or context-free grammars also allow ambiguous productions, but tracing their derivations is simplified, since any terminal need only be reduced to a single non-terminal rather than an entire string.

A second complication involves the issue of implementation. The use of a type 1 parser would entail multiplying the number of production rules by the number of contextual possibilities. This has two ramifications: first, it makes the specification of the grammar a rather onerous task, and second, the number of recognition steps in a parser of such a grammar becomes combinatorially explosive, since continual cross-references must be made into the production tables. Fortunately, it is possible to incorporate context-sensitive features into a grammar constructed only of context-free production rules, which are much more straightforward to parse. This is made possible *via* the use of *control procedures* or a *control language* which regulate the decision as to what production rules apply at any point in the parsing. Regulated grammars are further discussed below. Even with the use of simpler context-free productions, context-sensitive features can propagate ambiguity in a grammar and its associated expressions, and hence their use should be limited.

Grammars called *type 2* or *context-free* have turned out to be very useful with regards to natural and programming languages, primarily due to their relative ease of handling. Specifically, while a type 1 grammar allows strings on both sides of a production, a type 2 grammar expands only one non-terminal token (on the left-side of a production rule) at a time. Thus, production complexity is a simple linear function of the number of non-terminals in a derivation. The power of a type 2 grammars for music lies in their ability to represent multi-leveled syntactic formations, since any non-terminal (representing a macrostructural category like motive, phrase, sentence, section, or movement, and the like) may generate a string of tokens at a lower level. The ability of type 2 grammars to generate strings of tokens (which might represent a phrase, or phrases) in any one production makes them more elegant from a representational viewpoint than type 3 or finite-state grammars. However, the strongest power of a context-free grammar over a finite-state grammar results from production rules of the form: $A \rightarrow \alpha \; A \; \beta$, where A is a non-terminal, and α and β are members of V. Rules of this form are called *self-embedding*; in music, they result in *nested-motivic* formations. It is precisely this issue which convinced Chomsky that type 3 grammars (such as Markov chains) were too weak to handle phrase-structured languages (Chomsky: 1957).

A *type 3* or *finite-state* grammar is characterized by the restriction that no more than one non-terminal token may appear on each side of any production rule. A type 3 grammar with production rules of the form: $A \rightarrow a$ or $A \rightarrow aB$, where a is a terminal and A and B are non-terminals, is called a *right-linear grammar*, since the non-terminal B appears in the rightmost place in the rule. The ability of a type 3 grammar to represent any kind of intricate, multileveled tree structure (such as the macrostructure of a music composition) is extremely limited, because of the restriction concerning one non-terminal on the right-side of a production rule. A representation for music should at least have the power to handle nested phrases and motives, constructions which are technically excluded from type 3 grammars. Chomsky has characterized Markov processses as a type 3 grammar. (Chomsky: 1957, 1963) and has pointed out their inability to handle phrase structure. This has nothing to do with the issue of a stochastic process versus a deterministic process *per se*. In fact, a type 2 grammar may be made stochastic. The issue is rather in the form of the production rules, where a Markov system is limited to a linear, step-by-step production process whereas a type 2 production can generate a complete phrase (string of non-terminals) at once. In conclusion there seems to be no reason for limiting a music representation system merely to type 3 grammars.

The classical *transformation grammar* was proposed by Chomsky to describe the structure of natural languages like English. The grammar is tripartite, comprised of three components: (1) a *phrase-structure grammar* which generates abstract kernel sentences (2) *a set of transformation rules* which map kernel sentences into English sentences (3) *morphophonetic rules* which map English sentences into streams of phonemes (spoken sequences). Chomsky's reasons for adopting a mediating level between phrase-structure and morphophonetics stem from certain complexities associated with handling certain natural language constuctions such as active-passive verb relations, auxiliary verbs, and the like. No clear musical analogy between these natural language constructs and music constructs exists. Thus there is little justification for importing an obligatory transformation level (which would modify the effect of the production rules) into a music grammar. However, the process of morphophonetics – the process of interpreting abstract tokens into the lexicon of sounds – does have a pertinent musical analogy. This corresponds to the musical process of orchestration. To distinguish its technical meaning in the context of grammar systems, the term *lexical mapping* is useful. It is debatable whether the operations which comprise the lexical mapping could be characterized properly as "transformations" in Chomsky's technical sense of the term, and this debate is beyond the scope of this paper.

The last form of grammar discussed here is the *regulated grammar*. A regulated grammar typically involves a context-free (type 2) grammar controlled by procedures which determine not the *form* of the production rules but rather at which point they are to be applied. Through the use of control procedures or a control language the generative capacity of a context-free grammar can be greatly increased. A variety of regulated grammar systems have been devised and are discussed in (Salomaa: 1973). A design of a composing language incorporating control procedures as well as a lexical mapping system regulated by a control language is discussed in (Roads: 1978). Specifically, control systems over a grammar can be straightforward and useful ways of handling such special-cases as productions with several alternatives (a control procedure may be invoked to resolve the ambiguity) or recursive productions (a control procedure may be invoked which controls when to stop the recursion). Another device besides control procedures is available. A *control language* works by assigning a label to each production rule. The control language is said to function *over* the grammar. It contains expressions which are simply sequences of labels that determine what production rules are valid at any stage in the parsing process. It make sense to apply a control language only to a context-free grammar (type 2), since the generative capacity of a type i regulated grammar is the same as that of type i unregulated grammars for $i \neq 2$.

A Survey of Grammars Applied to Music Studies

In this section, we enter into the main part of the essay: a survey of the use of grammars to represent music structures. In recent years an increasing number of antecedents to support the idea of applying linguistic techniques to the study of music have been put forth. In the text to follow, the work of Ruwet, Nattiez, Laske, Smoliar, Winograd, Moorer, Jackendoff and Lerdahl, and others will be briefly synopsized.

Ruwet

The studies of Nicholas Ruwet fall into two periods, with somewhat contrasting visions as to the place of grammars in music studies. In his 1966 article: "Methodes D'Analyse en Musicologie" in (Ruwet: 1972), Ruwet the musicologist proposes a method of analysis which applies techniques of segmentation (or "partitioning") based on criteria of repetition (equivalences), opposition, and transformation. He uses linguistic techniques to analyse works by Debussy, 14th Century *Geisserlied*, and extracts from works by Rameau. He emphasizes the use of formal grammars as a tool for *verification of the analysis* (through a test of synthesis based on the grammar constructed from the analysis) and as a *unified formal model* of necessarily disunited analysis techniques.

By 1975, however, Ruwet, by then an established generative grammarian linguist, had altered his views on music theory construction (Ruwet: 1975). Following a line of reasoning argued earlier by Chomsky, Ruwet prefers a more "top-down" approach to music analysis, in which the construction of a theoretical model *precedes* the testing of its validity. He contrasts this top-down approach to the bottom-up method used in his earlier articles and in the work of Nattiez, which is strictly empirically-based. To Ruwet, the top-down approach is more scientific, in that like a theoretical physicist, one validates a theory by attempting to falsify it. Like Chomsky, he argues that there are no procedures for discovering theories; there only exist methods of falsifying them. Thus, strong, general, interesting music theories need to be intuitively proposed, and only then scientifically validated or invalidated. Ruwet goes on to suggest that a form of generative grammar (in detail not equivalent to Chomsky's *Aspects* schema (Chomsky: 1965) for natural language) could be an excellent model for such a purpose. The function of a grammar is thus shifted from that of a validating device of an inductive (bottom-up) theory to that of a starting point in the construction of a hypothetico-deductive (top-down) theory. Lastly, Ruwet believes that music analysis cannot simply be score-bound, but rather must treat the sonic domain more comprehensively.

Nattiez

The position of J.-J. Nattiez is somewhat different from that of Ruwet. In particular, Nattiez takes an empiricist's approach; he does not believe that it is possible to propose a significant set of synthetic (generative) rules without first passing through a process of detailed taxonomic description (Nattiez: 1975). Based on taxonomical procedures of the early Ruwet (who, in turn borrowed from early poetic analyses by Jakobson (*cf.* Jakobson: 1973)) Nattiez gives examples of analyses on scores by Stravinsky, Brahms, and Varèse. He goes beyond the early Ruwet in asserting the need for inter-textual analyses of whole families of scores in order to recover pertinent stylistic traits. As regards grammars, he suggests that "syntagmatic grammars" may be constructed to represent an analysis of a single work in the form of a system of generative rules for the specific signs in a piece. However, a large part of Nattiez's book is devoted to arguing the semiological specificity of music as distinct from natural language. Thus, he criticizes the bald translation of Chomsky's *Aspects* schema for natural language to music. While noting that one or another part of the theory may be useful, he specifically attacks the whole notion of importing a "transformational" stage into a music grammar.

Nattiez notes the analogy between Schenker's hierarchical graph notation and the tree structures (parse trees) of syntactic analysis which he himself uses. But he criticizes identifying Schenker's *Ursatz* (or "fundamental structure" (Forte: 1959)) with Chomsky's semantically-tied *deep structure*, feeling that this leads to a normative conception of music.

In a later essay (Nattiez: 1977) he confronts the use of grammar models built by analysis of extant works for compositional purposes. While citing no examples, he asserts that this can only produce "pastiches" of little interest because analysis models are descriptively poor in relation to actual works.

Laske

A common thread running through all of Laske's writings is the grammar form. He based his explorations into a "generative theory of music" on a grammatical conception which, in form, corresponds to Chomsky's schema of *Aspects of the Theory of Syntax* (Chomsky: 1965). The grammar is thus the form of the theory. (Laske: 1975) is a particularly comprehensive consideration of grammars for music. This grammar conception was, in Laske's mind, to be founded on "an explicit and formal model of empirical musical activity." The construction of Laske's theory rested on what he calls a "sonological/psychological base." Finding an integrated model for such a base to be nonexistent, Laske, in his later work, turned to the "investigation of the strategical task environment of such grammars." He characterizes this work as "studies in musical cognition." Its purpose is to study the formal properties of cognitive tasks undertaken in processing musical input. Instead of producing a taxonomic analysis of music structures, Laske has turned to what is essentially a project in cognitive psychology.

One of the most useful concepts of Laske's earlier studies is the notion of *sonology*, a generalization of the linguist's *phonology*, which is said to "express the relationship between the syntactic structure of a music and its physical representation in so far as this relationship is determined by grammatical rules" (Laske: 1975, section 1, p. 31). The issues involved in sonology are vital to composition, as well as analysis of more timbrally-complex works, since they touch upon systems of orchestration (or lexical mapping) and psychoacoustics.

In his later essays, Laske uses the notion of a grammar particularly as a model for musical task strategies, *i.e.*, the order of tasks is modeled as a list of production rules which define their sequence. In particular, Laske has adopted the use of "programmed grammars" (a form of regulated grammar) which specify the order in which task-production rules are to be applied. As his most recent writing demonstrates (Laske: 1977), (Laske: 1978), he has retained a grammar conception of musical structure (as multiple layers of musical phrases linked by production rules) but he has incorporated this interest in music structures into cognitive musicology, which associates the music structures to the behaviour which produced them.

Smoliar

For the purposes of this discussion, it makes sense to partition Smoliar's writings on grammars and music into two phases. In the first phase, Smoliar suggested the use of grammatical constructs for an analysis procedure which would proceed to parse a music signal from the bottom-up. Specifically, Smoliar's proposed music analysis system would "decompile" the electroacoustic signal of some music by generating what he calls "antiproductions." Such a decompilation process would ultimately produce a model of music perception, to be characterized in terms of a "high-level language" (Smoliar: 1976).

In the second phase, Smoliar has implemented an interactive language for music theorists in which structural descriptions of music compositions may be encoded. The program generates music strings based on the structural descriptions. The primitives of the language are a clever adaptation of Schenker's functional model of the tonal language. Starting from a musical "photo-structure" (Smoliar's term for Schenker's *Ursatz*) the theorist may specify various commands which invoke rewriting rules that derive a surface structure from the proto-structure. In graphic terms, the program generates a tree from the root; commands generate branches, delete branches, rearrange the levels of the tree, and so on (Smoliar: 1977).

Moorer

In Moorer's 1972 paper "Music and Computer Composition" the subject is the consideration of various algorithms for generating a certain class of compositions by computer (Moorer: 1972). The essence of the grammar aspects of this paper may be summarized as follows. First, Moorer considers both context-free (type 2) and finite-state (type 3) grammars and finds them lacking. In particular, they both show a lack of context-sensitive features needed for describing certain music structures. Specifically, finite-state Markov chains of high-order (eighth-order) produce mechanical sequences of "spliced" whole phrases, and low-order Markov chains generate sequences which meander aimlessly, while a BNF (context-free) notation is shown to be lacking in context-sensitive subtlety. Next, he suggests that heuristics may be added to these systems to augment their power. A model is then demonstrated, consisting of Markov-type algorithms extended by selection heuristics. Elsewhere in the paper, he suggests that a context-free grammar may be augmented with "transformational" procedures, thereby enhancing its context-sensitivity.

Winograd

Terry Winograd is presently best known for his work in the field of artificial intelligence on natural language understanding systems. In (Winograd: 1968) he describes a computer program which analyzes the harmonic structures of various tonal compositions. The program utilizes a "systemic grammar" in which syntax and semantics are closely related. In particular, semantic routines are used to guide the syntactic parsing. While a comprehensive treatment of music semantics is beyond the scope of this paper, the notion of semantics as used by Winograd is, briefly, the functional harmonic description of a piece of tonal music. Winograd's program first does a preliminary syntactic analysis of a music fragment which eliminates its context-sensitive features, a technique used in some compilers. This preliminary phase also includes processing which reduces the number of paths in the parsing. The program carries out some initial parsing, and assigns a degree of "meaningfulness" (in terms of the aforementioned notion of semantics) to any possible parsing. In this way, parsing paths that are grammatical with respect to the codified harmony but not very meaningful are eliminated.

The program was applied to selections from Schubert (*Opus* 33, *No.* 7) and Bach (*Chorales* 12 and 57) and the results are convincing. Winograd saw his system as a general model for semantically-directed parsing.

The principle of using syntactic parsing as a first analysis and calling semantic routines when consultation is required was carried over into Winograd's natural language understanding system SHRDLU. Semantic knowledge in this system could be broken down into two components: one, a data base of facts and assertions, and two, a set of procedures for deducing things about what is understood. Hence, "understanding" is active (Winograd: 1973). The SHRDLU system had the capacity to "learn" from its environment, both from externally-fed and internally-deduced information. One of the many notable technical aspects of Winograd's work is the expanded notion of grammar, as more than simply a list of production rules.

Lerdahl and Jackendoff

Lerdahl and Jackendoff have presented a new music analysis methodology which derives much meta-theoretical inspiration from Chomsky, while not importing exactly linguistic models to music (Lerdahl and Jackendoff: 1977). Their generative model is a theory specifically of tonal music, with implications for other musics and for musical cognition in general. Their model uses four types of analysis to derive layers of musical organization in pitch and metrical structure, the only two domains of organization discussed. While not asserting that all music is at all times organized on multiple levels, they concentrate on hierarchical aspects. The rules which assign structural descriptions are of three types: *well-formedness rules* which assign possible (tree) structures; two, *preference rules* which select more coherent and compelling structural descriptions from possible ones; and three, *transformational rules* which are needed to describe certain special cases such as elisions. These "transformational rules" are not meant to be exactly equivalent to their linguistic counterparts. While they introduce the notion of transformations into their description system, they admit that then "the problem is to constrain admissible transformations" since any phrase may be characterized as a transformation of any other. They also suggest that in Chomskian linguistics the role of transformations has been somewhat weakened in favor of an enriched phrase structure grammar. They then present examples of their analysis technique, and show how it differs from Schenker's, which they find too inexplicit. The use of the term "deep structure" is debated in connection with the notion of "archetypal music forms." While dismissing the normative connotations of "archetypes" they suggest that they may be a way of classifying perceived "regularities" or "congruencies" in phrase structures, particularly for tonal music, which may in turn suggest how music is processed by listeners.

Roads

The subject of (Roads: 1978) is the exploration of the notion of using "composing grammars" as a means of composing music structures. The paper briefly surveys some applications of grammars to music, and introduces some conceptual and notational tools from formal language theory which are adopted in order to describe the structural properties of various grammars. A graphic notation form for presenting formal grammars as tree structures is developed in parallel with a symbolic metalanguage for specifying grammars. A design for a composing language (COTREE) which references a composer-supplied grammar is detailed. Both the grammar specification language (TREE) and the composing language (COTREE) make use of context-free rewriting rules which can be augmented by control procedures. A syntax for control procedures is provided. Some computer programs useful for compiling composing grammars and realizing expressions in COTREE are discussed. A study of the semantics of compositions produced by grammars is presented. The process of mapping an abstract syntactic form into a lexicon of sound objects (*i.e.*, lexical mapping) is discussed in some detail, as is the related notion of sonology. An example of a composing grammar is given.

Other Studies

A number of other studies involving grammars have been undertaken. Among these is the paper of Lindblom and Sundberg in which the generative grammar approach is applied to the study of simple melodies (Lindblom and Sundberg: 1970). The methodological approach of this paper is probably of more general interest than are the specific results. Ulrich has described a system for analyzing the chords and their functions in harmonic jazz compositions, with the goal of modeling a form of jazz improvisation (*viz.* harmonic rhapsodizing) (Ulrich: 1977. While this system uses a chord grammar for identifying chords, different kinds of comparison procedures are used to identify key centers, and a functional analysis of the chords involves a table-lookup process.

In a separate musical context, there have been several other instances of grammatical models embedded in composing languages for computer music. One of the earlier grammatical constructions was implicitly embedded in Leland Smith's SCORE language, in that a *motive* specification facility in effect defined a context-free production. Transformations such as rhythmic offsets and retrogression could be applied to the motives (Smith: 1972, 1973). Buxton's hierarchical data structures (Buxton: 1978) allow a composer to work with what is essentially a parse tree of a composition. The composer may work at any level of the tree (represented as a complex linked-list) to define an abstract syntactic structure which is mapped to a lexicon of sound objects in a separate phase of composition.

Conclusions

To answer the questions posed at the beginning of this essay, it appears that linguistic techniques of segmentation, parsing, and the construction of production rules and multi-leveled grammar representations can be useful in music applications. The grammar form offers clarity in modeling the surface syntax and macrostructure of compositions. The range of possible grammars is quite broad, and their expressive scope is very extensive; thus there is no intrinsic technical reason why grammars should only be useful for representing traditional music. In no music is the grammar given explicitly. Older music has been most often associated with a grammar only because more was known about the conventional background structure. Nattiez's point – that grammar models for original composition built on models of extant compositions will be useless – is well taken. As Nattiez indicates, a new poetry is only possible with a new grammar. Examples of recent composing and control languages (*e.g.*, Roads: 1978, Buxton: 1978, and others) demonstrate how the specification of original and innovative composing grammars or parse trees may become increasingly available as a compositional technique. In such a technique, maximum flexibility is achieved by logically separating the *syntactic specification* (for building a grammar for a set of scores or a parse tree for a particular score) from the *sonic specification* (the lexical mapping or orchestration from the score to sound objects).

Throughout this paper, the term "deep structure" has been used interchangeably with "background structure" or "macrostructure." Whether this term deserves special status in a musical context is a debate beyond the scope of this paper.

It is evident that grammars and parse trees have been found to be useful formal models for representing compositional syntax. The structure of music expressions and of the grammar behind them can be modeled in a very concise manner since the primitive operation in a grammar, the production rule, is already a powerful abstraction. However, in nearly every study discussed in this paper the production rule has been shown to be insufficient by itself as a representation for music, particularly in a context-free form. Why not then simply adopt the use of context-sensitive production forms?

As explained earlier, the human problems associated with specifying context-sensitive rules coupled with the technical problems associated with parsing these grammars make them unattractive as working tools. Besides these issues, the whole notion of context-sensitivity as embodied in formal grammar theory is inadequate for music. In formal grammar theory, "context" is a sequential notion, while in music, context is both parallel and sequential. Further, much performed and improvisatory music is characterized by an "interrupt-context," *i.e.*, in which a phrase may be interrupted and a new phrase may begin. More work on "interrupt-driven" grammars needs to be carried out in order to arrive at a useful characterization of a real-time musical context. A good model for "context" will most likely involve a rather complicated data structure. All of these notions of extended musical context lead, in a computer implementation, to an extended grammar representation, *e.g.*, production rules augmented by embedded procedures. In any case, the notion of a grammar as simply a list of production rules is inadequate for music.

Grammars may lead beyond unified composing and analysis models and toward intelligent musical devices. An intelligent musical device will be able to convert the iconic musical signal into symbolic form, and be able to recognize for example, not only frequency, amplitude, and duration (as analog devices do today) but also larger syntactic forms such as phrases and other macrostructures as well as extra-syntactical aspects of the music. Acting from a base of programmed or even acquired grammatical knowledge, such a device will be able to listen and respond intelligently not just to sound, but to music.

Preceding all of this recent emphasis on grammars, music has long been the subject of a variety of formalizations. Ultimately, the question is not whether music conforms to the structure of formal grammars, but rather whether particular formal grammars can be designed which are useful representations of certain compositions. Grammars with embedded procedures can be powerful descriptive and expressive tools, but certainly formal languages will evolve, and in general, knowledge representations will grow more elegant.

Acknowledgements

Thanks to Curtis Abbott for his critical reading and comments during the preparation of this paper.

Bibliography

(1) Aho and Ullman (1972) *The Theory of Parsing, Translating, and Compiling,* Volume 1, Prentice-Hall, Englewood Cliffs.

(2) Buxton (1977) "A Composer's Introduction to Computer Music," *Interface,* Vol. 6, Swets, Lisse, pp. 57-72.

(3) Buxton, Reeves, Baecker, Mezei (1978) "The Use of Hierarchy and Instance in a Data Structure for Computer Music," *Computer Music Journal,* Vol. 2, No. 4, PCC, Menlo Park, pp. 10-21.

(4) Chomsky (1957) *Syntactic Structures,* Mouton, The Hague.

(5) Chomsky (1963a) "Formal Properties of Grammars" in *Handbook of Mathematical Psychology,* Vol. 2, Wiley, New York.

(6) Chomsky (1963b) "Formal Analysis of Natural Language" in reference (5) above.

(7) Chomsky (1965) *Aspects of the Theory of Syntax,* MIT Press, Cambridge.

(8) Chomsky (1979) 1979 Immanual Kant Lectures in Philosophy, Stanford University.

(9) Forte (1959) "Schenker's Conception of Music Structure," in *Readings in Schenker Analysis and Other Approaches* (1977) Yeston (Ed.) Yale, New Haven.

(10) Greimas (1970) *Du Sens,* Seuil, Paris.

(11) Jakobson (1970) *Main Trends in the Science of Language,* Harper and Row, New York.

(12) Jakobson (1973) *Questions de Poétique,* Seuil, Paris.

(13) Knuth (1971) "Top-Down Syntax Analysis," *Acta Informatica,* Vol. 1, No. 1, Springer-Verlag, Berlin, pp. 79-110.

(14) Laske (1972) "On Musical Strategies with a View to a Generative Theory of Music," *Interface,* Vol. 1, Swets, Lisse, pp. 111-125.

(15) Laske (1974) "Musical Acoustics (Sonology)—A Questionable Science Reconsidered," *Numus-West,* 6-74, Mercer Island, pp. 35-40.

(16) Laske (1975) "Introduction to a Generative Theory of Music," Sonological Reports, No. 1b, Institute of Sonology, Utrecht.

(17) Laske (1977) *Music, Memory, and Thought,* University Microfilms, Ann Arbor.

(18) Laske (1978) "Considering Human Memory in Designing User Interfaces for Computer Music," *Computer Music Journal,* Vol. 2, No. 4, PCC, Menlo Park, pp. 39-45.

(19) Lerdahl and Jackendoff (1977) "Toward a Formal Theory of Tonal Music," *Journal of Music Theory,* Vol. 21, No. 1, Yale, New Haven, pp. 111-172.

(20) Lindblom and Sundberg (1970) "Towards and Generative Theory of Melody," *Swedish Journal of Musicology,* Vol. 52, pp. 77-88.

(21) Lyons (1977) *Semantics,* Vol. 1, Cambridge U. P., Cambridge.

(22) Meehan (1978) "An Artificial Intelligence Approach to Tonal Music Theory," Author's manuscript.

(23) Molino (1975) "Fait Musicale et Sémiologie de la Musique," *Musique en Jeu,* No. 17, Seuil, Paris, pp. 37-62.

(24) Moorer (1972) "Music and Computer Composition," *Communications of the ACM,* Vol. 15, No. 2, pp. 104-113.

(25) Moorer (1975) "On the Segmentation and Analysis of Continuous Music Sound by Digital Computer," Department of Music Report No. STAN-M-3, Stanford University.

(26) Nattiez (1975) *Fondements d'une Sémiologie de la Musique,* Union Generale d'Editions, Paris.

(27) Nattiez (1977) "Le Dilemme d'IRCAME," *Musique en Jeu,* No. 27, Seuil, Paris, pp. 118-120.

(28) Roads (1977) "Preliminary Report on the CME Composing Language Project," Center for Music Experiment, UCSD, La Jolla.

(29) Roads (1978) "Composing Grammars," revised, Presented to the 1977 International Computer Music Conference, UCSD, La Jolla.

(30) Rowe (1978) "Grammar as a Programming Language," *Creative Computing,* Jan/Feb, pp. 80-86.

(31) Ruwet (1972) *Language, Musique, Poésie,* Seuil, Paris.

(32) Ruwet (1975) "Theorie et Methodes dans les Etudes Musicales," *Musique en Jeu,* No. 17, Seuil, Paris, pp. 11-36.

(33) Salomaa (1973) *Formal Languages,* Academic, New York.

(34) Schaeffer (1966) (New edition 1977) *Traité des Objets Musicaux,* Seuil, Paris.

(35) Sebeok (1975) "Six Species of Signs: Some Propositions and Strictures," *Semiotica,* Vol. 13, No. 3, Mouton, The Hague, pp. 233-260.

(36) Simon (1969) *Sciences of the Artificial,* MIT Press, Cambridge.

(37) Smith (1972) *SCORE Manual,* computer printout, Center for Computer Research in Music and Acoustics, Stanford University.

(38) Smith (1973) "SCORE–A Musician's approach to Computer Music," *Numus-West,* 4-73, Mercer Island, pp. 21-29.

(39) Smoliar (1974) "Process Structuring and Music Theory," *Journal of Music Theory,* Vol. 18, Yale, New Haven, pp. 308-336.

(40) Smoliar (1976) "Music Programs: An Approach to Music Theory Through Computational Linguistics," *Journal of Music Theory,* Vol. 20, Spring, Yale, New Haven, pp. 105-131.

(41) Smoliar (1977) "SCHENKER: A Computer Aid for Analysing Tonal Music," *SIGLASH Newsletter,* Vol. 10, No. 1-2, ACM, pp. 30-61.

(42) Ulrich (1977) "The Analysis and Synthesis of Jazz by Computer," *Proceedings of the 5th International Joint Conference on Artificial Intelligence,* Vol. 2, Dept. of Computer Science, Carnegie-Mellon University, Pittsburgh, pp. 865-872.

(43) Winograd (1968) "Linguistics and Computer Analysis of Tonal Harmony," *Journal of Music Theory,* Vol. 12, Spring, Yale, New Haven, pp. 2-49.

(44) Winograd (1973) "A Procedural Model of Language Understanding," in *Computer Models of Thought and Language,* Shank, Colby (Eds.) W. H. Freeman, San Francisco.

(45) Winograd (1977) "Five Lectures on Artificial Intelligence," in *Linguistic Structures Processing,* Zampoli, (Ed.), North-Holland, Amsterdam.