A Melody Description System for Jazz Improvisation

by

David Alex Levitt

B.S., Yale College
1977

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR
THE DEGREE OF

MASTER OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1981

Signature of Author_____
Department of Electrical Engineering and Computer Science
May 8, 1981

Certified by_____
Marvin Minsky
Thesis Supervisor

Accepted by_____
Arthur Smith
Chairman, Department Committee

A Melody Description System for Jazz Improvisation

by

David Alex Levitt

Submitted on May 8, 1981 in partial fulfillment of the
requirements for the degree of Master of Science in
Electrical Engineering and Computer Science

Abstract:

This thesis describes a program which embodies several computational aspects of music improvisation and composition. Given an initial melody and harmonic outline, the program produces a single voice of improvisation. While the improvisation is intended to resemble a jazz (e.g. "swing" or "bebop") solo, the program's descriptive techniques reflect computational elements common to a range of tonal music. These include building up phrases from partial descriptions, recognizing interesting features, and re-examining a remembered phrase to produce a variation in a new context. Throughout, the program uses various ideas of similarity, repeated patterns, defaults, and mutually constraining structures, trying to anticipate the audience's expectations to make the improvisations understandable and interesting.

CONTENTS

FIGURES

1. Introduction

People often talk of music making as though it does not involve intelligence: only esthetic, intuition, and feeling. But this is an excessively romantic view; we certainly solve problems when we make music. Composers and improvisors (from here on I'll use "musician" to mean both) fit melodic contours into new harmonic contexts, avoid "dissonances", and generally find ways of satisfying some description we have of the music we're trying to make. The solution to a specific, completely defined musical problem is often easy to find, and sometimes there will be many obvious solutions; other times we may need to search, and sometimes no solution can be found.

But where do the problem descriptions come from? First, many constraints come from the dialect or "style" of the music we've decided to make (which may in turn arise from an implicit contract with other musicians, or the audience). Beyond this, the question is problematic. For the musician, the problem is often simply "compose something interesting".

Issues like these distinguish Music from most other problem solving domains. The criteria for a good answer are very different from those of Engineering or Games. In a way it is subtler than many Natural Language problems -- imagine asking a language production program to look into its database and "say something clever".

Music demands a theory of attention because *the musician directs the thoughts of audience.* To do this, he begins with an estimate of what the audience already "knows", and uses this to control its attention.

I present a program embodies some ways of doing this. It uses overlapping, partial descriptions of musical events at several levels. Each note or phrase may have several harmonic, intervallic, and rhythmic descriptions, any of which could be important, or serve a particular function.

These descriptions are used in two ways: first, we can assign partial descriptions to a note or phrase event and later produce a complete event that satisfies them; and we can decompose a remembered event in several ways, choosing "interesting" decompositions as ingredients for new events.

Most importantly, the program embodies several kinds of tonal *similarity* which govern the program's *default behavior*, reflecting some musical psychology. The musician acknowledges the audience's limited information processing abilities (and conserves his own) by making "predictable" choices for incompletely described notes and phrases. In much tonal music, default behavior includes regular rhythms, small melodic motions within any voice, compatibility with the apparent harmonic plan, and frequent imitation of features in an earlier phrase or cliche motif. These are interdependent and often conflicting; much musical reasoning is concerned with negotiating between them. Within this context, larger "distances" from expected behavior are introduced and exploited to direct and hold the attention of the audience.

1.0.1 Psychology vs. Musicology

It has not usually been realized how important it is in theorizing about music to distinguish psychological issues from *musicological* concerns -- which here means the study of specific dialects. This focus has been a major guideline for the thesis. Various systems have tried to capture the "system of rules" that characterize features of a style, such as automating counterpoint exercises. That approach often obscures psychological issues, and is notably weak in explaining *novelty* -- when to bend or break the rules, and how to make new ones -- though this seems central to understanding what makes music interesting and evolve.

Here I have tried to cover more ground in less dialect-specific detail, emphasizing instead the psychological motivation behind each decision. Measures of similarity, which govern defaults, and *mutually constraining structures*, perform the functions ordinarily characterized by stylistic rules.

Still, it would be hard for listeners to evaluate the product of an improvisation program that wholly ignored their culture. In fact, many of the structures I have treated apply specifically to Western, "tonal" music in its various forms. Ideally, one would apply the description system to several dialects; but each interesting tonal style is sufficiently complex that to convincingly reproduce some would be well beyond the scope of this thesis. Instead, the description system has applied within the general "jazz" framework. Given a harmonic outline and initial melody, the program produces a single voice "solo", whose construction exercises the system's multi-level reasoning capabilities.

2. Basic tonal structures

This section describes some structural features common to much tonal music. It serves as an introduction to many of the ideas in the program, and includes an outline of the vocabulary I will use.

I will only consider pitch and time, and things constructed from them. Amplitude and timing dynamics, orchestration, etc. are important, but symbolic manipulations of them seem too subtle for study at this early stage. Here I ignore them, focussing instead on melody as it relates to harmony in time. Moreover, while some remarks concern the psychology of communication, discussions of music will usually refer to tonal, rhythmic Western music, with its triadic and diatonic structures, chromatic scale, and metrical rhythms. (Hereafter, simply "tonal music".)

2.1 Terminology

First I will to try define the vocabulary I'll be using. This section is brief, assuming basic terms like "pitch", "octave" "quarter note", "lower than" etc. to be understandable. I realize musical terms may be completely foreign to some readers, but this thesis can't include a more detailed music course. The terminology is fairly standard, and should be familiar to readers with a musical vocabulary. Still, if my use of the terms seems non-standard or labored at times, it is probably because I'm trying to use the terms in a somewhat different way.

Most musical terms don't describe things that are explicit in the score. Words like "pitch" and "duration" are fairly unambiguous; but what do we mean when we talk about a "chord" or a "key"? Often, not every chord element (from the "set of tones" which commonly describe the chord we hear) is present; still less often are all the pitches of a key. These terms are a way of communicating about the symbolic tools we use to understand or build a piece. The tools are so useful that it is usually convenient to talk about a chord as though it were an unambiguous, easily defined object. But as is often the case, this is perilous if we intend to study the mental tools themselves.

When we mention a "G dominant seventh chord" to a music student, we communicate a whole network of structures and expectations: the "default" G, B, D, and F tones we are likely to hear for the next little while, especially on the "stronger beats"; the tones right near them (Gb, G#, Bb, C, Db, D#) which we are *un*likely to hear, unless they are leading up to one of the default four; a sense in which G is "center" of the group -- so that, for instance, upcoming melodic motion might be analagous to recent motion around some earlier "center"; an expectation that, because it is a Dominant seventh, the next "center" will be C; and so on. Similarly, "key" is used to describe a set of tones, relationships between successive chords, and a "center", all at once.

Even without a technical music vocabulary, most listeners have representations like the "dominant seventh chord", which they recognize and use in all these ways, anticipating what will happen next. Musicians (often more articulately) have similar symbols, along with ways to realize the expectations in the form of little plans.

Since many Western music texts are unconcerned with psychological theories, and treat a narrow range of dialects, it is often convenient to leave these mechanisms mixed up. Here I have tried to

separate them; for example, there is no "key" data structure, although various functions commonly associated with "key" are used throughout. In general, I want to describe how representations are used, separating data-structure definitions from the expectation-systems that employ them.

2.1.1 Harmonic terms

The main harmonic principle in tonal music is *octave equivalence*: a C in any octave has an audible sameness. This divides the pitches of any chromatic instrument into 12 distinct harmonic classes, A, A#, C, ... G#. [1] Often called the 12 "pitch classes" (e.g. [Forte]), I will here simply call them the **tones**. A pitch is fully described by its tone and octave, the first octave starting with A0 at the bottom of the piano. Therefore middle-C is C3, and its tone is C.

The other basic harmonic principle might be called *harmonic relativity*: we usually locate a tone not by its absolute value, but by its relationship to some other tone (simultaneous, recent, or somehow implied) which functions as a harmonic base or center. We thus describe **harmonic intervals** between a **root** tone and a related tone. This can range from a Unison to a Major-7th up or down. Note that since the tones form a single-octave circle, a harmonic interval up a Major-Seventh is equivalent to a harmonic interval down a Minor-2nd, with respect to the same root. The table in Figure 1 shows the common names for the harmonic intervals within the octave, expressed as a distance in chromatic steps.

Observe that many of the interval names above are paired, like "Minor-3rd" and "Major-3rd", or "Perfect" and "Diminished" Fifth. Since most chords and modes contain only one element from any such a pair, we can often simply refer to "the 3rd" of the quality, and refer to both harmonic intervals as on the 3rd **harmonic degree**. This is really a kind of *approximation abstraction*, which we will use again later.

Next we describe the chief means by which different tones are collected into a single harmonic class -- the **harmonic quality**. Harmonic qualities describe a set of harmonic intervals with respect to a single, variable **root** of the quality. **Chord and mode types** are kinds of harmonic qualities, though (as we will see later) they are often used differently. The table in Figure 2 shows the important chord and mode types I will refer to throughout the thesis. Of course there are more.

A chord or mode is completely described by combining a quality with a particular root tone, so that the "G Dominant-7th" chord indeed describes G, B, D, and F with respect to G. The chord or mode divides the tones into **consonant** tones, which are members of such a set, and the other **dissonant** tones. A sequence of chords or modes in time is a **progression**. A change of mode is a **modulation**.

---

1. I refer to the chromatic scale as a pianist does, distinguishing the different spellings of a pitch only with regard to notational convenience.

Fig. 1. Common names for Harmonic Intervals

Distance	Common symbol
0	Unison (or Octave)
1	Minor-2nd
2	Major-2nd
3	Minor-3rd
4	Major-3rd
5	('Perfect') Fourth
6	Tritone (or Diminished Fifth)
7	('Perfect') Fifth
8	Minor-6th
9	Major-6th
10	Minor-7th
11	Major-7th

Fig. 2. Basic Chords and Modes

Quality Symbol	Intervals with respect to Root
BASIC TRIADIC CHORDS	
Major	Unison, Maj-3rd, Fifth
Minor	Unison, Min-3rd, Fifth
Diminished	Unison, Min-3rd, Tritone
Augmented	Unison, Maj-3rd, Min-6th
7TH CHORDS	
Dominant-7th	Unison, Maj-3rd, Fifth, Min-7th
Major-7th	Unison, Maj-3rd, Fifth,Maj-7th
Minor-7th	Unison, Min-3rd, Fifth, Min-7th
Half-Diminished	Unison, Min-3rd, Tritone, Min-7th
Diminished-7th	Unison, Min-3rd, Tritone, Maj-6th
MODES	
Major-scale	Unison,Maj-2nd,Maj-3rd,Fourth,Fifth,Maj-6th,Maj-7th
Melodic-minor	Unison,Maj-2nd,Min-3rd,Fourth,Fifth,Maj-6th,Maj-7th
Natural-minor	Unison,Maj-2nd,Min-3rd,Fourth,Fifth,Min-6th,Min-7th

I stress again that while these terms differ somewhat from those encountered in music texts, they are useful for us here. Also, some readers may find the idea of absolute consonance/dissonance too rigid; sometimes different pitches seem more and less "compatible" with a particular triadic chord, but by degrees. Later I will describe different ways of using this consonance idea, and others, to construct more subtle notions of which pitch "goes best" in a given harmonic context; but I will

continue to use "consonance" and "dissonance" as I have defined them here.

Closely related to the idea of consonance is that of **disambiguator**. Any harmonic progression must ultimately be expressed by the progress of pitches in the voices; we do not ordinarily acknowledge a change in the chord unless one of the voices has changed to imply this. [1] Still, when following a chord progression, one usually needn't hear *all* the pitches of the new chord at every chord change. The tones of the previous chord become the defaults, and new chords are detected by hearing differences from these. It thus becomes useful to distinguish the disambiguator set -- the set of tones consonant with the current harmonic in the progression, but dissonant with the previous one -- in our vocabulary. Any progression of chords or modes implies a corresponding progression of disambiguating tones.

Note that due to octave equivalence, the pitches that express the tones of a chord may appear in any octave. We call the various expressions of a chord or mode, in whatever octaves, **inversions** of the harmonic entity.

2.1.2 Intervallic terms

Note that "melody" commonly refers to any aspect of musical structure which is not obviously harmonic or rhythmic; here we will separate the notion into its local, "horizontal" intervals between successive pitches in a voice; and various kinds of "phrase imitation" at a larger scale.

To compare two pitches we introduce the **interval** concept. An interval describes the relationship between a pitch and some "origin" pitch. The interval is simply decomposed into a **direction** -- UP, DOWN, or SAME with respect to the origin -- and a magnitude or **size**, the distance between the two pitches in chromatic steps. It is often convenient to describe intervals of less than an octave using the terms given for Harmonic-interval, above.

Most lay listeners seem to easily distinguish the directional component, but are less sure in questions about size. In fact, many musically trained listeners answer such questions accurately by first fitting the pitches into a harmonic framework. This is evidence that precise interval size is not often strongly or "directly" perceived -- so that again, approximation abstractions are useful.

Finally, because interval direction seems so perceptually important, I include **contour** -- a pattern of several Directions in sequence -- in our intervallic vocabulary. This will be used by the phrase imitation part.

---

1. Actually, musicians play games to make us *expect* some chord change at a particular moment. This can make us "hear" it with less evidence, or be surprised if it does not appear. With the repeated chord progressions of jazz, this is easy. But explaining this in the main text would undermine the whole disambiguator definition.

2.1.3 Rhythmic terms

Any harmonic progression or melody has a rhythmic component. The rhythmic terms used in the program concern the approximate frequency, and relative "phase" of such events.

When we listen to tonal music, we hear pitches and percussion, and infer progress in chords and other harmonic levels. How do listeners interpret the timing of such events?

One of the simplest descriptions we can make is that events are progressing at a roughly constant rate. This is often true at pitch and harmonic levels, and when it is we call that rate the **tempo.** Notationally, the regular measure line sometimes delimits tempo of the chords; faster tempos often depict the melody, while larger regularities can outline repetition structures within the music.

By convention, these different levels are often organized hierarchically, as in Figure 3, so that a new note, chord, or section begins at a new branch in the hierarchy. Listeners can thus interpret a "phase" to the music; articulate ones can usually tell "where the measure line belongs" and generally, "where the beat is". Thus we will use term **beat** to capture the "phase" of a tempo. Moments which appear higher in the hierarchy reflect greater likelihood of a new event at any tempo; these are relatively **strong beats.**

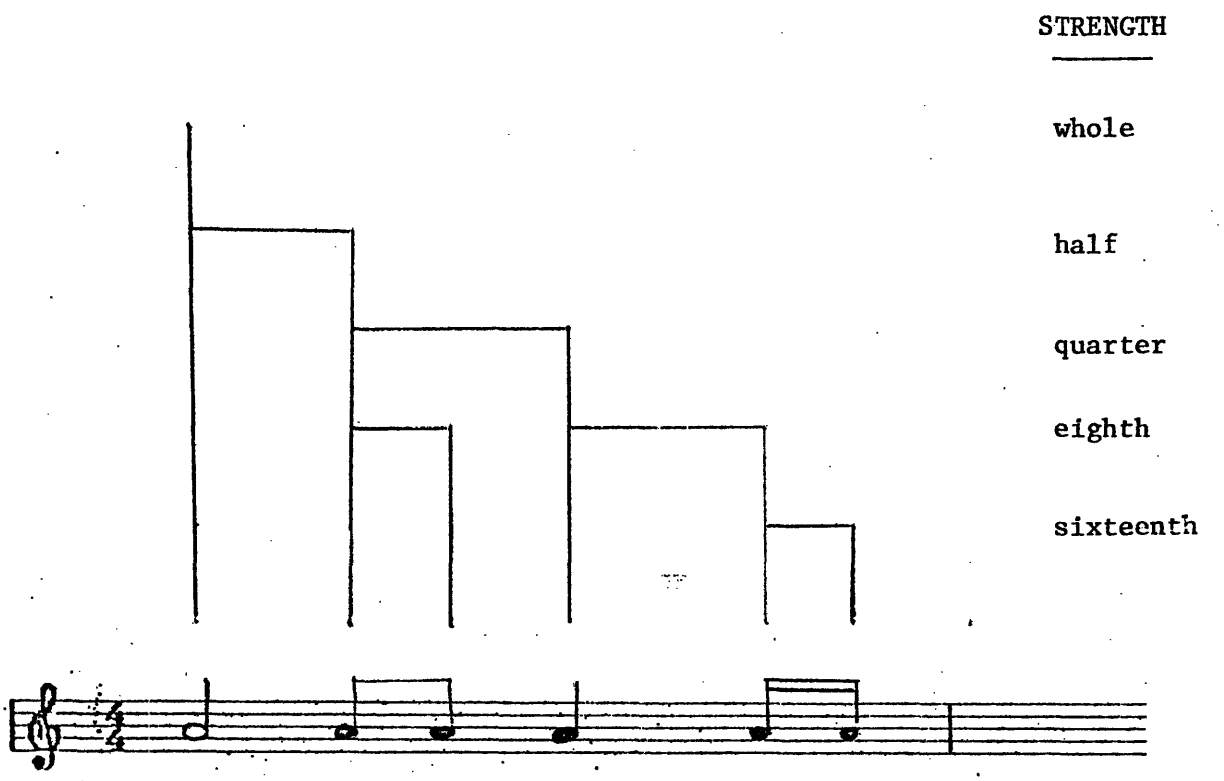
In general, identifying the beat is subtle; the meter changes, deliberate ambiguities are used, etc. Jazz conventions allow us to use a simple definition here, which still reflects fundamental structures in other music. A typical jazz tune has 32 bars in 4/4 time (sometimes with an additional head and tail). We can thus treat the beat pattern as a simple binary hierarchy, as shown in the Figure. The beginning of the first measure has a strength of "32 bars", the moment an 8th note later has a strength of "8th note", the second measure begins on "whole note" strength, etc.

This rather formal definition (similar to that used by [Martin]) accords well with the popular meaning of "strong beat", though that usually refers to such relative strengths within a measure. Stronger beats are our cues as to "where we are" in the piece. Since we expect events to happen on strong beats, we predict that if the tempo is halved, this will happen on an "odd" beat (the left branch in our hierarchy), letting subsequent events continue in phase with the piece. When a new event occurs sooner than we expected it, we can prepare for an increase in tempo.

This provides an easy way to fool listeners. The musician can simply *withold* a new event we predicted using the reasoning above -- by silence, or by continuing the previous one. We call this rhythmic surprise a **syncopation.** Often we hear this as an event whose onset time is "advanced" with respect to its expected time -- the upcoming, stronger beat. This interpretation is common if it meshes with other expectations. For instance, suppose we anticipate a chord change, in which a certain voice is expected to play a consonance. The voice plays a pitch which disambiguates the new chord, but the pitch begins on the final weak beat of the previous chord, continuing into the new one. This common device combines the surprise of the syncopation with the expectation that the disambiguating pitch will continue into the next measure; the "advance" is thus clearly perceptible.

Fig. 3. Typical Rhythmic Hierarchy

(after Martin)



2.2 "Near" -- Dimensions of Tonal Similarity

Already, in motivating the terminology, we have seen the applications of various features emerging. Each term describes a psychological anchor for the listener -- a way of economically "packaging up" what was just heard, and/or predicting what is to come -- or, in the case of syncopation, describing a surprise. We construct these different descriptions while listening.

We now consider two closely related classes of mechanisms for constructing these descriptions. The first includes measures of local and "vertical" proximity or "nearness" -- apparent relationships between simple events that occur in succession, or at the same time. Then I discuss applications of these similarity ideas to repeated patterns in larger structures. Both are used by the jazz program, and are treated in more detail in the description of it.

These ideas of nearness and similarity are important because they are the ways we compress musical information. By taking small steps and doing what is expected most of the time, the musician serves

his purpose -- whether it is to massage the listener with many different kinds of predictable steps (as in MUZAK), or to create interest at a particular level.

Some ideas of musical "nearness" are linear, and obvious. Two notes are intervalically near one another when the size of the interval between them is small, and analogously for tones and harmonic intervals. The same holds when comparing the durations of two events; when they occur in sequence, this is quickly interpreted as tempo, discussed above.

More interesting local proximities arise as we consider harmony. We have already discussed consonance or dissonance of a pitch with a chord or mode; this is a simple sense in which a pitch can be near or far from a harmonic structure. But there are various others.

In one important dimension of tonal similarity, tones separated by a perfect Fourth or Fifth are "near" one another. This occurs independently of the harmonic context, apparently because the tones frequencies are (almost) related by the simple ratio of 1/3 (modulo 2↑k). The same idea can be extended to other "inherently sonorous" intervals, (e.g. a ratio of 1/5 ==> Maj-3rd), but I will not explore this here.[1] Fourths and Fifths are used to structure root motion in harmonic progressions, as in the cadence. For now we will define a cadence as a progression in which the root of each chord is a Fourth above its predecessor. Really, the cadence is a kind of "repeated pattern", and will be treated later.

Harmonic qualities themselves can be similar by having intervals in common (e.g. a Half-Diminished is like the Dominant-7th because they each contain a Tritone). A pair of chords or modes may be similar because they have the same harmonic quality, or similar ones, or because they have roots that are the same, or separated by small harmonic intervals. Two chords may be similar in some context because in that they are both subsets of some other, larger set (viz. D Minor is "near" C Major if a C Major-Mode has been established). In a later section, a "similarity network" for two important Mode qualities -- Major-Mode and Melodic-Minor -- will be discussed as to its application in the program.

The important thing to notice here is that the different levels of description *overlap* and *compete*. The introduction of harmonic structures leads to so many ideas of similarity that we generally have many to choose from -- several from the preceding paragraphs might apply. But different dimensions of similarity can easily conflict. An ascending sequence of pitches can be "all consonant with the same triadic chord" or "no intervals bigger than a Major-2nd", but not both.

For comparisons in which there are several degrees of nearness, the dimension in which the "smallest distance" occurs usually provides the most concise, informative description. Thus, musicians "automatically" relate C Major and A Minor using common tones; C Major and C# Minor for the "chromatic" motion of the root; C Major and G Major by the Fifth in the roots, common quality, and common C Major-Mode parent mode; etc.

---

1. It seems likely that when people measure this, they employ a more general system for recognizing harmonic series -- like the one that identifies octaves. There is evidence for this in the pentatonic scales of other cultures, although the symmetry of the Western chromatic scale is especially suited to music that exploits this.

2.2.1 Repeated patterns, Grammars, and Style

Perhaps the most important ideas of similarity are the ones listeners construct dynamically, during a piece and over a lifetime. When we hear a pair of pitches in sequence that matches a recently heard pair, we may be led to expect the successors from the original sequence to follow. This also holds true for imitation at the other levels of description we have discussed, so that a rhythmic pattern, contour, consonance/dissonance pattern or other feature can invoke a larger context when repeated.

Of course, listeners will most easily expect this kind of repetition if they have been led to it -- for instance, if the musician makes a habit of repeating sequences of pitches. But any "habit" is itself an instance of a repeated pattern! Thus the imitation concept is applied to the various levels of description, and recursively to itself. Generally, imitation turns a complicated thing into a simple, "small distance", by making parts of it expected. Combinations of such repeated patterns ultimately constitute the structure of a musical "style".

Few audiences are so intelligent and attentive to a complex new piece that they can appreciate its structure, unless they recognize structures they have heard before in many ways. Effective communication requires musicians to repeat structures frequently within a piece *and collectively over many pieces*. Usually we view "musical style" and "theme and variation" as utterly different. Computationally and socially they are similar things with different time spans: style tries to exploit long term, "cultural" memory, while theme-and-variation exploits recent events. In either case the considerate composer uses an idea of what is already in the audience's head to make the piece understandable. (In fact, in my program they are thoroughly mixed.)

The prevalence of repeated patterns in music, applied recursively and combined with other abstraction mechanisms, has led to various "grammatical" formulations of musical structure. In the tonal realm, Heinrich Schenker's view of harmony [Schenker] is perhaps the most famous approach; repetition patterns like "AABA form" may be viewed as grammars too.

In our formulation, "grammars" and rules play a secondary role, in a framework of expectations and possible innovations captured by "small distances" and defaults. Recently, various experiments with grammars have suggested extensions which seem necessary to capture the subtlety in music. [Paseman] used two different grammars, employing "meta-rules" to decide between them; while [Lerdahl & Jackendoff] used four different hierarchical decompositions of the same piece (although they did not describe mechanisms by which they could interact). Thus evidence grows indicating that the interesting, universal problems in music concern deciding which *kind* of description is important or useful *when*, as much as understanding the structure of a particular one.[1]

In a similar vein, grammatical approaches which operate outside of psychological theories, and lack notions of "nearness", usually make it difficult to decide which of several "well-formed" options is most appropriate or most suitable. Almost universally ([Hiller] [Laske] etc.), grammatical

---

1. Various authors (e.g. [Narmour]) have argued at length against Schenkerian and other grammatical approaches; I share their view, but will limit my discussion of them here.

composition systems use a pseudorandom noise source to distinguish apparently equivalent choices and make decisions.

My theory offers natural ways for different levels of description to interact, to recognize important features, and to justify one "best" tone (or make competing justifications!) among various options. I will discuss other viewpoints in more detail near the end of the thesis.

2.3 Mutually constraining structures

In the introduction I stressed an issue that influences nearly every musical activity: *the problem is rarely completely defined in advance.* Moreover, the symbolic structures of music are malleable in ways that most problem domains do not encourage. Half-completed projects are discarded just as they become recognizable, their elements cannibalized -- indeed, this can make listening exciting, while the small penalty keeps music-making fun. But this freedom places special demands on the flexibility of musical description systems.

To build some "musical device" -- which might be a joke, suspense, or something subtler -- we start with some elements that are "handy" (both "in mind" and earlier in the piece), and an idea of how the completed device must be structured. For many musical devices, we can start with *any description* of certain parts, and choose the others so that the device is functional. Often this can be done "seamlessly" so that it is not obvious which parts were given and which were added, the way a good artist can weave a coherent sketch around almost any scribble.

This requires that the plan for the device be represented not as a single procedure, but as *mutual constraints* between the parts, which must be satisfied for the device to work. Of course, these constraints are ultimately realized by procedure fragments, each of which operates in particular environments of givens and unknowns; but the procedures themselves are conceptually related, and may contain functional inverses, generate-and-test procedures with common predicates, etc. In our conception of the device, we are served best by a system that lets us describe the relationships themselves, and invokes appropriate procedures as needed.

In improvised classical counterpoint, where harmonic structure may be decided dynamically, the improvisor might decide that the chord tempo is half-notes, and that a certain voice must play a chord consonance at each of these chord boundaries (strong beats), in his default behavior. This constraint might operate concurrently with various imitation and other structures. He is left the choice of deciding on the pitch according to the other criteria, and selecting a compatible chord, or planning a short sequence of chords and moving that voice so that the pitches fit with it. In the first case, the choice of pitch leads to a smaller set of valid chords; in the latter, the chord limits the pitch options.

My program doesn't reason about harmony in this fashion, because it adheres to the given harmony of the jazz framework. Still, the need for mutually constraining structures is apparent when we consider the problems of imitation outlined above. We seek to recognize some musical feature, and reapply it in a new context. Thus we should be able to manipulate the recognized feature (which may result from other various structures) and the procedure which enforces it in the new context, as a

single, "conceptual" constraint.

This problem has been addressed in various AI systems which use "multiple viewpoints" and "frames" [Minsky 74] as models of understanding. *Constraint systems* have been applied in various areas, from to model physical systems with mutually constraining parts, like electrical circuits (e.g. [Steele and Sussman] and [Borning]), and more abstract reasoning (e.g. [Steels]). Applications of constraints at intervallic, harmonic, and imitation structures are discussed in the description of the program.

3. The Jazz Problem

We now begin a more detailed discussion of a program that embodies these ideas. Until now we have remained concerned with issues of description rather than particular stylistic constraints. Still, I wish the program to produce music that is not utterly self-constrained and foreign. Jazz provides a nice solution to this problem. It offers a simple, fixed framework in which musicians improvise and innovate.

The jazz framework originated with New Orleans and dixieland jazz around the beginning of the century, combining the emergence of "popular tunes" with the needs of collective tonal improvisation. By agreeing in advance to stick to a well-known chord progression, the musicians plan their improvisations to fit with those chords. Sometimes in a New Orleans band, all the horns would improvise simultaneously; gradually, "soloists" have become more important, so that at any time one lead improvisor plays, while a "rhythm section" expresses the chords in an arrangement or limited improvisation. My program attempts the latter jazz form, in the role of soloist.

Typically, the soloist plays the melody of the popular tune once through, and then improvises during several repetitions of the same progression, or **choruses**. The rhythm section provides sufficient rhythmic and harmonic support that the context in the progression is always clear.

Thus for the jazz soloist, the given chords and melody play the role of "material to work with", rather than "constraints which must be satisfied". The solo must be judged on its internal *co*herence, in context, rather than its *ad*herence to a particular rule. This makes jazz a tonal framework for innovation -- and while it thereby provides a suitable domain for "non-stylistic" models, it simultaneously creates a subtle, open-ended problem. (This is one reason most "jazz theory" books can say little without analyzing a particular subdialect or soloist's work.)

The program incorporates the descriptive ideas developed above to produce single voice (i.e. "horn") solos which manifest coherence with respect to the chord progression, the given melody, and successive phrases within the solo. Throughout the discussion, elements will appear which originate in my own understanding of jazz, as a pianist and articulate listener.

3.1 The Improvisation Program

The program is divided into two sections, both of which use the ideas we have developed. First, a simple analysis made of the chords and melody. The chord progression is analysed in order to infer a plausible, concurrent progression of modes. This applies our "difference" theory in harmony, and provides an additional level of consonance against which to analyse the melody. The melody is then subdivided into phrases, which are ranked "most interesting" -first, in accordance with simple harmonic, intervallic, and rhythmic measures, as thematic material for the improvisation.

The program plays through the given melody once, a "straight" chorus, and "leads in" to the solo. Then the general approach is:

1) Choose an interesting phrase from the given melody, and produce a variation on its most

interesting features. If these features do not fully constrain the variation, use the default behavior for the unspecified description levels.

2) Repeat the process on *that* phrase -- look for interesting features in the variation; and continue this process, until it gets boring. Then begin again with another phrase from the melody.

Here, phrasing subtleties have been simplified in order to focus on other issues. The program chooses either an input phrase or the previous phrase, which ever is more interesting, according to the criteria defined below, with the addition constraint that no theme be varied more than twice. The result is a dynamic interleaving of "AABBCC..." and "ABCD..." rhythmic forms -- *not* a strict plan. This only hints at the kinds of dynamic boundaries improvisors use.

Of course, the whole algorithm is simpler than what most improvisors do. But it has two important characteristics: first, it allows us to exercise many of the abstraction mechanisms which I have argued are generally important in tonal improvisation; and, it employs a simple, sensible procedure to produce improvisation that is complex and "unpredictable" in some of the ways the random-number enthusiasts try to achieve -- without sacrificing the internal coherence of the music, or compromising the psychological model.

Now we fill in the details of the algorithm, including precise meanings for "default" and "interesting". These roughly opposite notions apply the same similarity ideas.

Each idea of similarity defined a particular mechanism for compressing musical material presented in sequence. Listeners expect the total rate of change to be small. Thus they have "defaults" by which, in addition to any other expectation, they expect the change at any level to be as small as possible. Larger changes at one level -- like leaps and certain dissonances -- make the listener wonder why; this suspense must be resolved by some musical explanation: "making it far at that level lets me make it near this way". Imitation is the most flexible kind of explanation, since it can use any of the local structures, and can be done "after the fact", resolving the suspense of the original large distance.

We can not be too single-minded about the nearness idea, though: repeating the same pitch over and over may fulfill several kinds of small difference in music -- but we also *expect* not to be bored! This subtle balance is one of the things that makes "good" music-making difficult to describe and teach. Repetition and human memory convolute our theory; even the most intricate forms go out of style. Not only are defaults tempered by defaults at other levels, but improvisors get bored, and seek material to work with that is unexpected in some way. So, in the program, a melody note is ordinarily *not* repeated; and phrases that are noticably uniform in other ways carry that as an "interesting feature".

4. Analysing the Melody

Before we begin improvising a new melody, we briefly analyse the melody and chord the progression we have been given. Our basic goals are to provide for default harmonic behavior, compatible with the jazz framework, and to collect some interesting features among the information we have been given.

The improvisor need not make a self-conscious effort to perform this analysis. I do not mean to suggest an improvisor who desires to manipulate the audience, or struggles in fear of boring them, when I say he finds interesting features to direct their thoughts.

Sometimes that description only makes sense when we remember that the improvisor is one of the audience. He may be directed mysteriously, inarticulate about his own default behavior. He may not care about the audience at all, just wandering, sifting the music through the rest in his head, playing some things over and over, the given tune acting as only a vague trigger.

Still, to whatever degree he uses the given material, he performs some analysis of it; and however complex or personal his search for interesting features is, his success with an audience will depend largely on their interest in hearing the same things and pursuing similar lines.

4.1 Inferring modes from chords

The given chords provide the harmonic framework in which to analyse the melody and produce the improvisation. In analysis, the given pitches are compared with the concurrent chord; the concepts developed above let us classify each note as a disambiguator, consonant, or dissonant with respect to that moment in the chord progression, and to determine the harmonic interval and scale degree with respect to the chord root.

Ordinarily jazz musicians also consider each chord to imply, in context, a concurrent mode which usually contains the chord's tones as a subset. This mode is a crucial aspect of the structure implied when musicians speak of "playing on a scale" or "being in a key"[1]
-- although no progression of modes was actually given. So, first the program infers a progression of modes from the chords. These will be used in the analysis of the melody and various behavior for the improvisor.

How do we infer the appropriate mode to use with a given chord? Here I employ both a basic similarity idea, and the expected sequence idea (specifically, the cadence).

We could also attempt a different sort of harmonic analysis -- searching ahead for cadences and getting clues from the melody, as students often do. (With sheet music, musicians also can get a clue

---

1. I have argued that "key" is too ambiguous to use here. Similarly, though "scale" is sometimes used, it can imply a sequence of motion as well as a set; thus the use of mode.

from the key signature, although this can be misleading.) Winograd's analysis program [Winograd 67] made a detailed treatment of this more complex, "textbook" analysis -- although, since the analysis proceeds strictly from right to left, it is not a model of the improvisation or listening.

Here as ever, my approach relies on little "stylistic" information, attempting a more general psychological approach. My simpler algorithm can be viewed as an approximation of certain listener expectations, or the kind of analysis a jazz musician makes the first time he reads through the chords of a tune. It is sufficient for the purposes of the improvisor here.

4.1.1 Nearby Modulations

First, we can divide the harmonic qualities into two basic types. **Symmetric** qualities describe the same tones when given several different roots. For instance C Dim-7th and and Eb Dim-7th both contain only C, Eb, Gb and A. The tones of a symmetric quality are therefore ambiguous in defining a particular root, "rootless". Conversely, the tones of an **asymmetric** quality define only one root for that quality, and are unambiguous in that sense. [1]

The distinction is important here, because listeners can not easily fit the tones of a chord into their framework of expectations if they are uncertain what the root is. When confronted with a symmetric quality they must often "suspend judgement", basing their harmonic expectations on the chords that preceded it. (This is one meaning of the musical term "passing chord".) The program does this.

Figure 4 shows part of a similarity network for the asymmetric modes we will be using: Major-Mode and Melodic-Minor. The criterion for similarity here is the number of common tones; only links between the "closest" modes are shown. Arcs indicate tones that distinguish similar modes.

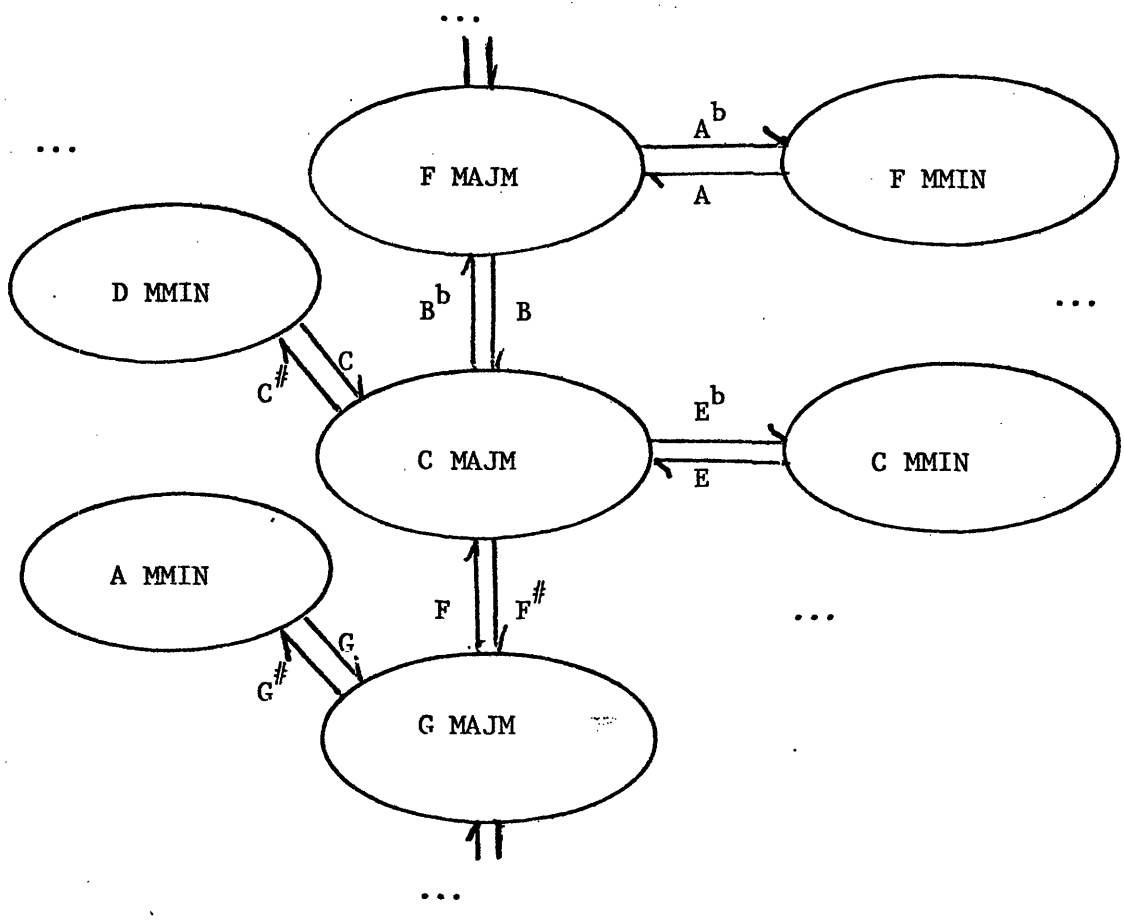
Whenever the chord changes, the minimal motion criterion is applied as follows: we remain in the previous mode unless the chord gives evidence that the mode has changed. If just one of the tones in the chord is dissonant with the previous mode, the new tone is interpreted as a disambiguator for the new mode; it directs a transition to the nearest mode consonant with that chord. The similarity network thus defines a sort of "psychological state machine", embodying one of our efforts to conceive the the progress of the sound. In much tonal music, these transitions are far more common than more disparate modulations. This is a convention, though not arbitrary -- it reflects the audience's limited capacity to predict and assimilate changes in harmony.

The table in Figure 5 is another representation of the same network. Note that various "textbook rules" are captured in this "difference" formulation. A C-Dominant-7th chord in a C Major-Mode context implies a transition to F Major-Mode; an E Major chord in the same context implies modulation to A Melodic-Minor; etc.

---

1. The *lowest* currently sounding pitch can help determine the root of an ambiguous chord; here we assume the improvisor does not control this, and will not consider it.

Fig. 4. Similarity Network for the Asymmetric Modes



The tones of the new mode must be a superset of the tones of the new chord,[1] but this may not be the case for any row in the table. If more than one modulation appears in the table, these are tried in the order shown until one succeeds; for instance, if an F Minor chord appears in a C Major-Mode context, this is incompatible with a modulation to A Melodic-Minor, so F Melodic-Minor is used. If all the nearby modulations specified in the table fail, the procedure below for "discontinuous" modulations is used.

Since the Natural-Minor mode is an inversion of a Major-Mode, the similarity networks are isomorphic -- the transitions in Figure 5 for C Major-Mode apply, transposed, to A Natural-Minor.

1. This simplifies the formulation here; I prefer to view related chords with non-modal tones as "altered" or "substitution" chords, which I do not explore here.

Fig. 5. Transitions Between Asymmetric Modes

MAJOR-MODE:

New Tone	New Root	New Quality	Example from C Major-Mode
Min-2nd	Maj-2nd	Melodic-Minor	C# ==> D Melodic-Minor
Min-3rd	Root	Melodic-Minor	Eb ==> C Melodic-Minor
Tritone	Fifth	Major-Mode	F# ==> G Major
Min-6th	Maj-6th	Melodic-Minor	G# ==> A Melodic-Minor
Min-6th	Fourth	Melodic-Minor	Ab ==> F Melodic-Minor
Min-7th	Fourth	Major-Mode	Bb ==> F Major

MELODIC-MINOR:

New Tone	New Root	New Quality	Example from C Melodic-Minor
Min-2nd	Min-6th	Major-Mode	C# ==> Ab Major
Maj-3rd	Root	Melodic-Minor	E ==> C Major
Tritone	Fifth	Major-Mode	F# ==> G Major
Min-6th	Min-3rd	Melodic-Minor	G# ==> Eb Major
Min-7th	Min-7th	Major-Mode	Bb ==> Bb Major

NATURAL-MINOR:

New Tone	New Root	New Quality	Example from C Natural-Minor
Min-2nd	Min-6th	Major-Mode	C# ==> Ab Major
Maj-3rd	Fourth	Melodic-Minor	E ==> F Melodic-Minor
Tritone	Min-3rd	Melodic-Minor	Gb ==> Eb Melodic-Minor
Maj-6th	Min-7th	Major-Mode	G ==> Bb Major
Maj-7th	Root	Melodic-Minor	B ==> C Melodic-Minor
Maj-7th	Min-6th	Melodic-Minor	B ==> Ab Melodic-Minor

Other times, a chord will imply a symmetric mode. Diminished-7th and Augmented chords are themselves symmetric; in the program they imply a Dim-Mode or Whole-Tone mode, respectively, with the same root as the chord. Again, subsequent transitions refer to the most recent Major-Mode or Melodic-Minor, rather than these ambiguous modes.

Actually, the program also infers a Dim-Mode mode during Diminished triads, too. This is justified in part by the difficulty in distinguishing similar Diminished triads in their various inversions, and corresponds with my own habits and intuitions.[1]

---

1. John Mehegan [Mehegan] considers *all* triadic chords in jazz to be "shorthand" for 7th chords. Since to him any Diminished triad is really a Diminished-7th chord, my view here is compatible with his.

4.1.2 Discontinuous Modulations

Of course, more severe modulations occur, and we must have a way of computing modes from the given chords in those cases. If two or more tones differ from the previous mode, or if the "nearby" algorithm fails, the program perceives a "discontinuous modulation": the "small difference" idea is temporarily revoked, as insufficiently powerful. In its place, the program uses a "typical progression" heuristic. To this end we elaborate a little more here on the cadence.

The cadence is one of the main harmonic conventions of tonal music. Earlier, we defined a cadence as a progression in which successive roots are related by ascending fourths. In fact, cadences are often much more structured; we now define a typical cadence as a three-chord progression, with the given root relationship, in which the chords are called the **Subdominant**, **Dominant** and **Tonic**. [1] The sequence of chord qualities in these cadences typically takes the forms: Minor, Dominant-7th, Major; or Half-Diminished, Dominant-7th, Minor; or a variant of these. This is often abbreviated "2-5-1", i.e. the 2nd, Fifth and Unison with respect to the root of the Tonic. Sometimes the Subdominant is altered or absent.

When the program encounters a discontinous modulation to an asymmetric chord, it assumes the progression is forming a new cadence in which the new chord plays one of those roles. The table in Figure 6 describes the quality of the new mode, and the role the chord plays in it, for the asymmetric chords.

Note that a discontinuous modulation to a Half-Diminished chord is shown to imply the Subdominant of a Natural-Minor mode. This is a variant of the cadence conventions, where the typical progression is 2-Half-Diminished, 5-Dominant-7th, 1-Minor. The case is interesting, because no Major or Melodic-Minor mode contains all of the tones for these chords. Many systems invent a "Harmonic-Minor" mode to accomodate this; here, the similarity heuristic will automatically modulate from Natural-Minor to Melodic-Minor, changing the appropriate defaults, when the 5-Dominant-7th chord begins.

Fig. 6. Default Modes for Discontinuous Modulations

Chord Type	Role w/r/t Mode	New Mode Quality
Major, Maj7	Tonic	Major-Mode
Dominant-7th	Dominant-7th	Major-Mode or nearest
Minor, Min7	Subdominant	Major-Mode or nearest
Half-Diminished	Subdominant	Natural-Minor

1. This is common textbook terminology, in which the final chord defines "the key". Here the use of the word Dominant reflects the fact that the middle chord is generally Dominant-7th chord.

I find I use similar heuristics when listening to an unfamiliar tune, to predict what the next chord will be, while humming or playing along. Jazz and popular progressions are usually full of cadences, so it works much of the time. Additional heuristics based on typical progressions, concerned with chord roots as well as consonance, would produce a more life-like performance.

Together, the small-difference and cadence heuristics capture important features in the harmonic expectations of tonal music. Again, while the approach here does not duplicate "textbook", "Schenkerian", or other formal analysis, it achieves its objective: the improvisation program now has concurrent melody, chord progression, and mode progression from which to build the improvisation. It also provides another example in which "switching levels" seems to be important.

Particularly, the presence of modes adds an additional level of harmonic subtlety. In the planned chord progression, the improvisor now has graded classes of harmonic nearness with which to negotiate: the **chord-consonant** tones, the 3 or 4 "nearest"; 3 or 4 **modal** tones that are not chord-consonant; and 4 to 6 **chromatic** tones, the "most dissonant" set. Also chord and mode disambiguators typically provide 1 or 2 tones of important harmonic information; these are especially at chord boundaries, and will be used to construct melodic "trajectories". Are appear as defaults when phrases are constructed from partial descriptions.

The chord progressions can also be used for melodic activity fixed in some way with respect to the roots. Note, however, that the nearness heuristic for determining the mode has no "root" component, and there is no procedure given to provide a preference for Major-Mode or Natural-Minor in the nearby modulation cases. In fact, the mode is treated entirely as a set of tones, and the Major-Mode root spelling is used by default in the analyser and ignored in the improvisor. (This approach is compared with that of [Ulrich] -- who relies on an idea of "key" -- in the Conclusions.)

4.2 Finding Interesting Phrases

In the final stage of analysis, we break up the melody into phrases, and organize them into suitable themes for improvisation. The phrase boundary problem is interesting and complex, but I have not pursued it here; some issues are outlined in the conclusions. Here I simply break up the melody at 2-measure boundaries, so the phrases are long enough to manifest patterns when features are repeated.

A simple scoring system is then used to rate these phrases, and give preference to phrases whose pitches violate the default behavior. It is not crucial that the "absolute best" input phrases be selected as themes; in fact, the improvisor need not use these themes at all. But it often helps the audience to find the most unusual elements of the melody -- its "signature" -- at the spine of the improvisation. So the program makes an effort to establish a preference order for the phrases.

Harmonically and intervalically, the most unexpected things are large distances without explanations at the other level. Since modal tones and small steps are the defaults, chromatic tones are generally "passing tones", followed (and often preceded) by modal tones a Min-2nd away. Similarly, large intervallic leaps attract attention especially if they are not arriving at a chord-consonance. As

discussed above, repeated notes attract attention in voices where motion is the default. Rhythmically, syncopations are surprises.

A phrase simply scores a point for each of these. An unresolved chromatic tone, non-chordal leap, or a syncopation will each score a point. An immediately repeated pitch scores a point, but if another repeat follows immediately, no new point is scored -- it's getting boring already. The phrases are then sorted, highest score first, in preparation for the algorithm that will use them for thematic material. The same criteria will be used to determine what phrase to vary, and how, in the next chapter.

5. Planning the Improvisation

The framework thus far provided simplifies the description of the improvisation algorithm itself. A few extensions appear in this part of the program: *trajectories* -- short term plans designed to end on a harmonically important pitch; *interest-driven imitation*, whereby the program recognizes large distances *or* utter uniformity within earlier phrases, as good, "noticable" areas for imitation; and boredom, whereby similar variations are not repeated too often.

5.1 Exploiting Features

After playing the the melody "straight" for the first chorus, the program chooses a suitable candidate phrase from which to produce the variation. Initially, the last 2-measure phrase from the initial melody is chosen; this creates a comfortable "lead-in" for the solo. After this, the phrase to vary will either be the phrase immediately past, or the next interesting 2-measure phrase in the list determined earlier. During each iteration of the program, this phrase is the "theme" from which variations will be produced.

The procedure outlined in Figure 7 is used to determine what sort of variation the program would perform on the theme. The program tests the theme for each feature shown; if any is recognized, the given action is may be performed. Since several features may be present, there are various ways to organize these. If a particular feature is strongly present (e.g. several leaps), the program might prefer a variation at that level. Many improvisors have high levels plans for exploiting particular kinds of features. I have barely experimented with these; the details of the selection procedure were contrived partly to demonstrated the various aspects of the system in the examples. This is essentially a production rule system, built on top of the constraint satifier.

Most of the variations types inherit their rhythm directly from the input phrase, "Uniform Features" being the exception. A "parallel" framework is thus created for the variation is created, letting the listener compare it readily to the theme.

Summarizing the table: If a repeated pitch is recognized, the contour of the phrase is inherited -- which means the repeated-pitch pattern is inherited, too. Similarly, if a leap of greater than a Fifth is recognized, interval size is inherited -- but *not* contour. This means that a leap up in one phrase may be followed by a leap down in its counterpart -- a common musical construction.

Fig. 7. The Variation-Making Algorithm

FEATURE	INHERITED PATTERN
Repeated Pitch	Contour
Leap > Fifth	Interval-size
Scalewise	Chord-Degree (all Modal)
Unresolved Dissonance	Mode- and Chord-consonance patterns
Uniform features	Repeat uniform features
Else	Contour

If all the pitches in the theme are modal with respect to the concurrent harmonic progression, and the steps are small -- all a Maj-2nd or less -- a "scalewise" feature is recognized. This is a good candidate for inheriting a parallel motion with respect to the chord root. We shift the variation into the new harmonic context, by enforcing the same harmonic-degree between the pitches and the new chord. This kind of variation is used in various situations, but is especially apparent when the harmonic *qualities* and relationships between roots of the chords in the theme and variation are the same.

If an unresolved dissonance is present -- using the definition given in the Analysis chapter -- the patterns of Mode- and Chord-consonance are inherited, so that a dissonances appear at the same points in the variation. Because these dissonances are so apparent to listeners, contour as well as rhythm are inherited from the original phrase, providing a strong parallel framework.

(Students trained in the European traditions may be troubled that unresolved dissonances should appear at all; but as jazz developed, innovators explored areas that were barely treated before this century. In counterpoint, and other dialects in which individual voices are the only source of harmonic information, the voices must cooperate -- controlling the progressions and remaining consistent with them are a single activity. In jazz, with the harmonic progression shared in advance with the other musicians (and often with the audience), the rhythm section leaves the soloist more harmonic freedom, somewhat paradoxically. He acknowledges the progression, but may coherently deviate from it in ways that become his signature.

Such harmonic experiments characterized much of the "bebop" era: Charlie Parker and Bud Powell [Mehegan] would build an entire solo on a repeated, unresolved dissonance; Thelonious Monk [Monk] would include dissonances, or substitute Whole-Tone modes in places no one else had tried; etc. The melodies these beboppers wrote began to include these elements, in a fashion that my program automatically exploits -- unresolved dissonances in the given melody are identified as interesting, and used to construct more "beboppy" solos.)

If none of those tests succeed, the program seeks "general features" whose imitation might be helpful in keeping the solo coherent. These take the opposite form of the earlier notions of "interest", in that they measure *uniformity* over the whole phrase, rather than large distances at a particular note. These are:

All pitches chord-consonant

All intervals the same size, within a Min-3rd

Uniformly ascending or descending.

If one of these is recognized, the variation is constrained to inherit the same feature. Either of the first two may be inherited, but not both; they are tried in the order shown. If the phrase is uniformly ascending or descending, this is always inherited. This arrangement ensures that inconsistent combinations of descriptions will not arise.

Since the uniformity features are inherently non-local, there is not as great a need to set up a "parallel context" for each note in the variation. Instead of inheriting the rhythm, a nearly uniform rhythm is chosen, at about the same tempo. (The last note duration may be extended to yield the same time total time for the phrase.) This is the only form of rhythmic variation currently used.

If none of these features are recognized, the rhthym and contour of the original are still inherited. Still, the recognizers need not "cover" the space of possible features as well as these do; if none is recognized, the program will always proceed with its default behavior, described in the next section.

5.2 Negotiating Defaults

Even if features are recognized and variations arranged, the pitches are not ordinarily fully constrained by their descriptions to a particular pitch value. A typical description might be "not chord-consonant, modal, and within Min-3rd of the previous pitch". Now we describe a procedure for choosing a particular pitch from this description.

The procedure is a simple combination of the defaults we have discussed thus far, in which we sucessively "narrow down" the set of valid pitches, until we are left with one. First we exclude any pitches that are incompatible with the harmonic and intervallic descriptions we have been given; this requires that we know already the harmonic context (i.e. the time), and the previous and/or next pitch -- hereafter, the "neighbor" -- to satisfy an intervallic relation.

Next the program tries successively to eliminate candidates, as follows:

Leaps larger than a Fifth (with respect to the neighbor);

Pitches close to the edge of the instrument range;

The neighbor pitch itself (i.e. avoid repeated pitches);

[Non-disambiguators -- see next section]

Leaps larger than a Maj-3rd;

Pitches dissonant with the mode;

The neighbor's *other* neighbor, if known;

Leaps larger than a Maj-2nd;

Pitches dissonant with the chord;

Leaps larger than a Min-2nd.

This procedure whittles away the options, simultaneously avoiding large leaps, non-modal tones, and local repetitions -- as long as this is consistent with the given descriptions. A typical horn instrument range restricts the improvisation to a few octaves; one of the defaults tries to rule out pitches that are in the upper or lower quadrant of the range. The argument for avoiding the pitch of the *neighbor's* neighbor is the same as the one for avoiding repeated pitches -- repetitions constitute a "feature", even if they are not immediately local. Here, this default prevents the program from bouncing back and forth between a pair of near pitches -- like the Root and 7th of -7th chord -- when few other constraints have been given.

All of the defaults shown are tried, successively constraining the pitch options. If any of these would result in *no* options, it is ignored, and the next is tried. We see the default schedule gradually going to increasingly constrained sets of harmonic and melodic options. The precise values were chosen in an effort to achieve a balance of chromatic motion, scalewise motion, and arpeggios, the way so many tonal dialects do. Of course each dialect really has a specialized system of defaults.

When the option-restriction has been performed, there may remain more than one pitch option left. The program chooses the one intervallically nearest to the neighbor. If there are two pitches that near, it chooses the lower one -- though I make no justification for this. Note that the algorithm is not very complicated, lacks noise sources, and, with that one exception, is motivated by the theory.

5.3 Short plans: Trajectories

Having chosen a rhythm for the phrase, constraints on each pitch, and a way of mixing given constraints with the various defaults, we now consider the order in which values for the pitches are chosen. I use two simple schemes here: chronological selection, and reverse-chronological planning with respect to a "trajectory" note.

In the first case, pitches are chosen from their descriptions left-to-right. In human improvisation, this corresponds to some situations in which a remembered phrase is being repeated verbatim, or modulated -- once the improvisor plays the first note, the other decisions follow naturally from it. In more explorative improvisation, I sometimes construct phrases this way, but somewhat slowly -- ordinarily I do not proceed without an idea of where I am going.

The latter case captures a level of planning that seems common to many kinds of improvisation. The improvisor decides where he wants to *end up*, and contrives the phrase to get him there. Here the other descriptions of the phrase constitute a partial plan obtaining the trajectory objective.

I have implemented these plans simply by selecting the pitches for the phrase reverse-chronologically, last pitch first. Improvisors are not generally fluent in reversing time this way for more than a couple of notes; they use remembered cliches and more complex strategies. Still, this captures some realistic musical reasoning, especially the kind of consideration one makes when learning a new solution, or constructing a long plan from several short ones.

In the program, the chord and mode disambiguators are the "important notes" the trajectories reach for. The 2-measure phrases are extended so the last note of the phrase is the first note of a new chord. The defaults in 7 have been devised to prefer disambiguators, as shown, when the note occurs at the beginning of a new chord. When the trajectory destination is being determined, it has no immediate neighbor yet, before or after. The last pitch in the previous phrase is used to govern the default behavior. This makes some musical sense; often a sequence phrases will be organized so that the relationships between their final pitches are constrained.

This use of disambiguators models the kind of improvisation in which the soloist wishes to contribute substantially toward outlining the harmonic plan, perhaps giving the rhythm section more improvisatory freedom. It also suggests kinds of constraints necessary in counterpoint, etc., where single voices cooperate to realize the harmonic plan.

5.4 Examples

Two examples demonstrate the program's behavior. The first is informative because it uses *only* the default decision-making; the other performs variations as per the description in Chapter 4. Both are improvisations on the New Orleans tune "Basin St. Blues". This 8-bar tune provides a brief demonstration in each area.

Figure 8 shows the original tune. The traditional key signature, F Major, has been left out, and only flat-spellings are used for accidentals. The boundaries above the staff show the 2-measure phrase

boundaries.

The figure also shows the effects of analysis. The modes shown below the chords correspond well with the modes I would choose for default improvisation. (Here MAJM, MINM, and DIMM refer to MAJOR-MODE, MELODIC-MINOR, and DIM-MODE, respectively.) The numbers above the phrases indicate their relative "interestingness" ranking. The phrases with both a repeated note and a syncopation win out. In fact, since no other database of phrases has been developed, *all* the input phrases are eventually used as themes in the examples.

Fig. 8. "Basin St. Blues": Input and Analysis

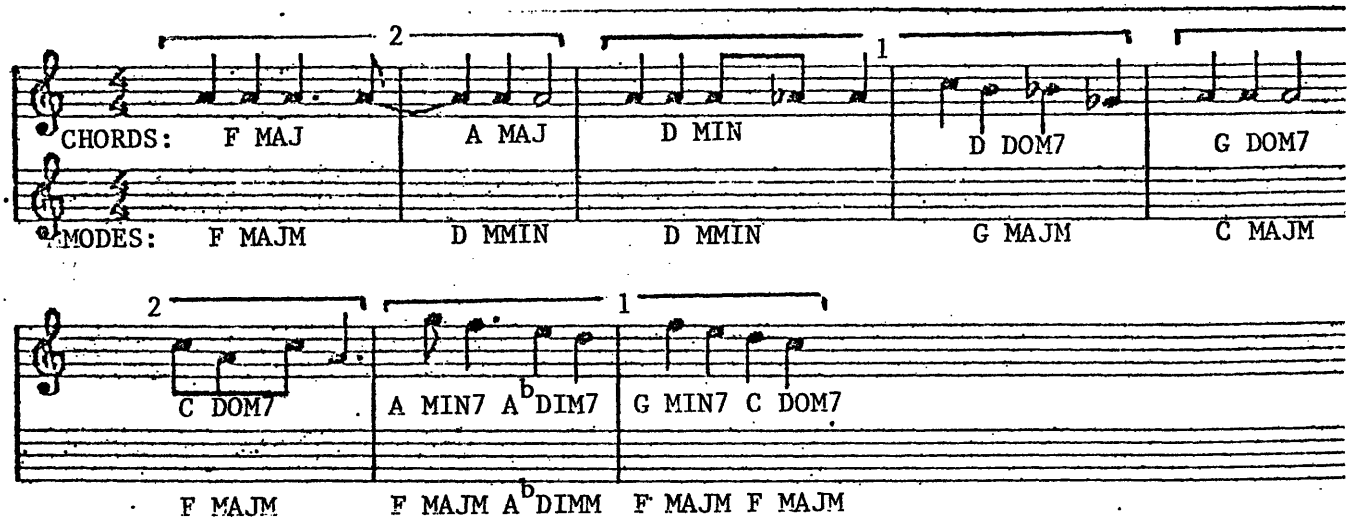


Figure 9 shows two choruses of improvisation by the system, using only the default behavior. Phrases here inherit only rhythm from their "parent" themes in the original melody. Following one chorus of the original tune, the first phrase copies the rhythm of its last 2-measure phrase. Variations on this phrase are not "trajectories" because the program only constructs those from prototype phrases that have a successor phrase. After this variation, the improvisor begins inheriting rhythms from the first elements in the sorted list of thematic material, seeking trajectories. No effort at "development" or "ending" was made; the last moments of both solos are padded with quarter notes.

Observe that the path toward the trajectory, from two measures prior, often begins with a leap. This is because in the strict reverse-chronological ordering, default intervals between the first note in the trajectory and the previous note are not observed. However, the suspense created by the leap is largely resolved in the "explanation" provided by the arrival at the disambiguator. The strong harmonic function of the last note gives the whole phrase a sense of purpose -- especially since the notes along the way are generally small steps between modal tones. In fact, the leaps become a

repeated theme throughout the solo, thus offering their own explanation for themselves! Very crudely, this shows how by choosing certain problems, and discovering classes of solutions, musicians develop personal, self-constrained styles.

Observe that the defaults respond differently in different contexts. Typically the algorithm results in scalewise motion, but the soloist may repeat a note or change direction to reach a nearby consonance or disambiguator. Thus interactions between the default algorithm, the inherited rhythm and the harmonic plan define a geometry that seems unpredictable to listeners -- even us, who know the algorithm. It is easy to see why human musicians become mysterious so quickly.

Fig. 9. Improvisation using Defaults

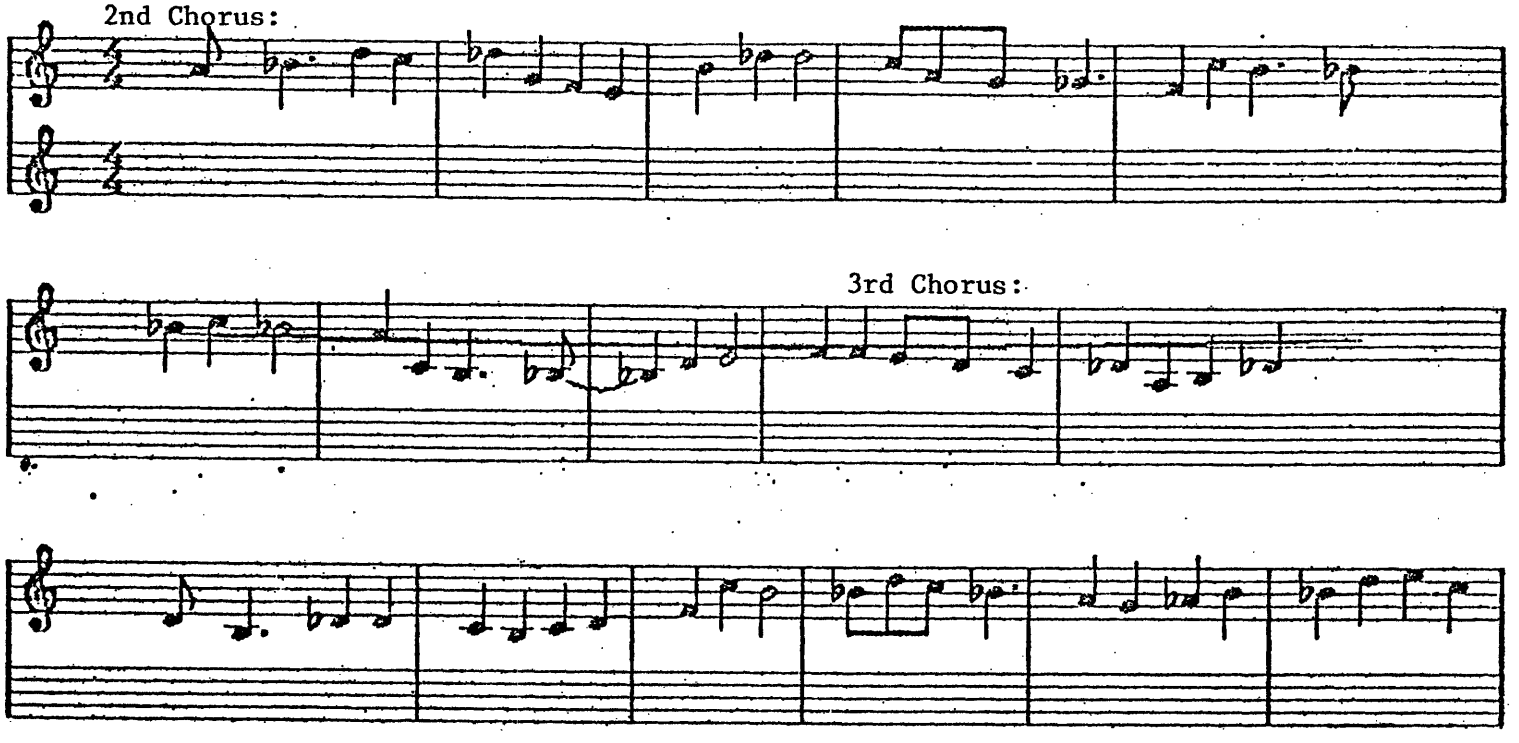
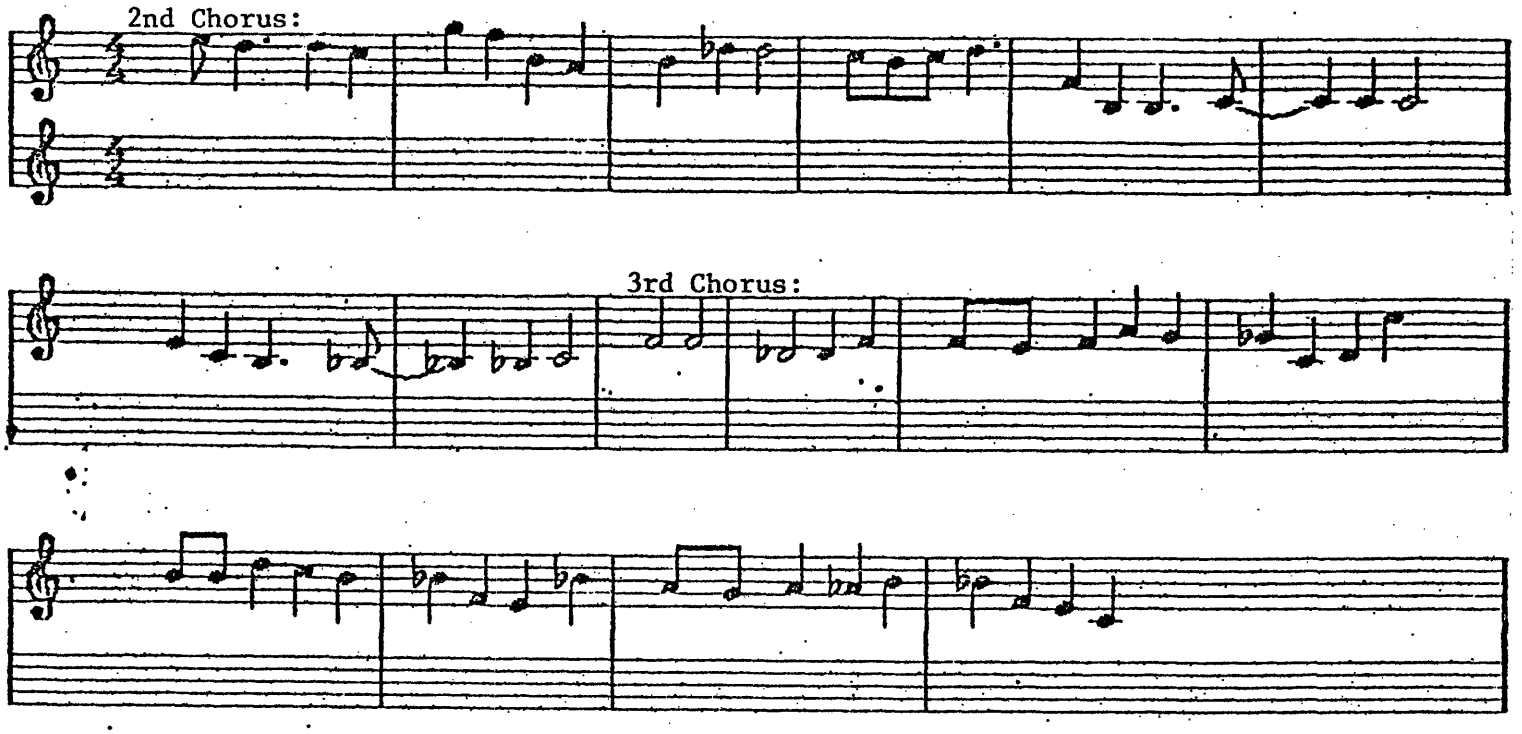


Figure 10 shows improvisation on the same tune, this time with feature recognition and imitation engaged. The same analysis is used, but now patterns are inherited from phrases in the original melody, or immediately preceding ones. The print-out in figure 11 shows which, and what kind of variation was performed. Oddly, while this solo demonstrates the system's variation-making capabilities, it does not sound as "thematic" as the previous one. My effort to show variations on different features in several themes in a brief space produced a less coherent solo. Even so, repeated notes were not explicitly repeated, and unresolved dissonances never appeared (essentially since

Fig. 10. Theme and Variation



Basin St. Blues does not contain any). Human soloists have a much larger vocabulary of features to imitate, *and* use them more judiciously; a fine solo might apply only a few.

5.5 Implementation

The program was implemented on [Lisp Machine] personal computers at the MIT AI lab. The Flavor message-passing system was used for its advanced data abstraction capabilities, which simplified some problems of data structure definition. For instance, I constructed an EVENT-SEQ data type, with generic parts and operations, and then specialized it into PROGRESSION and PHRASE types.

Below I outline some of the important data structures. Figures 12 and 13 show pseudo-LISP structure definitions, edited for readability, that reflect levels of description within the system.

First we treat the Note structure. A Note is a pitch event, occuring in time. Each symbol in the structure represents a constraint which may be in force on the selection of a particular Note. They take different forms: in the Note definition, the PITCH slot, if known, offers more information than the other slots; it precisely defines the pitch itself. Conversely, the harmonic data structures only

Fig. 11. Variations in the example

```
Variation on the last phrase of first chorus.
Inheriting scale degree with respect to chord root.
Approaching trajectory.

Variation on an input melody phrase.
Inheriting interval-size pattern, with leap.
Approaching trajectory.

Variation on an input melody phrase.
Inheriting scale degree with respect to chord root.
Skipping first note -- already known.Approaching trajectory.

Variation on the previous phrase.
Inheriting interval-size pattern, with leap.
Skipping first note -- already known.Approaching trajectory.

Variation on the previous phrase.
Inheriting features: Whole phrase will be chordal
Inheriting overall tempo.
Skipping first note -- already known.Proceeding left-to-right.

Variation on an input melody phrase.
Just inheriting contour, same rhythm.
Skipping first note -- already known.Approaching trajectory.

Variation on the previous phrase.
Inheriting interval-size pattern, with leap.
Skipping first note -- already known.Approaching trajectory.

Variation on the previous phrase.
Inheriting interval-size pattern, with leap.
Skipping first note -- already known.Approaching trajectory.
Approaching trajectory.

Trailing quarter-notes....
```

---

offer partial descriptions of the pitch, even when the chord and mode are known, since they use the classifying -CONSONANT?, -DISAMBIGUATOR?, and -DEGREE descriptions we discussed.

The CHORD-DEGREE constraint refers to the degree of the harmonic interval of the pitch with respect to the root of the chord or mode. Together, the chord, mode, CHORD-DEGREE, and MODE-CONSONANT? constraints fully describe the harmonic character of the pitch, its tone. For instance, in a C Major chord, values of 3rd for CHORD-DEGREE and NO for

Fig. 12. Data structure for Note

```
(DEFSTRUCTURE NOTE
        PITCH
;;The next group of elements are abstract descriptions
;; of the PITCH.
                                ;Harmonic constraints w/respect to chord:
        CHORD-CONSONANT?        ;YES or NO
        CHORD-DISAMBIGUATOR?    ;YES or NO
        CHORD-DEGREE            ;Unison, 2nd, 3rd, ... 7th

                                ;As above, w/respect to mode.
        ON-MODE?
        MODE-DISAMBIGUATOR?

        INTERVAL-WITH-RESPECT-TO-PREDECESSOR        ;Intervallic constraint.

        INSTRUMENT              ;Minimum and maximum range constraint.

;;Next, the temporal elements.
        ONSET-TIME
        DURATION
;;Finally, the related structures this Note will use...
                                ;...for harmonic constraints:
        CHORD                   ;The current chord, from the given progression
        DELTA-CHORD             ;Current chord disambiguators.
        MODE                    ;Inferred from the chord progression.
        DELTA-MODE              ;Mode disambiguators.

                                ;... and intervallic constraints.
        PREVIOUS-NOTE
        NEXT-NOTE)
```

MODE-CONSONANT? together imply Eb. (Unison and Fourth degrees are always consonant.) Although its definition is not shown, the HARMONIC-DEGREE constraint is implemented as a data structure similar to Interval below.

Because of the chromatic pitch domain, I was able to implement pitch constraints uniformly, as "filters" that retain or discard elements from an initial, finite "set of all pitches". When the Instrument is selected, this restricts the range of pitches, usually to a few octaves. (This is in fact the only use of the Instrument structure by the program; no attempt is made to simulate mechanical constraints.)

Since the program via "narrowing of options" in accord with related data structures, the use of constraints here is more like Waltz's scene-analysis solution [Waltz] than like the quantitative realms

in which some recent constraint systems (e.g. [Steele&Sussman] and [Borning]) have been applied. After a new description is added, the data structure may be given various, explicit "relaxation" commands -- to return a list of pitches remain valid for this description, choose one according to the default algorithm, or complain if the description is inconsistent. Thus propagation and consistency checking are not actually performed by "demons" here, for efficiency.

There are actually underlying mechanisms by which the system may perform a non-local exit upon "realizing" that a description is inconsistent. These could be used by "bail-out" procedures -- improvisors usually have cliche solutions to unexpected situations -- or backtracking mechanisms, in a non real-time composer model. In my experiments I avoided these problems by being careful about which kinds of descriptions I mixed; the consistency-checking exits were useful only for debugging.

Observe that the rhythmic descriptions are simple, without overlapping partial descriptions: just an onset time and a duration. In fact, the system recognizes features like beat strength and syncopation, but avoids some subtle problems of enforcing abstract temporal descriptions on isolated notes. Recall that most important rhythmic concepts, like tempo and syncopation, describe relationships between two or more events.

Here the rhythms were always chosen first; this simplified the other problems. The concurrent CHORD, MODE and disambiguator sets are easily determined from the temporal context. If the Note is syncopated, the harmonic context for the *end* of the Note is used, so that the pitch can anticipate the harmony in the manner described in the **syncopation** definition.

Once the PITCH is known, any empty slots may be automatically filled; now they serve as descriptions rather than constraints -- though they may be used to constrain a variation. The harmonic descriptions can be determined from the CHORD, MODE, etc., while the melodic ones require that the predecessor be known.

The Note's chief melodic constraint is a complex data structure, the INTERVAL with respect to its predecessor in the same voice. As with the PITCH, the first element in the structure completely describes it, while subsequent elements are abstractions or partial descriptions. The MIN-SIZE and MAX-SIZE abstractions allow flexibility, permitting descriptions like "bigger than a Minor-2nd but not as big as a Fifth", while the DIRECTION may be manipulated independently. (A value of Same for the DIRECTION biconditionally implies a SIZE of 0, of course.)

Phrases contain elements for the uniformity features described earlier. Like the Notes, these may be given, or propagated once the details of the Notes have been determined -- the notes in the NOTE-LIST inherit properties from the phrase as a whole, and vice versa. An exception is the TEMPO-RANGE, which cannot be inherited Phrase-to-Note, since Notes have no abstract rhythmic descriptions. An intermediate tempo can be inherited uniformly, however.

Phrases and Intervals interact as follows: when the top-level plan dictates a trajectory, the appropriate flag is set in the Phrase. Then, when the Phrase relaxes, the procedure that determines the pitch is constrained by the NEXT-NOTE, and the corresponding INTERVAL-WITH-RESPECT-TO-PREDECESSOR, if that bit is set.

Fig. 13. Interval and Phrase Structures

```
(DEFSTRUCTURE INTERVAL
        DISTANCE                ;Signed distance in chromatic steps;
                                ; between origin and compared pitch.

        SIZE                    ;Absolute magnitude of DISTANCE.

        MINIMUM-SIZE            ;Abstractions of SIZE:
        MAXIMUM-SIZE

        DIRECTION)              ;UP, DOWN, or SAME.

;;-------------------------------------------------------------

(DEFSTRUCTURE PHRASE
        N-OF-NOTES              ;Number of notes within it.
        NOTE-LIST               ; Them.

;;The next several descriptions summarize descriptions of each contained Note.
        INTERVAL-RANGE          ;Minimum and maximum values for
                                ; the predecessor-interval SIZE.
        TEMPO-RANGE             ;Min and max values for DURATION.
        CHORDAL?                ;YES if all chord-consonant:
        MODAL?                  ;YES if all mode-consonant.

        INSTRUMENT              ;One for the whole voice, of course.

;;The next two are only meaningful if this phrase is a variation on another.
        VARIATION-ON            ;The original phrase.
        MAPPED-FEATURES                 ;If the phrases have the same N-OF-NOTES;
                                ; this indicates features inherited
                                ; "one-to-one" between Phrase Notes.
        TRAJECTORY?)            ;Are Pitches selected sequentially,
                                ; or planned ahead?
```

I considered embedding the program in an existing constraint language. Unfortunately, recent experiments with these languages have not yet produced a "practical" version -- particularly, concepts like "Set" and "Sequence", so prevalent in music, are weakly handled, if at all. Then again, this is also true of LISP! More on this in the Conclusions.

Various utilities were developed to make building and debugging music programs possible. I developed a microcomputer-based clavier / polyphonic synthesizer system, with a software interface to the Lisp Machine and a simple editor. These enabled me to enter melodies and chords of various

songs, and to listen to the improvisations in their harmonic context. The music notation in the figures was generated using a program developed for the Lisp Machine by Bill Kornfeld, Matt Ben-Daniel, Keith Sawyer and Bob Kerns; an interactive music editor that uses these tools is currently under construction.

6. Conclusions

I have presented representations of various tonal music concepts, and applied them to simple problems of analysis and production in jazz improvisation. Various common ideas of harmonic compatibility were realized: between isolated tone pairs, chords and modes in sequence, and between melody and harmony; as were fundamental descriptions of intervals between notes. Abstractions -- partial and approximate descriptions -- permitted rich areas for comparison between objects. These were combined into temporal patterns, so that similar phrases could be constructed from a prototype at various levels of abstraction. Uses of defaults and simple planning were also developed.

I would feel inadequate and silly if I tried to show, by means of a few examples, that a vast range of tonal dialects employ harmony, intervals, repeated patterns, and plans. These kinds of observations have been made articulately and at length by many tonal musicologists. Certain harmony concepts, common patterns, and solutions to classical problems have become the basis of most musical pedagogy. Planning is evident in the left-branching grammars of various Schenker-like formulations.

Here, a different viewpoint was fostered by experiments with new tools. A left-branching grammar may capture some aspect of musical structure, but it tells us little about how a musicians goes about constructing it, less about when or how a listener would recognize it, and nothing about why we would want to use that structure in the first place. Here we motivated the description system with theories of what listeners notice and remember, and used defaults -- crucial embodiments of what listeners expect -- in ways earlier approaches barely suggested. This permitted construction of a crude but relatively plausible improvisation model.

6.1 Related Work

The view I have taken here closely resembles those in [Meehan]; his paper is an excellent summary of how modern AI techniques might be applied to tonal music.

[Ulrich] has also treated the jazz problem using both analysis and synthesis components. His analyser is more sophisticated in some ways, including procedures to glean the names of chords from chord inversions. His program includes versions of several of the ideas here, including a left-to-right analysis: he observes (and I concur) that in most jazz progressions, local harmonic structure provides a stronger analysis than a global harmony hierarchy.

Like many students, Ulrich's program analyses the entire piece in terms of keys involving a mode more significantly tied to its root. *All* chords are treated as versions of the cadence chords -- Subdominant, Dominant, or Tonic with respect to the root. This leads to a much more complicated analysis, in which many chords with modal tones have to be explained as substitions for cadence chords (e.g. in C Major-Mode, E Minor is a substitute for C Major), etc. -- all so the final analysis could continue to infer a static "key" in these situations! In my formulation, this is inferred simply by the absence of a mode disambiguator in the chord.

Ultimately, Ulrich's complex analysis is used only to determine modes to play on during the improvisation. He uses no notion of default; the tune's melodic phrases are simply modulated onto

the new modes -- as when variations inherit "harmonic degree" in my system. Minor tonalities are not handled. The improvisation produced is very simple.

Here, by decomposing the "key" concept, and motivating the analysis in accordance with its eventual use, and using defaults, I achieved both a simpler, more psychologically plausible analysis, and a more flexible system. Still, in many ways my approach is similar in spirit to his.

6.1.1 "Music Theory"

Various attempts have been made to generalize about structures in tonal music. Most of these have avoided psychological theories, producing catalogues of common, "acceptable" structures in the form of rules. The scientific failure of this approach has become strikingly clear during this century: the explosion and rapid evolution of jazz and popular forms -- for scientists, rich evidence for comparative study -- has been ignored by many students invested in the earlier "theory". It would be difficult to maintain those systems, or accomodate the new forms in them, without first taking a scary step into psychology; most did not. As a result, only the basics of "modern" textbook harmony (e.g. [Forte]) are readily applicable to jazz.

Pedagogically also, the decision by so many to divorce music understanding from psychology was apparently a mistake. The declarative rules -- the "what" of harmony -- would be more useful in procedural contexts describing *how* and *why*. Instead, many students learn complex, weakly supported rule systems, only to conclude that counterpoint improvisation is impossible, and innovation illegal. Explanations in terms of cliches, defaults and expectations would help with these problems.

Such rules have themselves been subjects of symbolic automation, with interesting results. Longuet-Higgins's effort to reproduce typical solutions to textbook counterpoint problems[1] is a case in point. Of course, if the textbooks truly provided an unambiguous procedure for generating interesting counterpoint from a theme, the problem would be straightforward. They do not. Longuet-Higgins discovered that his best efforts to translate the given harmony rules produced dull, awkward counterpoint; but by writing a program and listening to the results, he was able to invent new rules that improved the results considerably.

This demonstrates the computer's value in helping us articulate structural features in musical designs. Few musical areas have been systematized as extensively as classical counterpoint; a huge body of literature has accumulated, much of which analyses similar music in slightly differing terms. But without a way to automate these systems, it has been difficult to apply them. Students have relied on their own (often inarticulate) knowledge of how things should sound, and accepting the textbook rules as rough guidelines -- lest they produce awkward counterpoint themselves.

Longuet-Higgins has shown that as new tools for experimenting with these systems are become more

---

1. Personal communication.

prevalent, we can begin making practical formulations, building on and debugging earlier ones. I suspect other discoveries about what makes good counterpoint will emerge.

6.1.2 Noise

Various other composition systems -- including [Fry], [Hiller], [Moorer], my own work with Tenzer, and many others -- have used pseudo-random noise in efforts to increase the apparent complexity of their compositions. My program demonstrates that these uses of noise are unnecessary, and can even stifle progress in representation.

Certainly there are conflicting goals, confusion and errors in musical decision making. But we can not make progress in understanding these if we model them as uniform thermal effects. Certainly it is dangerous to confuse noise with ideas of similarity, difference and boredom -- when models of the latter can be readily produced.

Fry's multi-voice improvisor -- the most interesting "random" jazz I have heard -- shows typical problems with this approach. Noise in the pitch selection process for a voice *tends* to avoid repeated pitches, but since this is unreliable, they appear occasionally without apparent reason -- why not avoid them directly? Similarly, randomly chosen consonant tones in several voices tend to express the chord; but occasionally an ambiguity will result from a missing disambiguator -- Fry's system lacks this concept, too.

With so many representation problems still unexplored, it is disturbing to see so many programs use noise to achieve complexity. "Markov methods" continue to appear, often without any explanation of what the noise is supposed to model. Meanwhile, representations of *disambiguator, contour,* and *trajectory* -- which seem so basic to all kinds of tonal music -- have not appeared previously in the in the synthesis literature. My refusal to use noise forced me to develop subtler structures.

There may be several reasons for the dependence on noise. Minsky [Minsky 81] suggests that reluctance to introduce cliches, boredom, and psychological elements into music analysis has been due to their basic, informal inelegance -- analysts would suffer "mathematics envy". Similar feelings could fuel desires to divide things neatly into "noise sources" controlled by "correctness rules".

Other explanations include practical experience with inarticulate behavior. When a listener can not explain why he thinks one melody is more interesting than another, what do we model? Usually something complex is going on. The measures I have used are a crude first approximation, though they are clearly a better try than random choice. Perhaps others have been embarassed to propose that listeners count syncopations and leaps in making these decisions; I view these as necessary, important steps toward a realistic theory.

6.2 Further Research

This thesis suggests many further ideas, most of which would be direct extensions of the ideas developed here.

Analysis could be enhanced in many ways. The use here of important features in a phrase is too gross; each phrase can be described in terms of the different important features for *each* note, or short sequence of notes, within it. This way a variation could inherit details of the theme more faithfully. This would be a straightforward extension of the present system.

Decomposition of melodies into phrases is a subtle problem involving the various levels of description. Here we used simple metrical and chord boundaries; but long notes and Rests, tempo changes, interval direction changes -- interruptions of uniformity at any level -- can characterize an apparent boundary. Phrase decompositions at these other levels would quickly enhance the rhythmic variety of the present system.

Recognizing repeated patterns at the various levels is one of the most difficult problems in this sphere. Limiting the search to options that include patterns that tonal listeners tend to recognize would illuminate alot about how we listen and think about music.

Musicians solve problems by controlling rhythm in ways I have not explored here. In phrase realization, tempos are varied, or syncopations introduced, in order to realize strong constraints on the pitches. This important factor has not remains to be treated in the synthesis literature.

Multiple voice experiments would provide much more vivid exercise for the harmonic theory. Disambiguators are especially important here. The same dimensions of harmonic similarity should be useful in generating *passing chords* -- inserted chords that are "near" both their predecessor and the successor in some way. This could also be combined with the trajectory notion to construct original harmonic plans. Aspects of "key" that constrain harmonic progressions would become important here.

I have barely touched upon problems of cliche, style, and heuristic search. The phrases constructed here were relatively unconstrained; solutions to a description could be found through local relaxation and solution of the "best" default answer. Gradually, improvisors develop a vocabulary of solutions to more severely constrained problems; for example, given the *first and last pitch* of a phrase, and other descriptions (e.g. modal, strictly ascending, etc.) we construct a phrase that satisfies them by varying rhythm, repeating notes, etc. -- anything that is not already specified in the problem, as we did in the case of single notes.

We might collect a basis of procedures for determining whether a particular phrase description is overconstrained, and for finding a solution if it is not. Valuable heuristics for this sort of procedure would include finding the most constrained notes and building the solution around them -- generalizations of the simple chronological and trajectory schemes used here.

However, most improvisors seem to learn many standard solutions that are compatible with the "style", and solve particular problems by modifying pieces of them. Finding such solutions would

entail building a fairly dense feature-indexed database of stylistic cliches. These would be organized in a similarity network designed to make it easy to find and modify a phrase "near" the one specified in the problem. This network would include the harmony network discussed, but would be much more complex -- along the lines of the difference networks discussed in [Winston]. The features would include all of those treated here.

6.3 Constraint Languages

Constraint languages allow data relationships be represented as such, organizing the various procedures that enforce them. This idea has proven useful in AI, where multiple viewpoints and reasoning about sparse data are so important. These languages also offer a valuable principle for programming in general, since program behavior and implementation decisions are specified in distinct modules.

Recently, several efforts to produce "general purpose" constraint languages have raised hopes. [Borning], [Steele] and [Steels] developed systems that handled new domains, but which are still unusable for most practical work. The problem is not one of efficiency, but of abstraction mechanisms: most importantly, they have no "multi-object" or "Plural" abstractions -- groupings of objects of the same type, like Sets, Sequences and Tables, such as are conventionally implemented using lists or arrays.

These entail subtle problems. Much recent work on program understanding (the Programmer's Apprentice group: [Rich, Shrobe, Waters]) concerns the various complex ways Plurals are implemented in conventional languages. In a constraint language the problems are even worse. How do we implement constraints like "One (or N or Some) of the notes in this phrase is syncopated"? This is easy to measure if the durations of all the notes are known; but how do we enforce it while constructing a phrase from scratch? Recognize a conflict? Problems like these also made general phrase level abstractions difficult to build.

Another weakness in existing constraint languages concerns quantitative inequalities -- i.e. Greater-Than. The system in [Steele] has some provisions for these -- but to use them in planning phrases, an understanding that these relationships are transitive would be important; his system lacks this, and other symbolic reasoning utilities.

But these are not music problems! Pursuing them takes us outside the domain of this thesis. Practical constraint languages -- and programming languages -- need general procedures to handle ideas like Less, Some, Successor, Delimiter, and Permutation in the flexible ways we think about them. This is not simple, but to me it seems a straightforward extension of the existing constraint languages: adding various Plurals and other types, and building generic procedures for them as language primitives. That is another story.

7. Bibliography

[Borning]
Borning, Alan "ThingLab: A Constraint-Oriented Simulation Laboratory", Xerox Palo Alto Research Center (1979)

[Cannel&Marx]
Cannel, Ward and Marx, Fred "What music is and how to make it at home", Doubleday & Co. Garden City, New York (1976)

Laske, Otto. "Musical Semantics - A Procedural Point of View", Utrecht State University, Utrecht, Netherlands (1973)

[Forte] Forte, Allen. Tonal Harmony in Concept and Practice, Holt, Rinehart and Winston, Inc.

[Fry] Fry, C. "Computer Improvisation", Computer Music Journal Vol.4 #3 (1980)

[Hiller&Isaacson]
Hiller, L.A. and Isaacson, L.M. "Experimental Music", McGraw-Hill, New York (1959)

Levitt, David and Tenzer, Michael. "A Computer Simulation of Jazz Improvisation" AI term project report, Yale College (1976)

[Lerdahl and Jackendoff]
Lerdahl, F. and Jackendoff, R. "Towards a Formal Theory of Tonal Music", Journal of Music Theory, Vol. 21, No. 1 (1977)

[LISP Machine]
Weinreb, Daniel and Moon, David. LISP Machine Manual, Third Edition. MIT AI Laboratory (Cambridge, March 1981)

[Martin]Martin, James G. "Rhythmic (Hierarchical) versus Serial Structure in Speech and Other Behavior" Psychological Review Vol 79 No. 6, 487-509 (1972)

[Meehan]
Meehan, James R. "An AI Approach to Tonal Music" Computer Music Journal Vol.4 #2 (1980)

[Mehegan]
Mehegan, John "Tonal and Rhythmic Concepts" and "Jazz Rhythm and the Improvised Line" Watson-Guptill Publications, New York (1962)

[Minsky 74]
Minsky, Marvin. "A Framework for Representing Knowledge", MIT Artificial Intelligence Laboratory Memo No. 306, Massachusetts Institute of Technology, June (1974)

[Minsky 81]
Minsky, Marvin. "Music, Minds and Meaning", MIT Artificial Intelligence Laboratory Memo No. 616, Massachusetts Institute of Technology, March (1981)

[Moorer 72]
Moorer, James A. "Music and Computer Composition", Communications of the ACM, Vol. 15, No. 2, February (1972)

[Piston]
Piston, Walter. *Principles of Harmonic Analysis* E.C. Schirmer Music Co., Boston, 1933.

[Rader] Rader, Gary M. "A method for composing simple traditional music by computer", Communications of the ACM, Vol. 17, No. 11, November (1974)

[Rich, Shrobe, Waters]
Charles Rich, Howard E. Shrobe, and Richard C. Waters
Computer Aided Evolutionary Design for Software Engineering
MIT-AI Memo 506 (1979)

Schenker, Heinrich. "Harmony", edited by Oswald Jonas, MIT Press, Cambridge, Mass. (1954)

[Shrobe]
Shrobe, Howard Eliot "Dependency Directed Reasoning for Complex Program Understanding", revised doctoral thesis. MIT-AI Technical Report 503 (1979)

[Stallman & Sussman]
Stallman, Richard M. and Sussman, Gerald J. "Forward Reasoning and Dependency Directed Backtracking in a Program for Computer Aided Circuit Analysis", *Artificial Intelligence Journal*, October 1977

[Steele]
Steele, Guy L. "The Definition and Implementation of a Computer Programming Language Based on Constraints." MIT-AI Technical Report 595 (1980)

[Steele&Sussman]
Steele, Guy L. and Sussman, Gerald J. "Constraints", MIT Artificial Intelligence Laboratory Memo No. 502, Massachusetts Institute of Technology, Cambridge (1978)

[Steels]
Steels, Luc "Reasoning Modeled as a Society of Communicating Experts", revised Master's thesis MIT-AI Technical Report 542 (1979)

[Ulrich]
Ulrich, "The Analysis and Synthesis of Jazz by Computer", Proceedings of the Fifth International Joint Conference on Artificial Intelligence, Vol. 2, Department of Computer Science, Carnegie-Mellon University, Pittsburgh (1977)

[Waltz] Waltz, David L. "Understanding Line Drawings of Scenes with Shadows" in "The Psychology of Computer Vision", Winston, ed. McGraw Hill (1975)

[Winograd 68]
Winograd, Terry "Linguistics and the Computer Analysis of Tonal Harmony", Journal of Music Theory Vol. 12, No. 1 (1968)

[Winston]
Winston, Patrick. Learning Structural Descriptions from Examples In: P. Winston (ed.) The Psychology of Computer Vision. New York: McGraw Hill (1975)