F. Pachet, (1997) Computer Analysis of Jazz Chord Sequences: Is Solar a Blues? in Readings in Music and Artificial Intelligence, Miranda, E. Ed, Harwood Academic Publishers, February 2000

Computer Analysis of Jazz Chord Sequences: Is *Solar* a Blues ?

François Pachet
SONY CSL-Paris
6, rue Amyot, 75005 Paris
pachet@csl.sony.fr

Abstract:
This chapter investigates the issue of the role of the computer in musical analysis. Starting with a survey of the main approaches in computer analysis, we focus on the particular problem of Jazz chord sequences harmonic analysis. We propose a theory of chord sequence analysis, based on an explicit conceptual hierarchy of analysis objects. We discuss the implementation of the theory and its results on a typical example (*Blues for Alice*, by Charlie Parker), for which the system produces an analysis which conforms exactly to human interpretation. We also exhibit a chord sequence, *Solar* (by Miles Davis), for which the results of the system do not conform to human perception, i.e. it does not find it is a Blues. We conclude on the issue of the role for the computer in musical analysis.

Key words: harmonic analysis, Jazz, Blues, computer analysis, production rules.

1. Introduction

There is a fascinating force governing musical analytical processes: the pleasure of possession. Since musical analysis became a pursuit in its own right, i.e. approximately the end of the XIXth century, various analytical methods have been devised and, to a large extent, formalized: formal, chordal, functional, Schenkerian, see e.g. (Bent & Drabkin, 1987) or (Cook, 1987). These methods differ in the nature of the musical material under study, and in the form of their output, but they are similar in their goal and operating mode: they consist in chopping the musical material into pieces, comparing these pieces and classifying them, in order to eventually reformulate the original material with the terms of an established corpus of concepts. Through such a reconstruction, a successful analysis may eventually provide a sense of possession, an intimate feeling of appropriation of the analyzed material which is comparable to the feeling the composer has for his own creation. Regardless of the prominent place analysis holds in musical aesthetics and compositional theory, analysis, seen as a way of understanding music by reformulation, is an enjoyable process. This pleasure of analysis certainly accounts for a large part in the leading role of analysis in musical studies.

The use of the computer as a partner in musical analysis is as old as the computer itself. There has undoubtedly been a dream behind the use of computers in musical analysis: the dream of entirely automating the analytical task, to analyze quickly large corpuses of musical material. However, the result is not clearly in favor of the computer. In 1980, Bo Alphonce argued that musical theory is not mature enough to be used as a basis for computer programs (Alphonce, 1980). This pessimistic statement is somewhat confirmed by the absence of analysis programs in the *Humdrum* general-purpose software tool developed at University of Waterloo (Huron, 1994). Humdrum is one of the most ambitious attempts so far at providing computational power to musicians in order to perform complex analysis of musical pieces. A laconic sentence may be found in the introductory documents: "Programs to do automatic functional analysis are not sufficiently reliable to be used in music scholarship".

However, lots of work have been done in the field of computer analysis and these works did produce interesting and insightful results, if not fulfilling the dream of an autonomous, complete and automatic analyzer. Indeed, the computer has assumed virtually any possible role in the analytical game: from a simple passive tool to a simulation of an intelligent analyzer agent. Artificial intelligence techniques in particular have been used extensively to provide various kinds of frameworks in which analytical processes may be investigated, studied, and highlighted in various ways.

This chapter investigates the issue of the role of the computer in musical analysis. Starting with a survey of the main approaches in computer analysis, we focus on the particular analytical problem of Jazz chord sequences. We exhibit a model that allows to analyze automatically such sequences. We discuss its implementation and results on a typical example (*Blues for Alice*, composed by Charlie Parker), for which the system produces an analysis which conforms satisfactorily to human interpretation. We also exhibit a chord sequence, *Solar* (composed by Miles Davis), for which the results of the system do not conform to human perception - it does not find it is a Blues. We conclude on an emerging role for the computer in musical analysis.

2. Survey of Computer Analysis

Artificial Intelligence has traditionally been split into two categories of pursuits: developing techniques to produce sophisticated artifacts, or building computational models of human cognition. These two categories correspond to two main roles that have been assigned to the computer in musical analysis: 1) the computer is used as tool for musicologists, and 2) the computer is used as a simulator of a model devised by a cognitician to account for the analytical process. A third category of experiments is when the computer itself is the subject of study, and musical analysis becomes an experimental framework in which knowledge representation techniques are evaluated, and musical theories are empirically validated.

2.1. The computer as a tool

One of the most straightforward uses of the computer for musical analysis is software designed to perform various "low-level" tasks in order to help musicologists in their routine work. These tasks are typically computations of frequency distribution for pitch classes or intervals in melodic lines. Various forms of such statistical techniques were used in the 70s (Lincoln, 1970) to perform style analysis. These techniques are appealing because they apply to all kinds of musical corpuses, including ethnical music. The more recent work by (Mason, 1985) is in the same vein, using vectorial analysis and complex number notations and representation. However, the abstract musical concepts which are necessary for fully-fledged analysis (e.g. *cadences*), are out of reach of purely numerical techniques, thus limit the scope of these systems (see e.g. a critical comment by (Rothgeb, 1971)).

In reaction to this numerically oriented trend of work, research was conducted using some symbolic representation of music. Several software packages were developed to provide computer-aided analysis of tonal music (Kassler, 1975), (Byrd, 1977), (Brinkman, 1980) i.e. sets of programs to be used by a human that perform limited, well defined analytical actions, but that do not propose a complete analysis. In the same spirit, (Smoliar, 1980) developed tools for computer-aided Schenkerian analysis. These tools were designed with the goal of understanding the intricacies of the theory - in this respect, they could also be classified in the third category, following section - but concluded with the impossibility of building a totally automated analyzer (Frankel et al., 1978). More recent works by (Camilleri et al., 1987) attempted to combine the brute force of numerical analysis with smarter types of hierarchical parsing in a single software package, also to be controlled by human (students or musicologists). The work of Pierre-Yves Rolland (see his chapter in this issue) on pattern induction in Jazz corpuses may be seen as a continuation of this trend, using more advanced numerical techniques (dynamic programming), together with a rich representation of musical objects (the *MusES* system, (Pachet, 1994a)).

The work mentioned above usually applies to tonal music. As far as post-tonal music is concerned, let us mention two kinds of efforts in opposite directions. The theoretical work of Chemillier showed that serial music is, in some sense, rational (Chemillier, 1990), but this result did not lead to actual implementations of serial music analyzers. On the other hand pitch-set theory was primarily designed with implementation constraints in mind. A tremendous amount of work has been devoted to building computer implementation of pitch-set theory, e.g. (Forte, 1973a; Forte, 1973b), (Rahn, 1980a), (Harris & Brinkman, 1986). The most recent attempt to provide a complete set of tools is the *Contemporary Music Package* by (Castine, 1994) which acquired the status of a real working system, as opposed to the myriad of prototypes designed so far. Finally, the case of electronic music is special, since no stable analytic theory is available, which poses new problems as yet unaddressed (Stroppa, 1984).

A more elaborate use of the computer as a tool for analysis is to validate specific, user defined theories of a musical piece. Here the computer is used as a simulator of a model, carrying its own semantics, rather than as a simple neutral tool. This is typically the case of the *Morphoscope* project, in which the computer is used to rebuild a score (Mesnage, 1993), with numerous and convincing applications (Mesnage & Riotte, 1990), (Rokita, 1996). An alternative and interesting work in the same spirit is the reconstruction of a fragment of Ligeti's *Melodien* by (Chemillier, 1995), using a model specifically designed for the material studied. The model is implemented with the *Patchwork* system (Laurson & Duthen, 1989). Through an explicit reconstruction of the entire score, these studies emphasize the idea that analysis and composition have strong, organic relations (Mesnage, 1995).

2.2. The computer as a simulator of the analytic process

The second main role of Artificial Intelligence is to model human cognition. This is an ambitious goal since such models in principle, require experiments in cognitive psychology to be validated. Musical analysis is then seen as a particular form of a general, typically human ability. Using the computer as a simulator of such a model is one of the goals of Minsky's model of K-lines (Minsky, 1986), which has been applied to the simulation of analytical processes in music (Minsky, 1985), including Jazz (Horowitz, 1995). In the same spirit, Greussay's Beethoven graphs (Greussay, 1973; Greussay, 1985) is a model for analyzing Beethoven's *Diabelli variations*. The idea here is not so much to analyze the material itself, but rather to validate a conception of the analytical process, seen as a kind of cooperation between various autonomous agents. The models in this category are designed to be computationally tractable, and therefore may also have practical applications, especially in the context of performance-oriented analysis. The *Cypher* system for instance (Rowe, 1993), contains a real-time analyzer that draws on Minsky's model, in which agents essentially operate frequency computations (see chapter by Rowe in this issue).

More generally, the segmentation of musical data was identified as a general issue for cognitive sciences. Grouping and segmentation was addressed by Baker (Baker, 1989a; Baker, 1989b), who compared two techniques applied to the same segmentation problem, and (Camilleri et al., 1990), who use an expert system approach on the same issue. Quite appealing computational models were developed to find boundaries in atonal music, e.g. Forte (Forte, 1973a) using pattern recognition procedures, or Polansky's hierarchical analysis inspired by the gestalt theory (Polansky, 1979), (Tenney & Polansky, 1980). Other interesting models were developed with the goal of modeling the perception of the human analyst but without direct corresponding implementations, e.g. (Hasty, 1978) or (Chouvel, 1990).

2.3. The computer as the subject-matter

The third category of studies corresponds to a radical shift in focus, compared to the two previous ones: the computer becomes the actual subject-matter, instead of being used as a passive tool or simulator. Moreover, music analysis is considered as a rich experimental field, a source of *well posed* problems. The main goal is not so much to study musical corpuses or theories as such, but rather to study how certain types of knowledge can be put into a computer, and hence belong to the field of knowledge representation at large.

Almost all AI techniques have been applied to some kind of analytical problem. Procedural approaches were used by (Ulrich, 1977), to analyze Jazz chord sequences, and various algorithms were developed to realize specific forms of analysis (Brinkman, 1986; Brinkman, 1990), (Elliott, 1987), (Mouton, 1995). The work of (Winograd, 1968) for analyzing musical scores using systemic grammars paved the way for a whole generation of studies on grammars. Grammar-based approaches became very popular in the 70s and were applied to virtually all available musical corpuses (see e.g. (Baroni, 1984) and a review on the use of grammars for musical analysis in (Roads, 1988)), culminating in the reference work of (Lerdahl & Jackendoff, 1983). More classical AI mechanisms such as *production rules* were used to develop analytical programs of all sorts: (Bein & Winold, 1983), (Maxwell, 1984; Maxwell, 1992). Languages from the area of *logic programming* were also put to work. For instance, Schenkerian analysis is performed as one of the viewpoint of Ebcioglu’s choral generator (Ebcioglu, 1992). His system is based on a particular logic programming language (BSL) developed specially for the purpose of the application, and uses heuristics to control bottom-up parsers (Ebcioglu, 1987). The formalism of *conceptual dependency* (Schank, 1973) was used by (Meehan, 1980) to implement the implication/realization theory of Narmour. Representation of general-purpose musical structures for analysis and composition were developed by (Smaill & Wiggins, 1990), who use an adaptation of Ruwet’s analytic theory (Ruwet, 1972) to analyze pieces of Debussy’s *Syrinx*.

This third category is large: it contains most experimental work in AI and Music. It is also the subject of much litigation, as its objectives are not always clearly stated. Indeed, these approaches have all in common a tendency to turn into an "exercice de style", with a limited impact: applying a sophisticated technique to a complex problem may not show much more than the virtuosity of the technician. As (Smoliar, 1992) convincingly argues: "each technological advance becomes a new temptation to put the (musical) data into the computer again", and strong critiques were issued against this trend of work (see e.g. (Rahn, 1980b)).

However, there are often misconceptions about the objectives of these experiments. Although a computer implementation *per se* may not prove anything on the music model from which it is inspired, it may nevertheless provide two kinds of benefits. First, these achievements may actually be used to validate general knowledge representations technique, in a complex domain. In this context, music theories are interesting for AI only to the extent that they are computationally tractable. These issues are not purely technical, though. They embody an experimental view of Artificial Intelligence, in which the study of music is seen from a broad perspective of knowledge engineering, and where the goal is to elicit knowledge with practical objectives in mind, thereby raising important issues in knowledge representation.

Second, even if the epistemological status of theories in analysis is not clear (after all, what is a theory without theorems ?), there is a growing community of analysts who look for musical theories having predictive capacities, to the point of becoming falsifiable (Kunst, 1987). In this regard, work in our third category may reveal the incompleteness of a theory, as it was for instance the case with the study of (Rothgeb, 1968) on unfigured bass, which revealed inconsistencies and omissions in official treatises of harmony.

We will illustrate these two issues in the next section, on a simple and practical analytical problem, and analyze its strengths and limitations.

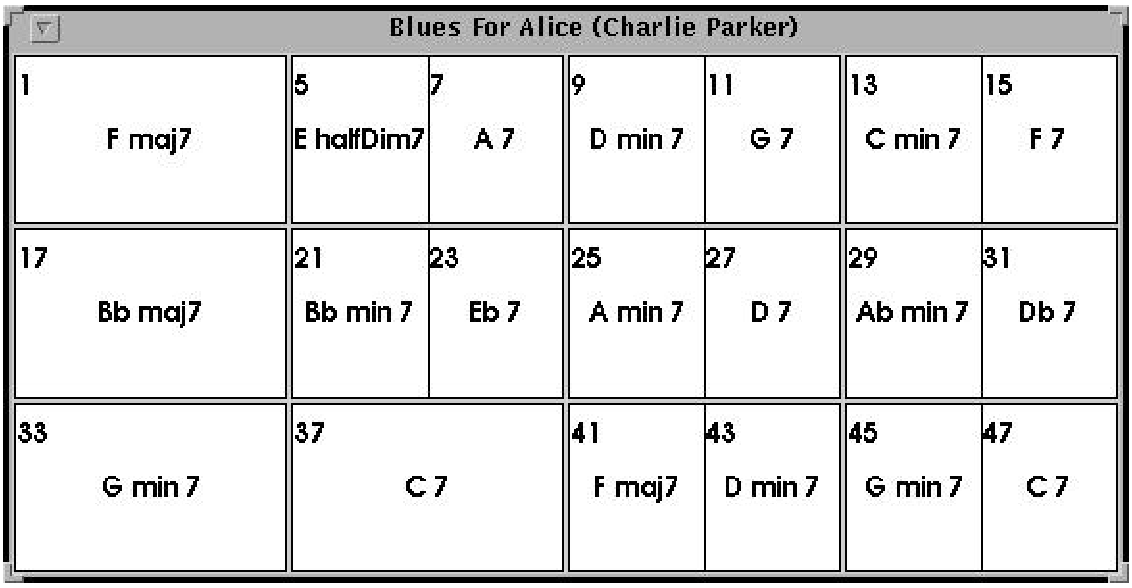
3. Analysis of Jazz Chord Sequences

The problem of harmonic analysis of Jazz chord sequences perfectly illustrates the third role of the computer in musical analysis. It is a well posed problem, of relative simplicity, and it may be used to illustrate our two arguments stated above. We will first state the problem in musical terms, review three approaches on the problem, and discuss their limitations. We will then propose an analytical model and its implementation using Artificial Intelligence techniques. The model will be illustrated on two examples, one successful (*Blues for Alice*), and one problematic (*Solar*). Finally, we will discuss two issues raised by experimenting with such a system; one issue concerning musicology, one concerning Artificial Intelligence. We will finally conclude on the role of Artificial Intelligence in computer analysis.

3.1. Problem statement

Like classical harmony, tonal Jazz harmony is a well studied domain, as one can see by the large number of books on this subject. This profusion of literature is directly related to the size of the Jazz musical corpus, typically illustrated by books such as (Fake, 1983; NewReal, 1991; Real, 1981). Such corpuses contain approximately 2000 Jazz chord sequences, referred to by experts as "standards". Most of these tunes were composed by Jazz musicians in the 50s (the be-bop period, including Charlie Parker, Dizzie Gillespie, Miles Davis) and later (the so-called hard-bop period, and more recently and to some extent the Jazz-rock period).

We will state here the problem of Jazz chord analysis informally, and propose a more rigorous theory in the following sections. The problem consists in analyzing harmonic functions in chords in Jazz chord sequences, i.e. finding, for each chord in the sequence, its underlying tonality and its function in this tonality. The input of the problem is a chord sequence as found in the literature (see example Figure 1). The output is an annotation of the sequence, in which each chord is labeled with its harmonic function. The harmonic function is usually represented by a degree (a number, written as a Roman numeral) and a tonality, itself consisting in a root (a pitch class) and a scale type, e.g. I of Cb major, IV of G minor, etc.



Blues For Alice (Charlie Parker)

1 F maj7	5 E halfDim7 7 A 7	9 D min 7 11 G 7	13 C min 7 15 F 7
17 Bb maj7	21 Bb min 7 23 Eb 7	25 A min 7 27 D 7	29 Ab min 7 31 Db 7
33 G min 7	37 C 7	41 F maj7 43 D min 7	45 G min 7 47 C 7

Figure 1. Chord sequence *Blues for Alice* composed by Charlie Parker, as found in the Real Book (Real, 1981). Each square lasts four beats. Starting beats are indicated in small font on the top left of each chord.

There is a distinctive feature of Jazz chord sequence analysis, which makes it rather unique in the field of analytical studies in general: the fact that the output of the analysis has an unusually precise and practical goal. These analyses are typically performed prior to improvisation because they yield *valid scales* the soloist may use to build his chorus. Indeed, the starting point of improvisation is the fact that the harmonic function (e.g. IV of G minor) contains enough information to deduce the set of "valid" notes that can be used, as well as, for each of them, their relative importance (the notes of the tonality, starting from the given degree, and alternatively considered as strong and weak notes). We do not address here the problem of how precisely improvisation may be built from such an analysis, which has received much attention (see e.g. (Ulrich, 1977), (Fry, 1980), (Giomi & Ligabue, 1991), (Johnson-Laird, 1991), (Walker et al., 1992), (Baggi, 1992)). But in all cases, a good improvisation relies on a precise harmonic analysis of the chord sequence. Moreover, not only tonalities and harmonic functions are used, but also more abstract harmonic patterns, such as two-fives, or turnaround, which are directly associated with memorized "licks" from which the improviser builds his melodic lines (see, e.g. the model of improvisation developed by (Ramalho & Ganascia, 1994), based on the association of such licks with such harmonic patterns using a case-based representation of musical memory). Because these patterns play a central role in the improvisation, their recognition in Jazz chord sequence is one of the main duties of analysis.

This practical aspect of analysis has an important consequence concerning its hierarchical nature: a tune may be globally in C major, but some parts may be in F (modulation), and these parts may themselves contain sub parts that modulate, and so forth. Firstly, the level of embedded modulations may be high in a typical Jazz chord sequence, as opposed to the usual levels of embedded modulation found in e.g. Baroque music. This situation arises mainly because of chord substitutions, which are used systematically and recursively. Second, contrarily to classical music in which only the surface level of embedded modulations are usually interesting, in Jazz all levels may potentially be relevant, because the improvisation will build musical phrases of varying size, and thus will be based on tonalities of varying depth, as was shown experimentally by (Järinen, 1995). A chord therefore is not really analyzed in one single tonality, and several tonalities may be considered, depending on the depth of the analysis considered. In example of Figure 2, the first chord (A7) may be considered in the tonality of D, from a short-sighted viewpoint, or G, from a higher viewpoint, or, even here, C. In each case different scales would be used for improvisation, and the choice of the tonality depends on the nature of the musical improvisation being conducted. Note that we consider the problem of choosing a "best" tonality for chords to the responsibility of the improviser, and not to the responsibility of our analyzer.

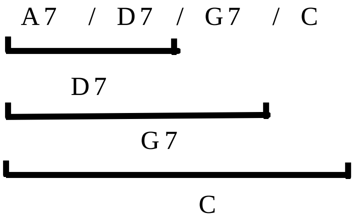


Figure 2. The hierarchical nature of the analysis of a group of chords.

We will now review three typical attempts in building systems that perform such a chordal analysis task automatically.

3.2. Previous work on automatic chordal analysis

We review briefly three exemplar systems that propose a solution to the chord sequence analysis, emphasizing their main advantages and drawbacks. The first two are operational systems, the third one is a model with no implementation.

3.2.1. Maxwell's system

Maxwell's system (Maxwell, 1984; Maxwell, 1992) produces chord function analysis of Baroque pieces using a knowledge-based, expert system approach. Based on the preceding work of (Bein & Winold, 1983) in the automatic analysis of Bach chorales, Maxwell considers two main tasks for chord function analysis:
1 - Determine which vertical sonorities constitute a chord worthy of a function label (this problem is irrelevant in our case since chords are already explicitly mentioned on the score).
2 - Determine the tonal regions in which chords should be analyzed.

The chord analyzer produces an annotation of the score similar to a human annotation, i.e. consisting in chord functions for each chord of the sequence. These functions are roman numeral labels together with tonalities. The output of the analysis is therefore flat, which is justified by the nature of the corpus being analyzed (Baroque music), in which the hierarchical aspect of analysis is not of central importance. One of the main goal of the system is to minimize modulations. The reasoning consists mainly in detecting cadences, and then interpreting the rest of the chords according to these cadences. The system is implemented by a set of *production rules* associated with numerical priorities, and handled by a prioritized agenda to control conflict resolution. It proceeds from left-to-right, and decides to modulate only when some numeric threshold of functional weakness is exceeded. The most obvious critique that can be issued is the intensive use of numerical values in rules. For example, rule 43 states: "if the root motion from pre-cadence chord to goal chord of a p-cadence is a descending perfect fifth AND the pre-cadence chord contains a major third above the root, THEN the p-cadence is authentic, and its strength should be increased by 20". The drawbacks of using numerical values in rules has now long been established: they are difficult to justify, difficult to maintain, and have poor explanatory capacity. Also, the proposed approach has a strong procedural component and the left-to-right scheme is counter-intuitive. To reduce that rigidity, rules such as rule 41 are introduced (and used extensively): "If the chord cannot be analyzed in the key of the previous chord OR the analysis is "doing badly" AND enough is known about future keys to look for a better key THEN determine which key will provide the strongest analysis of the new few chords or measures, modulate to that key, analyze the present chord in the new key, and assign pivot functions to some previous chords". However, Maxwell's system produces interesting and occasionally insightful results for musicologists.

3.2.2. Ulrich's system

The system described by (Ulrich, 1977) solves the harmonic analysis problem for simple Jazz chord sequences, as a part of a general system for building Jazz improvisation. Similarly to Maxwell's system, Ulrich's system produces a flat, one-to-one labeling of chords, in which each chord is assigned to a key enter (a tonality) and a function, such as "tonic", "subdominant", or "transition". As in Maxwell's system, the analysis system tries to minimize the modulations, i.e. changes in tonality, but uses a symbolic rather than a numeric tactic. Its implementation is also based on a left-to-right algorithm that parses the sequence. Chords are progressively "eaten" when they can be analyzed in the tonality of the already analyzed sequence. The system has an appealing organic quality: it is based on an "island-growing" mechanism in which isolated groups of chords (islands) try to grow as much as possible to encompass adjacent chords. Ulrich concludes his presentation by noting that "Jazz encodes so much harmony in the local structure of the music that global considerations can be ignored". Indeed, his system may be used to analyze simple chord sequences (*How high the moon*, by Morgan Lewis, not a terribly difficult one), and Ulrich proposes a simple algorithm for building improvisations as juxtapositions of motifs, adapted to the harmonic functions found by the analysis system. Expectedly, the quality of the system's improvisation is poor (as stated by the author himself). One reason is that the analysis system provides only a small part of the information required to build a more elaborate melody. It does not detect the specific harmonic patterns of Jazz harmony, and cannot detect embedded modulations. It is therefore unable to provide a hierarchical, semantically rich view of the global tonality.

3.2.3. Steedman's model

The model proposed by (Steedman, 1984) aims at describing *12-bar Blues*, a particular, and important, subset of Jazz chord sequences. One important characteristic of the model is that it explicitly takes into account the hierarchical nature of Jazz chord sequences. The model is particularly representative of the trend of grammar-based research: it only describes the corpus, and is not intended to be directly implemented.

Steedman identifies a set of 6 transformation rules that model the 12-bar Blues, by applying, recursively, various transformations to an initial, simple chord sequence, considered as an "essential" Blues. These rules are showed to be sufficient to generate a large set of complex 12-bar Blues tunes. The output of Steedman's model (produced manually) is a derivation tree that displays all the transformation rules to apply to the initial axiom of the Blues to reconstruct the given input chord sequence.

A spectacular aspect of this work is the fact that a small set of rules captures a large amount of possible variations from the original 12-bar template. Steedman's model is, in a way, validated by the existence of a large quantity of 12-bar Blues, that all fit nicely in this formalism, i.e. can all be generated by his set of rules. It is difficult indeed to resist the appeal of this model, and not see it as a sort of "Maxwell equations" (no relation with the author mentioned in the preceding section) or "abstract truth" of the Blues, as quoted by Steedman from a song by O. Nelson (Nelson, 1961).

However this model suffers a number of drawbacks. First, the model is not implementable. This comes essentially from the presence of so-called *context-dependent rules* of the model. Theoretical work on grammar parsing concluded that context free grammars can be parsed using automatic tools, but context dependent grammars are much more problematic (see e.g. (Roads, 1988)). For instance, rule (4) is as follows: $D_X7\ x(7) \rightarrow b\mathrm{St}_X(7)\ x\ (7)$. This rule states that any seventh chord that resolves (the x(7)) can be substituted by the seventh chord of the tritone ($b\mathrm{St}_X$ stands for "flattened supertonic of x"). This rule is inherently context-dependent because the application of the substitution may occur only within a particular context (x(7)). Other rules are even more problematic from a computational point of view. Rule number 3 writes as follows: $w\ \ x7$ ® $D_X7\ \ xm7$, where *w* may match any chord. However, to avoid problems of infinite loops, it is necessary to impose the constraint that *w* should not match a chord that has had its root changed by the previous application of a substitution rule!

The second problem is that the model only accounts for "well-formed" sequences, and therefore is very sensible to perturbations, idiomatic progressions and other harmonic "mistakes". Similarly, while the model captures some 12-bar Blues, there are a lot of other "initial" chord sequences from which Jazz chord sequences can be derived. Moreover, interesting Jazz chord sequences often do not derive from a particular, known, initial chord sequence (e.g. *Nardis* by Miles Davis). Finally, the model produces an output that is only a small part of the analysis. When the tune is analyzed as a Blues, the model provides a derivation tree from which is not clear how to find the underlying tonalities (e.g. to be used for improvising). When the tune is not analyzed as a Blues, the model simply answers "no", even if portions of the tune could have been correctly analyzed. Note, however that such a grammar can be used without problem in a generative mode, to produce variations of existing chord sequences, as proposed e.g. by (Johnson-Laird, 1991). In short, the proposed model accounts convincingly for the recursive nature of bluesy substitutions in chord sequences, but does only a small part of the analytical job.

We will describe now our solution to the problem of automatic Jazz chord sequence analysis. Conversely to the systems presented here, our system produces a hierarchical, complete analysis of Jazz chord sequences that is robust to harmonic mistakes. It uses an entirely symbolic approach (no numeric values), and can be seen as an extension of Ulrich's island growing mechanism. Finally, its output may directly interpreted to produce tonalities to be used for improvisation. We will first emphasize the role of analysis objects in the reasoning, and propose an ontology for these objects. Then we will propose a model for the reasoning process based on this ontology. Finally we outline the implementation of the theory using AI techniques and discuss its results.

3.3. Analysis objects

As we said in the introduction of this chapter, the very act of analysis consists in recreating the chord sequence. This re-creation involves the reconstitution of a imaginary process of composition that produced the sequence. However, in the preceding approaches to the chord analysis problem, this reconstitution is not made explicit, since there is no real language for expressing it. Ulrich's and Maxwells's systems use a scarce representation for the chord labeling ('dominant', 'subdominant', etc.). There is no representation of the abstract entities that are manipulated by the analysis process, so there is no real reconstitution in this sense. Steedman's model produces a tree of rule derivations which can be seen as a kind of reconstitution, since it expands an initial, essential axiom of the Blues down to the complete chord sequence. But this reconstitution is effected in a abstract, uniform world, in which the only reconstituted action is the anonymous application of a generative rule (in technical terms, the tree is not an abstract syntax tree, i.e. nodes in the tree are not labeled with harmonic functions.)

In the particular context of Jazz, the output of the chord analysis plays, as we have seen, a most important role. Not only is the analysis inherently hierarchical, to account for the many tonal levels, but its expression must coincide with the patterns identified by Jazz musicians, on which they can build their melodic lines. The flat outputs, as well as the semantically neutral derivation tree are not adapted to this aim. We claim that a relevant representation of the analysis should, in this case, be a hierarchical structure (a tree), whose nodes denote meaningful analytical abstractions.
These abstractions can be classified into two main categories: *basic building blocks* that represent partial chord sequences, and *harmonic operations*, that manipulate these building blocks to produce chord sequences. The main idea of our approach is to explicitly represent both building blocks and harmonic operations, by turning them into abstract concepts with which the analysis, *in fine*, will be formulated. More precisely, we propose to classify all the objects making up the analysis under a common concept: the *shape*. A shape is a temporal object, describing a collection of chords. The basic building blocks as well as the reified harmonic operations are represented as special kinds of shapes.

The basic building blocks are simply the chords themselves, considered as atomic entities, as well as a small number of fixed idioms belonging to the corpus under study. These idioms are typical, cliché, sequences of chords which bear harmonic meaning in themselves, such as *turnarounds*, *two-fives*, or *two-five-ones*. At a higher level, global macro forms such as *Blues* or *AABA* are also considered as high-level idioms, to which we add, for the sake of completeness, shapes describing chord sequences with less prominent structure (*MonoTonalShape*, *BiTonalShape*, etc.).

The harmonic operations consist in producing chord sequences by combining or modifying existing chord sequences, according to some combination rules. Typical harmonic operations are *chord substitutions*, whereby a chord in a certain context is substituted by another chord. More general forms of harmonic operations include, for instance, *ExtendedShape*, i.e. shapes obtained by extending a shape with a chord analysable in the tonality of the shape. Particular cases of shapes such as *ModalBorrowing* are also classified in this category (see below). This classification results in the hierarchy of concepts illustrated in Figure 4 (the inheritance relation represents a generalization relation between types of shape).

An important characteristic of shapes is that they manifest themselves through well identified patterns of chords, or of other shapes. To each shape in the catalog of shapes corresponds such a pattern. These patterns can be found in most books on Jazz harmony (Coker, 1964) (Beaudoin, 1990). For instance, a turnaround shape such as (C maj7 / Eb7 / Dmin7 / Db7) is perfectly identified by a set of harmonic relations between adjacent chords. Here the relations could be abstracted by: (a chord of root X, a chord of root (X + minor third) seventh, a chord of root (X + second minor) seventh, a chord of root (X + diminished second 7). Similarly, a *Blues* shape may be described as three adjacent shapes covering the whole sequence, such that the first and last one are in the same tonality, and the middle one is analyzed in the fourth of the others.

More complex shapes such as "modal borrowing" follow similar patterns. A modal borrowing is a local modulation which may be considered as non significant, when it comes in between two shapes analyzable in the same tonality, and when this local perturbation may be analyzed in the relative minor tonality of the adjacent shapes. This pattern is illustrated in Figure 3.

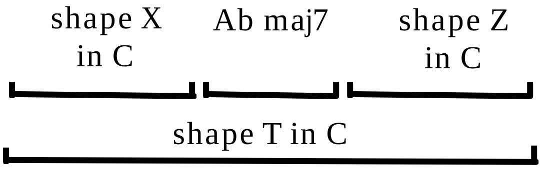


Figure 3. Modal borrowing configuration. Here, the local perturbation is a Ab major chord. Ab major may be analyzed in C minor (VIth degree) and therefore be considered as a modal borrowing in C major.

```
TemporalObject
  Shape
     BuildingBlocks
          IsolatedChord
          Idioms
               TurnAround1
               TurnAround2
               TwoFive
               TwoFiveOne
               ResolvingSeventh
          GlobalShape
               BluesShape
               AABAShape
               ABABShape
               MonoTonalShape
               BiTonalShape
               TriTonalShape
               QuadriTonalShape
               PentaTonalShape
     ReifiedHarmonicOperation
```

- ChordSubstitution
    - TritoneSubstitution
    - BluesySubstitution
- GeneralHarmonicOperation
    - ComposedShape
    - ExtendedShape
    - ExtendedIntersectedShape
    - TransitionShape
- ModalBorrowing

Figure 4. The ontology of analysis objects. Indentation denotes inheritance.

3.4. A theory of Jazz chord sequence analysis

Based on our ontology of analysis objects, we propose to specify formally the problem around the three following points:

A) Basic principles
The theory is based on two basic principles: legality and minimization.

1) A legality principle

This principle says that each chord, out of any context, can be analyzed in a fixed set of possible tonalities. For instance, a given C major chord may be analyzed as: I degree of (the tonality of) C major, IV of G major, V of F major, VI of E harmonic minor, and so forth. This legal set may be simply computed, once for all, by extracting scale-tone chords from all possible scales (Pachet, 1994b).

2) A minimization principle

In the context of a sequence, the choice of the "good" tonality for a chord will of course depend on its metrical location, and on its relation with adjacent chords. The idea here is that the best tonality will be the one that minimizes modulations, i.e. the one that is common to the greatest number of adjacent chords. For instance, the chord sequence (C / F / E min / A min) has only one tonality that is common to all chords: C major.

B) Shape identification and analysis
As we saw, a central hypothesis in the analysis is that to each shape in the ontology of analysis objects corresponds a configuration of chords that perfectly identifies the shape. Moreover, recognised shapes have *specific tonalities*, which can be computed directly from their structure. For instance, a turnaround such as (C maj7 / Eb7 / Dmin7 / Db7) should always be analyzed in the tonality of its first chord, here C major. The main problem - and exciting part - of analysis comes from the fact that the analysis of a shape may violate the legality principle for some of its chords. The turnaround (C maj7 / Eb7 / Dmin7 / Db7) should be analyzed in C major, regardless of the fact that C major does not belong to the legal set of Db7 and Eb7. These chords *in abstracto* may not be analyzed in C major, but they can be within a given shape. In other words, idioms are configurations of chords that bear harmonic meanings *in themselves*.

C) Recursion
Finally, our analysis is recursive: any recognized shape may itself be considered as atomic for a higher level of analysis. This recursive nature accounts for the hierarchical nature of the analysis, and is of the utmost importance in Jazz as we argued in the Problem Statement section. For instance, resolving seventh chords may be considered as preparations, and therefore may be integrated in their resolving chord, as illustrated in Figure 2.

3.5. The output of the system and its interpretation

Let us put our theory of analysis to work on the sequence *Blues for Alice* of Figure 1. The output of our system is illustrated in Figure 5, in which the tree is to be interpreted as follows. Each line corresponds to a node in the analysis tree. The node is labeled by the lapse (starting and ending beats), and the name of the shape identified. Here, the chord sequence in itself is recognized as a Blues in F major, as indicated by the root of the tree (*1-48 BluesShape in F MajorScale*). The three subshapes making up the Blues are 1) *45-4 ResolvingSeventh in F MajorScale*, 2) 5*-20 ChordSubstitution in Bb MajorScale* and 3) *17-48 ExtendedShape in F MajorScale*. Each of these three shapes in turn is decomposed into various shapes, until the chords are reached. The explanation for the starting beat of the first shape (45-4) is that the identified shape starts at the end of the sequence (beat 25) and ends at the beginning (beat 4). More details on this aspect are given in the section on Circularity.

```
1-48 BluesShape in F MajorScale
    45-4 ResolvingSeventh in F MajorScale
      45-48 ChordSubstitution in F MajorScale
        45-48 TwoFive in F MajorScale
            [G min 7 ]
            [C 7 ]
      [F maj7 ]
    5-20 ChordSubstitution in Bb MajorScale
      5-20 ResolvingSeventh in Bb MajorScale
        5-16 ChordSubstitution in Bb MajorScale
            5-16 TwoFive in Bb MajorScale
               5-14 ChordSubstitution in C HarmonicMinor
                  5-14 ResolvingSeventh in C HarmonicMinor
                       5-12 ChordSubstitution in C MajorScale
                          5-12 TwoFive in C MajorScale
                             5-10 ChordSubstitution in D HarmonicMinor
                                 5-10 ResolvingSeventh in D HarmonicMinor
                                     5-8 ChordSubstitution in D HarmonicMinor
                                         5-8 TwoFive in D HarmonicMinor
                                            [E halfDim7 ]
                                            [A 7 ]
                                     [D min 7 ]
                             [G 7 ]
                       [C min 7 ]
               [F 7 ]
        [Bb maj7 ]
    17-48 ExtendedShape in F MajorScale
      17-46 ExtendedShape in F MajorScale
        17-44 ExtendedShape in F MajorScale
            [Bb maj7 ]
            21-44 ExtendedShape in F MajorScale
               21-42 ChordSubstitution in F MajorScale
                  21-42 ResolvingSeventh in F MajorScale
                       21-40 ChordSubstitution in C MajorScale
                          21-40 ResolvingSeventh in C MajorScale
                             21-32 ChordSubstitution in Db MajorScale
                                 21-32 ResolvingSeventh in Db MajorScale
                                     21-28 ChordSubstitution in D MajorScale
                                         21-28 ResolvingSeventh in D MajorScale
```

```
                                                  21-24 ChordSubstitution in Ab MajorScale
                                                        21-24 TwoFive in Ab MajorScale
                                                                 [Bb min 7 ]
                                                                 [Eb 7 ]
                                                  25-28 ChordSubstitution in G MajorScale
                                                        25-28 TwoFive in G MajorScale
                                                                 [A min 7 ]
                                                                 [D 7 ]
                                          29-32 ChordSubstitution in Gb MajorScale
                                               29-32 TwoFive in Gb MajorScale
                                                  [Ab min 7 ]
                                                  [Db 7 ]
                                 33-40 ChordSubstitution in F MajorScale
                                     33-40 TwoFive in F MajorScale
                                          [G min 7 ]
                                          [C 7 ]
                          [F maj7 ]
                [D min 7 ]
        [G min 7 ]
[C 7 ]
```

Figure 5. An analysis of the chord sequence of Figure 1. Leaves are between brackets and correspond to the chords of the sequence. Intermediary nodes correspond to recognized shapes (e.g. two-five, chord substitution), preceded by their respective start beat and last beat, and followed by their tonality.

Let us mention here again the fact that such a tree actually fills the need of the improviser, which is the main driving force behind this model. The tonality of a chord depends on the width of the musical phrase being played, as mentioned in C) of the preceding section. This tonality is given by the tonality of the corresponding segment in the chord sequence. Here the tonality of a segment is determined by choosing the smallest shape in the tree that contains the segment. For instance, the analysis of chord (Bb min 7) in measure 21, seen with a short segment of two bars, would be Ab major. Seen from a larger segment starting at beat 21 and ending at beat 28, it would be D major (shape *21-28 ChordSubstitution in D Major*). Considered from an even higher level, say beat 21-40 it would be C major. A very long segment from beat 21 to, say, 44 would consider the chord as being in F, and so forth. Four different tonalities may thus be considered for this single chord, depending on the span of the musical segment considered for improvising. The tree not only gives a variable tonality for a given segment, but also yields a corresponding shape type, which can then be associated to various musical idioms (see again the work of P.-Y. Rolland, in this issue, on the detection of such formulaic patterns). These two aspects make the output of the analysis system directly usable for improvisation.

3.6. Implementation of the theory

The reasoning process as described in our theory is represented by 1) an object-oriented representation of the analysis objects, 2) a representation of the reasoning by first-order forward-chaining production rules, and 3) a model of circular time.

The first aspect does not pose any problem. Object-oriented languages implement conceptual hierarchies by the mechanism of class inheritance, which is, in our case, perfectly adapted for representing the simple tree-like hierarchy of our musical concepts.

3.6.1. Aggregation and forgetting

The second aspect is more interesting, since the problem is to simulate the process by which the analysis tree is built. Nicholas Cook in his treatise (Cook, 1987), p. 16, says: "there are essentially two analytical acts: the act of omission and the act of relation". In our context, this metaphorical statement turns out to be unexpectedly operational. Inspired by the work on temporal reasoning of (Dojat & Sayettat, 1994), we used a model of analytical reasoning based on two main analytical actions: aggregation and forgetting. The reasoning process *per se* is represented by a set of rule bases which perform two kinds of tasks, by observing the initial chord sequence (in no particular order):

- a "pattern recognition" task in which shapes are built by aggregating configurations of already identified shapes,
- a "forgetting" task, in which irrelevant or redundant shapes created in the preceding task are destroyed.

Pattern recognition (or aggregation) rules all follow the same pattern: they consist in detecting configurations of adjacent shapes (the IF part, consisting in sequences of logical assertions separated by periods), and in building larger shapes by aggregation (the THEN part). The semantics of a rule is given by the types of shapes that are detected, the harmonic relations between them, and the type of shape that is created. For instance, Rule 1 below recognizes a two-five in major (such as Dmin7/G7), in an english-like syntax:

```
recognize major Two-Five
FOR any c1 c2 instances of IsolatedChord
IF
   c1 is minor.
   c1 has no flat fifth.
   c2 is after c1.
   c2 is major.
   c2 has a minor seventh.
   c2 root = fourth of the root of c1.
THEN
Create a TwoFive object, covering durations of c1 and c2, and analyzed in the tonality: fourth of the root of c2, major scale.
```

Rule 1. A rule to detect a "two-five" shape. The rule is executed for all couple of objects matching the IF part. These objects are instances of any subclass of Shape.

Other rules describe shapes such as resolutions (A7 / D), turnarounds, and substitutions, as well as more complex shapes such as modal borrowing. Aggregation rules are also used to describe macro shapes, covering the whole chord sequence. For instance, the pattern identifying a Blues shape may be expressed by Rule 2:

```
recognize Blues
FOR any s1 s2 s3 instances of Shape
IF
   s1 is analyzed in X.
   s2 is analyzed in Y, where Y = fourth of X.
   s3 is analyzed in X.
   s2 is after s1. s3 is after s2.
   Begin beat of s1 = 1.
   End beat of s3 = length of the chord sequence.
THEN
Create a BluesShape object, covering the whole chord sequence and analyzed in X.
```

Rule 2. A rule to recognize a Blues by a succession of three shapes.

The second type of rules describe typical situations in which recognized shapes may be safely forgotten, to speed up the reasoning process, and avoid combinatorial explosion. Such rules embody knowledge on "omission" according to the vocabulary of (Cook, 1987), i.e. situations in which analyzable shapes may be safely forgotten, and removed from the working memory. A typical example is Rule 3, that allows safely forgetting a shape without loosing information, when it is subsumed by another shape of the same tonality:

```
removeSubsumedShapes
FOR s1 s2, instances of Shape
IF
   s1 subsumes s2.
   s1 is different from s2.
   s2 is analyzed. s1 is analyzed.
   Tonality of s1 = tonality of s2.
THEN
   remove analysis s2
```

Rule 3. A rule to forget irrelevant shapes.

3.6.2. Circularity

The system as it is presented here suffers from a problem related to the harmonic stability of starting chords. Because the system favors longer shapes rather than small ones, it may happen that small shapes occurring at the beginning of a tune may be eaten up by larger shapes following them, when these shapes are in close tonalities. For instance, in the tune *Blues for Alice* of Figure 1, the underlying tonality of the starting chord (F major) could be interpreted in the tonality of the second shape, i.e. in Bb major, as a IVth degree of Bb major. This would have the advantage of forming a large shape in Bb (*ExtendedShape*, in our ontology). However, it is not the right analysis here, the initial chord in F being clearly a 1st degree of F. Our experiments showed that this correct analysis can be ascertained by simply remarking that the tune is circular. By linking the starting chord with the unresolving seventh chord (C 7) of the end of the tune, the system is then able to discover a shape of non atomic length in F: the unresolving end of the tune ensures the tonal stability of the beginning of the tune (Cf. Figure 6).

D-7 (II)	G 7 (V)	C (I)	

C (I)			
		D-7 (II)	G 7 (V)

Figure 6. A normal 2-5-1 on the left. A 2-5-1 that wraps around the end/beginning of the song on the right.

More generally, we frequently need to manipulate abstract temporal shapes that can wrap around the beginning of a song. Although we could use a purely linear model of time (such as Allen's), this would imply a systematic test for each shape to recognize. This led us to introduce a circular representation of time in our model, described in details in (Pachet et al., 1996). This model allows us to describe configurations of shapes in a tune using circular relations, instead of the usual linear relations. We showed that such a model allows the reasoning system to correctly recognize the harmonic stability of initial shapes.

The overall reasoning is therefore represented as a series of rule bases alternating shape recognition and shape forgetting, using the conceptual hierarchy and a circular model of time. The precise list of rules and scheduling of tasks in described in details in (Pachet, 1997). At the end of the reasoning process, a complete analysis tree is produced, such as the one in Figure 5. The system proved capable of analyzing correctly most standard chord sequences found in the reference corpuses, including Blues sequences deemed difficult by (Steedman, 1984), as well as other non-Blues chord sequences (from *Autumn leaves* to *Nardis* or *Stella by starlight*). Let us mention now one problem that illustrates the limitation of the approach.

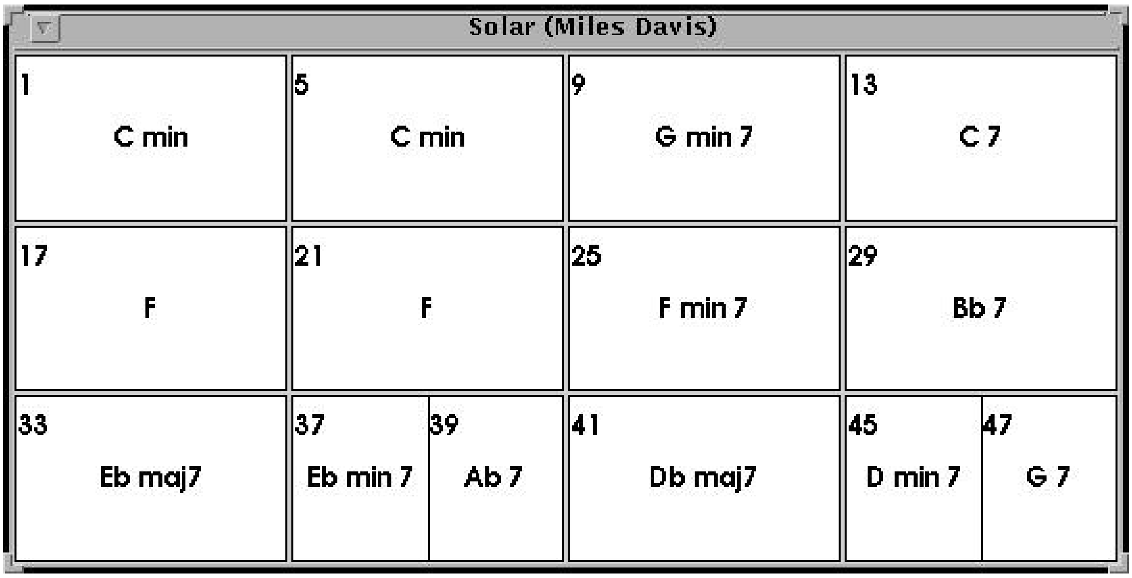
3.7. The Solar problem

Solar is a tune composed by Miles Davis in the 50's (see Figure 7). This tune is generally recognized as a Blues in C minor by Jazz musicians. However, to our knowledge, no system, including ours, would be able to classify it under this category, hence the "*Solar* problem". Maxwell's system would probably recognize cadences and resolutions, but would not say anything about the global structure. Ulrich's system does not provide a hierarchical view of the piece either, and would not recognize embedded modulations, nor specific harmonic patterns. Steedman's grammar would simply return false, i.e. the tune is not a Blues, because it cannot be derived from its set of rules (we did not actually *prove* this impossibility, but only tried unsuccessfully to find the derivation).

Our system produces an analysis which is not what a human would do, but which is nevertheless interesting: a *Pentatonal* shape, globally analyzed in Bb major, and decomposed as follows:

1-12: an *ExtendedShape* in Bb major.
9-24: an *ExtendedSh ape* in F.
25-36: a *ChordSubstitution* in Eb.
37-44: a *ChordSubstitution* in Db.
45-48: a *ChordSubstitution* in C major.

This result is surprising at first glance, but looking at it in more detail it is not absurd. The first shape is analyzed in Bb major, which is false, but harmonically plausible. The mistake here is to try to make a large shape including the three first chords, and Bb major does allow to do that. C minor - the correct answer - was actually discovered, but removed in subsequent steps of the reasoning process, because it could not resist the weight (in size) of the larger Bb shape. The problem comes from the ending two-five, which is correctly analyzed as a two-five in C major, but which is here a kind of "mistake". Had the final two-five been in C minor - for instance by having a (D min 7 *flat 5*) instead of (D min 7) - the initial shape in C minor would have acquired the necessary stability to establish itself as the right tonality, thanks to the circular reasoning explained in the preceding section. The culprit is the system's intolerance on major/minor substitutions which are so prevalent in Jazz tunes. But this is a "minor" mistake, from our point of view, which could be corrected by relaxing the major/minor constraints in the aggregation rules. The rest of the analysis is correct, but of course the Blues structure is not discovered. Recall that in order to recognize a Blues shape, the system ought to find *three* shapes, related by the constraints outlined in Rule 2. The main problem here is the inability of the system to "aggregate" the local streak of eccentricity of measures 37-44 (incidentally correctly analyzed in Db) into the passage in Eb, which itself should be interpreted as a major equivalent of C minor, in order to "see" the underlying, hidden Blues structure. Instead, the system simply sees a *PentatonalShape* (there are eventually five shapes covering the whole sequence), whose tonality is, by definition, the tonality of its first shape, hence the erroneous global tonality in Bb major.



Solar (Miles Davis)

1 C min	5 C min	9 G min 7	13 C 7
17 F	21 F	25 F min 7	29 Bb 7
33 Eb maj7	37 Eb min 7 39 Ab 7	41 Db maj7	45 D min 7 47 G 7

Figure 7. The tune *Solar* by Miles Davis is a Blues, but how can the system see it ?

A human analysis of *Solar* could be the following: the tune is a Blues in C minor, with a set of substitutions that form a logical sequence of 2-5-1. The composer chose to extend this sequence, by a "logical" 2-5-1 (the Eb min 7 / Ab 7 / Db maj7 part in measures 37-44). He did so, however at the cost of breaking the initial Blues structure, since the ultimate 2-5-1 is no longer analyzable in the tonality of Eb major (or C minor). But the resulting sequence is a Blues, so a musician would feel, because some kind of essential property of the Blues is still preserved. That this essential quality escapes our system, as well as other approaches on the same problem, does not prevent the musician to see *Solar* as a Blues, and of the best sort.

4. Discussion, Conclusion

We feel that the solution to the problem of Jazz chord analysis as it is described here is a good solution in the sense that it allows to represent faithfully a large corpus of knowledge related to harmonic analysis, and that it is validated by the results obtained, especially in comparison with other approaches. One of the main qualities of the system is its ability to analyze "incorrect" tunes, at least partially, and to produce results which are directly understood, and usable, by humans.

This experiment raises theoretical and technical issues for computer scientists. The mechanism described here was the source of a larger work on ontologies of reasoning mechanisms, which resulted in a general framework for representing hierarchical temporal reasoning, of which our analysis is a special case. The framework has an important application in the medical field, for the automatic control of respiratory devices in intensive care units (Dojat et al., 1997). Another issue is the comparison between the power of an entirely constructivist approach such as ours, using *forgetting rules*, with descriptive methods such as generative grammars, which rely on backtracking mechanisms.

The experiment also raises issues concerning musicologists. In our context, a circular model of time allowed to solve a technical problem. But the importance of circularity in Jazz tunes remains an issue for musicologists interested in the harmony of Jazz. The question of whether or not circularity has an importance is posed here in unusually precise terms.

Another issue concerning musicologists is the status of theory, as exemplified by the *Solar* problem. In the context presented here, the *Solar* problem could be solved easily in two ways, none of which are convincing. The first solution would be to add more rules to the system (ours or Steedman's). A rule could say for instance, that particular 5-part shapes such as the one our system discovered are indeed Blues. We could also devise a rule that somehow aggregates a part in Db with a shape in Eb, according to some "continuation" principle. In all cases, these rules would clearly play the role of *ad hoc* patches, and would also lead the system to qualify non Blues chord sequences as Blues. The other solution would be to simply stipulate that *Solar* is not a Blues, at least in its primary form, and that considering it a Blues is a matter of taste, personal education, or some other kind of irrational human conduct, which lies beyond the responsibility of a pure analyst. In both cases, the solutions amount to giving up on the explanatory power of the model. This failure simply shows that theories are not strong enough to support the construction of simulators with a degree of sophistication attained by human experts. Our Jazz analysis system will probably never be able to produce subtle arguments pro and con when asked the question "is *Solar* a Blues or not ? ", not because the techniques used are bad, but because there is no reasonable theory of the Blues available.

However, there are limitations to purely automatic analysis from surface input only, such as, in our example, a bare collection of chord names. Two directions of research may prove interesting to escape from the world of syntax. One important aspect of musical analysis is the semantics implicitly used for interpreting musical data. New proposals have been made for instance by Steedman in (Steedman, 1995) in this direction, using the two dimensional space of (Longuet-Higgins, 1962).

Another direction concerns the relation between analysis and emotions, as we suggested in the introduction of this chapter. Automatic analysis could integrate results in Cognitive psychology, by taking into account emotional aspects of musical perception. (Riecken, 1992) showed how a rudimentary model of emotions could be used in a model of musical creativity. Additionally, integrating models of musical *memory* in analytical models (as argued by (Smoliar, 1992)) could help understanding the inner mechanism of analysis by posing the same questions from a different perspective. The question would then not be about an abstract truth (*Is Solar a Blues ?*), but rather about a subjective judgment (*do you think it is a Blues ?*). An explicit representation of musical memory, accounting for past, active, and emotionally rich experiences with numerous Jazz chord sequences could then provide an answer such as: "*I enjoy Solar as a Blues*".

5. Acknowledgment

I wish to thank Professor Jean-François Perrot for his careful and patient reading of earlier versions of the manuscript, his insightful advice and his continuous support.

6. References

Alphonce, B. H. (1980). “Music analysis by computer.” *Computer Music Journal*, 4(2), 26-35.

Baggi, D. (1992). “NeurSwing: An Intelligent Workbench for the Investigation of Swing in Jazz.” Computer-Generated Music, D. Baggi, ed., IEEE Computer Society Press, 79-93.

Baker, M. (1989a). “An Artificial Intelligence approach to musical grouping analysis.” *Contemporary Music Review*, 3, 43-68.

Baker, M. (1989b). “A cognitive model for the perception of musical grouping structures.” *Contemporary Music Review*, Spring(Special issue on Music and the Cognitive Sciences).

Baroni, M. C., L. (1984). *Musical Grammars and Computer Analysis. Atti del Convegno (Modena, 4-6 ottobre 1982)*, Olschki, Florence, Italy.

Beaudoin, P. (1990). *Jazz, mode d'emploi*, Outre Mesure, Paris.

Bein, J., and Winold, A. (1983). “Banalyse: An Artificial Intelligence System for Harmonic Analysis of Bach Chorales.” *Unpublished manuscript*, Indiana University.

Bent, I., and Drabkin, W. (1987). *Analysis*, Macmillan Press, London.

Brinkman, A. R. (1980). “Johann Sebastian Bach's Orgelbüchlein.” *Music Theory Spectrum*, 2, 46-73.

Brinkman, A. R. (1986). “Representing musical scores for computer analysis.” *Journal of Music Theory*, 30(2), 225-275.

Brinkman, A. R. (1990). *Pascal Programming for Music Research*, The University of Chicago Press.

Byrd, D. (1977). “An Integrated Computer Music Software System.” *Computer Music Journal*, 1, 55-60.

Camilleri, L., Carreras, F., and Duranti, C. (1990). “An Expert System Prototype for the Study of Musical Segmentation.” *Interface*, 19(2-3), 147-154.

Camilleri, L., Carreras, F., Grossi, P., and Nencini, G. (1987). “A Software Tool for Music Analysis.” *Interface*, 16(1-2), 23-38.

Castine, P. (1994). *Set theory objects. Abstractions for computer aided analysis and composition of serial and atonal music*, Peter Lang, Frankfurt am Main.

Chemillier, M. (1990). “Langages musicaux et automates: la rationalité du langage sériel.” *Colloque International "Musique et Assistance Informatique"*, Marseille, 211-227.

Chemillier, M. (1995). “Analysis and computer reconstruction of a musical fragment of Ligeti's Melodien.” *Muzica*, 7(2), 34-48.

Chouvel, J. M. (1990). “Musical Form: From a Model of Hearing to an Analytic Procedure.” *Interface*, 22, 99-117.

Coker, J. (1964). *Improvising Jazz*, Simon & Schuster, Englewood Cliffs, New Jersey.

Cook, N. (1987). *A Guide to Musical Analysis*, Oxford University Press, Oxford.

Dojat, M., Pachet, F., Guessoum, Z., Touchard, D., Harf, A., and Brochard, L. (1997). “NéoGanesh: a Working System for the Automated Control of Assisted Ventilation in ICUs.” *Artificial Intelligence in Medicine. Special issue on Decision Support in the Operative Theatre and Intensive Care.*

Dojat, M., and Sayettat, C. (1994). “A realistic model for temporal reasoning in real-time patient monitoring.” *Applied Artificial Intelligence*, 10(2), 121-143.

Ebcioglu, K. (1987). “Report on the CHORAL project: an expert system for harmonizing four-part chorales.” *RC 12628*, IBM Watson Research Center, Yorktown Heights.

Ebcioglu, K. (1992). “An Expert System for Harmonizing Chorales in the Style of J.-S. Bach.” Understanding Music with AI: Perspectives on Music Cognition, M. Balaban, K. Ebicioglu, and O. Laske, eds., AAAI Press.

Elliott, R. (1987). “The development of an algorithm for the detection of cadences.” , Eastman school of Music, unpublished paper.

Fake. (1983). *The World's Greatest Fake book*, Sher Music Co, San Franscisco.

Forte, A. (1973a). “The Basic Interval Patterns.” *Journal of Music Theory*, 17(2), 234-272.

Forte, A. (1973b). *The Structure of Atonal Music*, Yale University Press, New Heaven.

Frankel, R. E., Rosenschein, S. J., and Smoliar, S. W. (1978). “Schenker's theory of tonal music-Its explication through computational processes.” *International Journal of Man-Machine Studies*, 10, 121-128.

Fry, C. (1980). “Computer Improvisation.” *Computer Music Journal*, 4(3), 48.

Giomi, F., and Ligabue, M. (1991). “Computational Generation and Study of Jazz Music.” *Interface*, 20(1), 47-63.

Greussay, P. (1973). “Modèles de descriptions symboliques en analyse musicale,” Ph.D., Université Paris 8.

Greussay, P. (1985). “Exposition ou exploration: graphes Beethovéniens.” Quoi ?, Quand ?, Comment ? La recherche musicale, Christian Bourgois, IRCAM, Paris, 165-183.

Harris, C. R., and Brinkman, A. R. (1986). “A unified set of software tools for computer-assisted set-theoretic and serial analysis of contemporary music.” *International Computer Music Conference*, La Haye, 331-336.

Hasty, C. F. (1978). “A theory of segmentation developed from late works of Stefan Wolpe,” Ph. D., Yale University.

Horowitz, D. (1995). “Representing Musical Knowledge: Processing Melodic Lines in a Jazz Improvisation.” *International Congress In Music and Artificial Intelligence*, Edinburgh, 103-118.

Huron, D. (1994). “The Humdrum Toolkit Reference Manual.” , Center for Computer Assisted Research in the Humanities, Menlo Park.

Järinen, T. (1995). “Tonal Hierarchies in Jazz Improvisation.” *Music Perception*, 12(4), 415-437.

Johnson-Laird, P. N. (1991). “Jazz Improvisation: A Theory at the Computational level.” Representing Musical Structures, P. Howell, R. West, and I. Cross, eds., Academic Press, 291-325.

Kassler, M. (1975). “Proving Musical Theorems I: the Middleground of Heirinch Schenker's Theory of Tonality.” *103*, University of Sydney, Sydney.

Kunst, J. (1987). “Remarks on analysis.” *Interface*, 16, 1-11.

Laurson, M., and Duthen, J. (1989). “PatchWork, a graphical language in PreForm.” *International Computer Music Conference*, San Francisco, 172-175.

Lerdahl, F., and Jackendoff, R. (1983). *A Generative Theory of Tonal Music*, MIT Press, Cambridge.

Lincoln, H. B. (1970). *The Computer and Music*, Cornell University Press, Ithaca.

Longuet-Higgins, H. C. (1962). “Letter to a musical friend.” *Music Review*, 244-248.

Mason, R. M. (1985). *Modern methods of music analysis using computers*, Schoolhouse Press.

Maxwell, H. J. (1984). “An Artificial Intelligence Approach to Computer-Implemented Analysis of Harmony in Tonal Music,” Ph.D., Indiana University.

Maxwell, H. J. (1992). “An Expert System for Harmonic Analysis of Tonal Music.” Understanding Music with AI: Perspectives on Music Cognition, M. Balaban, K. Ebcioglu, and O. Laske, eds., AAAI Press, 335-353.

Meehan, J. (1980). “An Artificial Intelligence Approach to Tonal Music Theory.” *Computer Music Journal*, 4(2), 61-65.

Mesnage, M. (1993). “Morphoscope, a Computer System for Music Analysis.” *Interface*, 22(2), 119-131.

Mesnage, M. (1995). “Un logiciel de modélisation de partitions comme outil commun à l'analyse et à la composition musicale.” *Deuxièmes Journées d'Informatique Musicale*, Paris, 31-40.

Mesnage, M., and Riotte, A. (1990). “Un modèle informatique du 3ème regard sur l'Enfant-Jésus d'Olivier Messian.” *Colloque International "Musique et Assistance Informatique"*, Marseille, 187-209.

Minsky, M. (1985). “Musique, sens et pensée.” Quoi ?, Quand ?, Comment ? La recherche musicale, Christian Bourgois, IRCAM, Paris, 137-163.

Minsky, M. (1986). *The Society of Mind*, Simon and Schuster, New-York.

Mouton, R. (1995). “Outils intelligents pour les musicologues,” Ph.D., Université du Maine, Le Mans.

Nelson, O. (1961). “The Blues and the abstract truth.” , Impulse.

NewReal. (1991). *The New Real Book*, Sher Music Co, Petaluma.

Pachet, F. (1994a). “The MusES system: an environment for experimenting with knowledge representation techniques in tonal harmony.” *First Brazilian Symposium on Computer Music - SBC&M '94*, Caxambu, Minas Gerais, Brazil, 195-201.

Pachet, F. (1994b). “An object-oriented representation of pitch-classes, intervals, scales and chords.” *Premières Journées d'Informatique Musicale*, LaBRi, Université de Bordeaux, 19-34.

Pachet, F. (1997). “Analyse harmonique de séquences d'accords par objets et règles.” , Laforia, Université Paris 6, Paris.

Pachet, F., Ramalho, G., and Carrive, J. (1996). “Representing temporal musical objects and reasoning in the MusES system.” *Journal of New Music Research*, 25(3), 252-275.

Polansky, L. (1979). “A hierarchical gestalt analysis of Ruggle's Portals.” *International Computer Music Conference*, 790-852.

Rahn, J. (1980a). *Basic Atonal Theory*, Longman.

Rahn, J. (1980b). “On Some Computational Models of Music Theory.” *Computer Music Journal*, 4(2), 66-72.

Ramalho, G., and Ganascia, J.-G. (1994). “Simulating Creativity in Jazz Performance.” *12th AAAI Conference*, Seattle, 108-113.

Real. (1981). *The Real Book*, The Real Book Press.

Riecken, D. (1992). “Wolfgang - A system Using Emoting Potentials to Manage Musical Design.” Understanding Music with AI: Perspectives on Music Cognition, M. Balaban, K. Ebcioglu, and O. Laske, eds., AAAI Press, 207-236.

Roads, C. (1988). “Grammars as Representations for Music.” Foundations of Computer Music, C. Roads and J. Strawn, eds., MIT Press, 403-442.

Rokita, L. (1996). “Modèle rythmique d'une pièce pour clarinette d'Igor Stravinsky.” *Troisièmes Journées d’Informatique Musicale, JIM’96*, Ile de Tatihou, 277-286.

Rothgeb, J. (1968). “Harmonizing the Unfigured Bass: a Computational Study,” Ph.D., Indiana University.

Rothgeb, J. (1971). “Musical research by computers: some current limitations.” *Computer and the Humanities*, 5(3), 178-182.

Rowe, R. (1993). *Interactive Music Systems*, MIT Press.

Ruwet, N. (1972). *Langage, musique, poésie*, Le Seuil, Paris.

Schank, R. (1973). “Identification of conceptualizations underlying natural language.” Computer Models of Thought and Language, R. Schank and K. Colby, eds., Freeman, San Francisco.

Smaill, A., and Wiggins, G. (1990). “Hierarchical music representation for composition and analysis.” *Colloque International "Musique et Assistance Informatique"*, Marseille, 261-279.

Smoliar, S. W. (1980). “A Computer Aid for Schenkerian Analysis.” *Computer Music Journal*, 4(2), 41-59.

Smoliar, S. W. (1992). “Music Notation: Cognitive Red Herring.” *IJCAI-89 Workshop on AI and Music*, Detroit, Michigan, 53-62.

Steedman, M. J. (1984). “A Generative Grammar for Jazz Chord Sequences.” *Music Perception*, 2(1), 52-77.

Steedman, M. J. (1995). “The Blues and the abstract truth: music and mental models.” *Draft.*

Stroppa, M. (1984). “The analysis of electronic music.” *Contemporary Music Review*, 1(1), 175-180.

Tenney, J., and Polansky, L. (1980). “Temporal Gestalt Perception In Music.” *Journal of Music Theory*, 24, 201-241.

Ulrich, W. (1977). “The Analysis and Synthesis of Jazz by Computer.” *Fifth International Joint Conference on Artificial Intelligence*, MIT, Cambridge, Ma, 865-872.

Walker, W., Hebel, K., Martirano, S., and Scaletti, C. (1992). “ImprovisationBuilder: improvisation as conversation.” *International Computer Music Conference*, San Jose (Ca), 190-193.

Winograd, T. (1968). “Linguistic and Computer Analysis of Tonal Harmony.” *Journal of Music Theory*, 12, 2-49.