THE ANALYSIS AND SYNTHESIS OF JAZZ BY COMPUTER

John Wade Ulrich

Computing and Information Science Department
University of New Mexico
Albuquerque, New Mexico

ABSTRACT

The jazz musician has two problems. The first is the creation of melodic material. The second is the problem of fitting the melodic material into a particular harmonic structure. The work described in this paper concentrates on this second problem. To solve the problem we utilized a harmonic theory which allowed the identification of key centers and the assignment of functional meaning to chords. From this we constructed scales that were compatible with the analyzed harmonic structure. These scales were then used to translate abstract melodic material into sequences of notes that sound compatible with the background chords.

INTRODUCTION

The eventual goal of the work described here is to produce Jazz improvisations by computer. The computer's input will be a piano score of a particular song. The output will be a set of jazz variations on the original melody represented as a standard music score. This problem bears close resemblance to musical composition but there are two important differences. The jazz variations should be reminiscent in character to the original melody and they must fit with the harmonic structure of the original piano score. These two constraints define a problem which is significantly different from that of composition in general. That it is a practical and non-trivial problem can be attested to by any practicing jazz musician.

The work described here concentrates on the problem of fitting melodic material to a particular harmonic background. This is accomplished by first performing a function analysis of the song. The term functional analysis refers to the process of identifying with each chord its function within the song, and grouping together measures that move the tune from one key center to another. To identify functional structure, we first identify the key centers. Within these key centers, chords can be named and assigned functional roles. Having named chords and identified key centers we are able to compute scales which are compatible with the harmonic background.

OVERVIEW AND DEFINITION OF THE PROBLEM

As music is performed it passes through a number of temporary key centers. These temporary key centers are sometimes very difficult to establish. Jazz musicians learn to identify key centers by listening for harmonic qualities in music- These harmonic qualities depend to a large degree on which notes are sounded together during the execution of a piece of music What the musician hears has to some extent been codified in terms of a harmonic theory. The program presented by this text depends primarily on a theory described in Coker [1].

A quick but incomplete grasp of this process of identifying key centers can be made in terms of scales. Roughly speaking, a scale is an ordered set of notes from which melodies can be built. Each musical key is associated with the notes of a particular scale- Suppose that the union of the notes of three consecutive chords is identical with the notes of a particular scale. Then the chances are great that the key center is one associated with the identified scale.

While giving a good intuitive feel for how chords may be used to determine keys, the above observations may not be applied directly. There are several reasons why this is so. Firstly, in jazz, variety is achieved by substituting certain chords for others. These substitute chords are not identical with the original chords and in fact may contain notes not found in the scale associated with the key. They do, however, function harmonically as though they were the chords that they replace. Secondly, some chords are played with notes omitted. In such cases the sense of harmonic history allows the listener or musician to fill in the chord with his ear. Finally the chord may be functioning as a transition chord between two other chords and as such does not have to conform to the harmonic restrictions of the key.

Because of the above considerations, it is necessary to approach the problem of determining key centers in a rather indirect manner. The intuitive idea is this. We try out different key centers until we find one in which the observed chords each have functional meaning. Again this intuitive idea requires some refinement. The principle problem is that a set of notes determines a unique chord only if the key has already been specified. For example, the notes (D FA C) determine a D minor 7 chord in the key of C. The same set of notes determine an F major 6 chord in the key of F- We seem to be blocked because we cannot identify the key until we identify the chord, but we cannot identify the chord without knowing the key.

The approach taken to circumvent this problem is to first attach to each set of notes all possible chords with which it may be identified. The results of this process are then passed on to a key analyzer. The analyzer associates with each set of notes a set of keys in which some identified chord can function. Some key will occur in each set throughout a section of the music When this is observed, the repeating key is assigned to the corresponding section of the music It may turn out that some sections are very short and have many keys associated with them- This usually indicates a section of the music that is harmonically ambiguous and functioning as a transition section between two key centers or between two functional chords of the same key center. Finally the music and analyzed key centers are passed to a functional analyzer. The functional analyzer identifies and outputs the function of each chord with respect to its key center.

THE REPRESENTATION OF SONGS

Because of our interest in harmony we view a musical composition as a sequence of chords. Each chord is identified by a set of musical notes- (As already pointed out the set may identify more than one chord.) Each chord is preceded by a number which represents the number of beats the chord is held before the next chord is sounded. To simplify the problem somewhat we assume that chords change only on the beat. Additionally we assume that the absolute pitch of the notes is not important and that the voicing of the chords is such that each note is less than an octave apart from the next highest note of the chord. We represent the note sets as lists with the lowest note occuring first, next lowest second and so on. As an example consider the first chord of Figure 1. The list is (0 B+ D F). The convention specifies that the chord is held for two beats. We use + and - to represent the usual sharp and flat signs-

FIGURE 1: BLUE MOON (Bridge section)

```
1.1  input to chord namer
     (2(G B+ D F)2(C E G B+)4(F A C E)2(G B-D F)
        2(C E G B-)
      4(F A C E)2(B-D- F A-)2(E-G B-D-)4(A-C E-G)
        2(C E G B-))
1.2  output from chord namer/input to key analyzer
        step1.
     (2(G(3 7 10)B-(4 7 9))2(C(4 7 10))4(F(4 7 11)
        A(3 7 11))
      2(G(3 7 10)B-(4 7 9))2(C(4 7 10))4(F(4 7 11)
        A(3 7 11))
      2(B-(3 7 10)D-(4 7 9))2(E-(4 7 10))
        4(A-(4 7 11)C(3 7 11))
      2(C(4 7))2(G(4 7 10))2(G(3 7 10)B-(4 7 9))
        2(C(4 7 10)))
1.3  output from step 1
     ((16(F) ((2(G(3 7 10)B-(4 7 9)))(2(C(4 7 10)))
              (4(F(4 7 11)A(3 7 11)))(2(G(3 7 10)
                 B-(4 7 9)))
              (2(C(4 7 10)))(4(F(4 7 11)
                 A(3 7 11))))
      (8(A-) ((2(B-(3 7 10)D-(4 7 9))
                 (2(E-(4 7 10)))4(A-(4 7 11)
                 C(3 7 11))))
      (4(C)((2(C(4 7)))(2(G(4 7 10)))
      (4(F)((2(G(3 7 10)B-(4 7 9)))(2(C(4 7 10)))))
```

BASIC CHORD FORMS

In jazz the harmonic quality is largely determined by chords. Each chord is associated with a particular harmonic quality. Most people are familiar with the major and minor qualities found in religious music In jazz there are many other qualities. Two distinct chords may have the same quality even though they contain none of the same notes. This is because quality is determined by the relative distances between notes rather than absolute pitch. Additionally chords with identical notes may have a different quality depending on the harmonic context in which they are sounded-

Associated with each quality is a chord type. A chord type is just a list of numbers which give the distance from the lowest note or root to each of the other notes of the chord. This distance is measured in half steps. In western music a half step is the smallest distance that separates two notes. In Table 1, the reader is given several octaves of musical notes. The numbers give the distances from each note to the lowest note of the table . The letters are the standard names for the notes. The symbol "*" refers to sharps and flats.

TABLE 1: HALF STEP TABLE

0	C	10	*	20	*
1	*	11	B	21	A
2	D	12	C	22	*
3	*	13	*	23	B
4	E	14	D	24	C
5	F	15	*	25	*
6	*	16	E	26	D
7	G	17	F	27	*
8	*	18	*	28	E
9	A	19	G	29	F

To compute the distance between a pair of notes we take the absolute value of the difference between the associated numbers. If x is a note and n a number then the expression x + n will refer to the note n half steps above x. For instance D + 7 refers to the note A in the table- We can determine the notes of a chord when we know the lowest or root note of the chord plus its type. If x is the root and (n, ...n_k) is a chord type then the notes of the chord are (x,x+rL. , - . . ,x+n_k). For example if the root is I) and the type is (3 7 10) then the notes that are determined are (D FA C) commonly known as D minor Y We shall usually prefer the notation D(3 7 10) to (D F A C). When such notation is used the letter in front will be called the root and the list will be the chord type.

There are two types of transformations which can be performed on chords which preserve to some extent their quality. We call these transformations inversions and omissions- Inversion changes the order of the notes while omission removes one or more notes of the chord. Inversions in practice usually leave the lowest note unchanged- Such inversions we call root 1 inversions. Less frequently the inversion changes the lowest note. This is done most often to accommodate the base line. Inversions whose lowest note is the second note of the original chord we call root 2 inversion. Inversions whose lowest note was the third note of the original chord we call root 3 inversions and so forth. Omissions provide additional problems because to determine the chord from a set of notes we must find out which notes were left out. In practice these missing notes are determined by the harmonic context of the chord . This can be accomplished only if the basic functioning parts of the chord are left in tact.

Consider the notes (FA C D) which identify F major 6 chord and a root 2 inversion of D minor 7. If we omit the A we get (F C D). This group of notes sounds much more D minor 7 than F major 6 even though F is the lowest note of the group. The problem is that we removed the second note of the chord- For similar reasons removing the F or the D would render the chord unsuitable for F major 6. The general principle that we adopt is to disallow the omission of the lower two notes and the highest note in the chord. (This is in accordance with with standard musical practice). Omission of any or all of the other notes will be allowed and we shall need to find ways to determine the omitted notes.

We are now ready to define the entire set of basic chords. This is accomplished with the chord grammar given in Table ?. In addition to the complete forms of the chords the grammar generates all possible forms with omitted notes. The names of the nonterminals may be suggestive of the usual names of the chords for those familiar with standard chord nomenclature. The chords generated are most of the chords in common use . Table 5 is provided for the reader who wishes to make a quick association between some chord names used by the program and those names in common musical usage. The variable x of the table can be replaced with any musical note. For instance if G is substituted for x the table tells us that G major 7 is named G(4 7 11) by the program G minor 7 is named G(3 7 10) and so on.

TABLE 2: CHORD GRAMMAR

```
<chord> ⟶ <root><type>
<type> ⟶ (4<major>)|(4<seven>)|(3<minor>)|
  -(3<dim>|(5<seven>)
<major> ⟶ 7|7<majortail>|<majortail>|
  6<major7flatfivetail>
<major7flatfivetail> ⟶ 10
<majortail> ⟶ 11|11 <M6/7tail>|9 <M6/7tail>
<m6/7tail> ⟶ 14|14 <m9tail>|<m9tail>
<M9tail> ⟶ 18|18 <M+11tail>|<M+11tail>
<M+11tail> ⟶ 21
<minor> ⟶7|7<mtail>|<mtail>|6 <halfdimtail>
<mtail> ⟶9|9 <m6/7tail>|10|10 <m6/7tail>|11|11
  <m+7tail>
<halfdimtail> ⟶ 10|10 <m6/7tail>
<m6/7tail> ⟶ 14|14 <m6/7tail>
<m+7tail ⟶ 17|17 <m+7,9tail>
<m+7,9tail> ⟶ 21
<seven> ⟶ 7 <dom7>|<dom7>|6 <dom7>|8 <dom7>
<dom7> ⟶ 10|10 <seventail>
<seventail> ⟶ 14|14 <7,9tail>|13 <7,9tail>|15
  <7,9tail>
<7,9tail> ⟶ 18|18 <7,9,+11tail>
<7,9,+11tail> ⟶ 21
<dim> ⟶ 6 <dimtail>
<dimtail> ⟶ 9
<root> ⟶ [any of the names for musical notes]
```

THE RECOGNITION OF CHORDS

The grammar of Table 2 may not be used directly to recognize chords because not all inversions of chords are generated. Instead a somewhat more complicated approach must be taken. First one of the note is selected as the root. Then the rest of the notes are seen to be explainable in terms of a permutation or a basic chord type This is done for all possible choices of root. The algorithm given inputs a list of notes and outputs a list of chords which may be identified with the input. For instance if the input is (DFAC) then the output will be (D(3 7 10)F(4 7 9)) since (D FA C) represents D minor 7 root 1 or F major 6 root 4. If the input is (F A C D) the output will be identical to the first case since (FA C D) represents D minor 7 root ? or F major 6 root 1. Finally if the input is (C E B) we have a situation where a note was omitted. The original chord should be (C E G B) or (C E G- B). In this case the algorithm returns *C{4* 7 11) corresponds to (C E G B). It is assumed that the reader is familiar with the concept of a nondeterministic algorithm. The choose...thendo..-else construction used in the chord recognition algorithm is to be interpreted as follows: if some choice can be made which satisfies the conditions between the choose and the thendo then the code following the thendo is executed. Otherwise control passes to the code following the else- The execution of back track(x_1 ....x_n) has its usual meaning with control passing to the point of the last successful choice and the variables x_1 ..x being reset to their value at the time just prior to the last successful choice.

TABLE 3: SUMMARY OF COMMON CHORDS

COMMON NAME	PROGRAM NAME
xM7	x(4 7 11)
xm7	x(3 7 10)
x7	x(4 7 10)
x6	x(4 7 9)
x7#3	x(5 7 10)
xdim	x(3 6 9)
xø	x(3 6 10)

Recognition algorithm

```
input: chord x, chord grammar; L:=nil;n:=0;
   m:=nil;
choose an element e of x which is not in m
   thendo m:=cons(e,m); y:="<type>";z:=nil;
   loop:remove e+n from x;
      if x is empty and y="<stop>"then add the
         chord e(z) to L;
      if x is empty or y="<stop>" then
         [A]backtrack(z,y,x);
      choose an unmarked rule of the form
         y→p,w such that e+p belongs to x
            or y→w is a rule of the grammar
               thendo
         mark the rule; n:=p;  [w matches any
         z:=cons(z,n)           nonterminal-
         y:=w                   p matches any
         goto loop              terminal]
      else choose an unmarked rule of the form
         y→p such that e+p belongs to x thendo
         mark the rule; n:=p;
         z:=cons(z,n);
         y:="<stop>";
         goto loop;
      else do
```

```
            unmark all rules that begin y-;
            [B]backtrack(z,y,x)
              od
         else outputL;
```

There are two backtracking operations marked A and B. At A either x is empty or y=<stop>. If x is empty then we have exhausted the set of notes. If y=<stop> then there are no more nonterminals to process. If both of these conditions are true we have succeeded in identifying each note with a chord element. The backtrack at B means that we have exhausted all rules beginning with y.

The chord namer applies the recognition algorithm to each group of notes in the song. Tt produces an output identical to the input except that the groups of notes are replaced by a set of possible chords. Figure 1 illustrates this process. Figure 1.1 gives a transcription of the bridge section of BLUE MOON as it would be presented to the chord namer. Figure 1-2 represents the output from the chord namer.

ANALYSIS OF KEYS

The next step in determining the functional structure takes place in the key analyzer. This program partitions the music into harmonically uniform segments and assigns to each segment a key center. This process is accomplished in several steps. Each step may be thought of as a refinement of the previous steps. The process is roughly as follows. We begin be segmenting the song into as many segments as there are chords. Our aim is to reduce the number of segments to as few as possible. We reduce the number of segments by combining adjacent segments. Segments may be combined in just the case that our musical theory can analyze all the chords of both segments in a single key. Thus each segment must be harmonically consistent. This condition is trivially satisfied when each segment contains a single chord.

The combining of adjacent segments takes place in a single program called "combine". Since it is used at each stage of the key analyzer, we begin by describing combine. The input to combine is a list of segments (s_1 s_2 ... s_n). Each segment is a list of three elements (1 k c) where 1 is a number representing the length of the segment, k is a set of possible keys for the segment and c represents the sequence of chords making up the segment. Each element of c is a list of two elements (n cset) where n is the number of beats the chord is held and cset is a set of possible names for the chord as determined by the chord namer. See for example Figure 1. Let (L_1 K_1 C_1) and (L_2 K_2 C_2) be two adjacent segments. We can combine them if the common part of K_1 and K_2 is not empty. In this case the combination is just list (L_1 + L_2, intersection^- ,K_2) , append(C_1,,C_2)). The combining algorithm performs the combining step until it no longer applies to any pair of segments. The outcome certainly might depend on the order in which the pairs were processed and one can construct artificial examples to prove this point. With actual music it seems sufficient to process from left to right and ignore other possibilities.

```
combine(s);input s; output t; t:=nil;
  while s is not empty do
    (n k c):= first(s);
    s:= tail(s);
    loop: if s not empty then do
      (n' k' c'):= first(s);
      if k' and k have an element in common thendo
        n:= n + n';
        k:= intersection(k,k');
        c:= append(c,c');
        s:= tail(s);
        go to loop;
        od                              \
      od
    t::= append(t,list(list(n,k,c)));
  od
end of combine;
```

The theory we use for determining possible function is discussed in [1] . It is based on what are commonly called seventh chords. For the purpose of applying this theory we trunce.te our chords by removing all numbers greater than 12 from the type. This effectively discards all superpositions of the basic seventh chords. In jazz harmony there arc three important functional roles called the tonic, dominant and subdominant. If x is the key then the chords x(4 7 11), x+7(4 7 10) and x+?(3 7 10) respectively play these roles. The rough harmonic analysis performed at the first stage of key analysis partitions the music based on this fact. In doing this some mistakes will be made and must be corrected later by using a more refined harmonic theory. The above observations yield the following:

1. *x{4* 7 11) and x(4 7 9) and *x{4* 7) function as tonic in x
2. *x{4* 7 10) functions as dominant in x-7
3. x(3 7 10) functions as subdominant in the key of x-2.

The first step of key identification processes each set of possible chords as they arc determined by the chord namer. If any of the chords are of the above type it is assumed to be functioning in one of the above roles. The key center is determined on the basis of this assumption. For an example of the execution of this step refer to Figure 1.3.

```
stepl(Ll,L2);input Ll; output L2; L2:=nil;
  while Ll not empty do
    y:= nil;
    n:= first(Ll); Ll:= tail(Ll);
    chordset:= first(Ll); Ll:= tail(Ll);
    for each chord x of chordset do
      let rt = root of x and t = type of x;
      if t = (4 7 11) or t = (4 7 9) or t = (4 7)
        then add rt to y else
      if t = (3 7 10) then add rt-2 to y else
      if t = (4 7 10) then add rt-7 to y
      od;
    L2:= append(L2,list(list(n,y,list
      (n,chordset))))
    od
  L2:= combine(L2);
end stepl;
```

The song HOW HIGH THE MOON is illustrated in Figure 2. Notice that the output from stepl contains some very small segments of questionable harmonic meaning. This will be true in general because of the simple harmonic theory used in stepl. There are three possible reasons why these small .segments may belong to an adjacent segment. First of all the segment may contain chords that are functioning in the place of other chords. This is a common practice in jazz music since it provides harmonic variety to the music. For example in the key of C, E(3 7 10) may function as C(4 7 11). The original analysis, by not taking this into account would have concluded that E(3 7 10) was in the key of D instead of C (see 2.2). The second reason why a small segment may be assigned the wrong key is that it is functioning as tonic relief. Tonic relief is a harmonic, device which is used to strengthen chord progressions by moving the key center up five half steps. The center usually returns to the original key within a few measures. For this reason most musicians do not regard tonic relief as a proper key change and prefer to retain the original key throughout the segment. Finally the small segments may be sequences of transition chords which arc used in music to provide a smooth change from one chord to another. The use of transition chords is an area least well covered by functional theory. This is because transition chords actually function outside the usual harmony of the music. Invariably they resolve into something functional or into another transition. The reader will find it helpful to follow the example of Figure 2. To make reading easier to follow, multiple names *of* chords have been omitted from the example input.

FIGURE 2: HOW HIGH THE MOON (first 16 bars)

```
2.1  output from stepl
     ((3( )((3(B)(5 7 10))))
      (8(G)((4(G(4 7 11)))(4(G(4 7 9))))
      (4(F)(4(G(3 7 10)))
      (4(B)((4(G-(4 7 10))))
      (8(F)((4(F(4 7 11)))(4(F(4 7 9))))
      (4(E-)((4(F(3 7 10))))
      (4(A)((4(E(4 7 10))))
      (4(E-)((2(E-(4 7 11)))(2(C(3 7 10))))
      (2( )((2(A(3 6 10))))
      (6(G)((2(D(4 7 10)))(4(G(4 7 9))))
      (4( )((4(B(5 7 10))))
      (6(G)((2(G(4 7 11)))(2(A(3 7 10)))
        (2(B(3 7 10))))
      (2( )((2(C(3 7 11))))    )

2.2  output from step2
     ((3( )((3(B(5 7 10))))
      (8(G)((4(G(4 7 11)))(4(G(4 7 9))))
      (16(F)((4(G(3 7 10)))(4(G-(4 7 10)))
        (4(F(4 7 11)))(4(F(4 7 9))))
      (12(E-)((4(F(3 7 10)))(4(E(4 7 10)))
        (2(E-(4 7 11)))(2(C(3 7 10))))
      (2(B-)((2(A(3 6 10))))
      (6(G)((2(D(4 7 10)))(4(G(4 7 9))))
      (4( )((4(B(5 7 10))))
      (6(G)((2(G(4 7 11)))(2(A(3 7 10)))
        (2(B(3 7 10))))
      (2( )((2(C(3 7 11))))    )

2.3  output from step3
     ((11(G)((3(B5 7 10)))(4(G(4 7 11))).
        (4(G(4 7 9))))
      (16(F)((4(G(3 7 10)))(4(G-(4 7 10)))
        (4(F(4 7 11)))(4(F(4 7 9))))
      (12(E-)((4(F(3 7 10)))(4(E(4 7 10)))
        (2(E-(4 7 11)))(2(C(3 7 10))))
      (18(G)((2(A(3 6 10)))(2(D(4 7 10)))
        (4(G(4 7 9)))(4(B(5 7 10)))(2(G(4 7 11)))
        (2(A(3 7 10)))(2(B(3 7 10))))
      (2( )((2(C(3 7 11))))     )
```

Step2 refines the initial segmentation by applying possible chord substitutions. The substitutions used are listed in Table 4. The second column of Table 4 specifies the function of the chord that it is substituting for. It is to be understood that x varies over the set of musical notes/keys. For instance if x is F then the tablc tells us that *F[4* 7 10) can function as the dominant of the key of F-1 or E. Step2 begins by augmenting the key sets of small segments by allowing substitute chords to function as the tonic, dominant, or subdominant of the segment's key. A small segment is defined to be shorter than P1 where P1 is a system parameter. When this is completed combine is used to merge adjacent segments that may now be combined.

```
step2(L2,L3);input L2; output L3; L3:= nil
  while L2 not empty do
    (n k seg):= first(L2);
    L2:= tail(L2);
    if n<pl then
      let k = the set of keys w such that each
        chord of seg has a substitution function
        in w according to table 4;
    L3:= append(L3,list(list(n,k,seg)));
  od;
  let L3 = combine(L3);
end step2;
```

TABLE 4: TABLE OF CHORD SUBSTITUTIONS

CHORD	FUNCTION	KEY
x(3 7 10)	tonic	x-3
x(3 7 10)	tonic	x-9
x(3 6 9)	dominant	x-11
x(4 7 10)	dominant	x-1
x(3 6 9)	dominant	x-6
x(3 6 9)	dominant	x-3
x(3 6 9)	dominant	x-11
x(4 7 10)	subdominant	x-2
x(4 7 11)	subdominant	x-5
x(4 7 9)	subdominant	x-5

Step3 removes transition chords by locating all segments x that contain a single chord. The key set of x is replaced by the keyset of the next segment. The modified segments are then combined. Figure 2.3 illustrates the output from this step.

```
step3(L3,4)  input L3; output L4; L4;= nil;
  L3:= reverse(L3);
  (N' k' seg'):= first(L3); 13 = tail(L3);
  while L3 not empty do
    n,k,seg:= n'k' ,set' ;
    (n' k' seg'):= first (L3); L3:= tail(L3);
    if seg' has one chord then k':= k;
    L4:= cons(list(n,k,seg),LU);
    od;
  L4:= combine(L4;
end step 3;
```

The final step in the analysis is the removal of segments that represent tonic relief. These segments are recognized by being short (i.e., less than p2 beats where p2 is a parameter) and by being five half steps above the next segment. Figure 3 illustrates the execution of step 4.

FIGURE 3: ALL THE THINGS YOU ABE (first 4 bars)

```
3.1  output from step3
     ((4(E-)((4(F(3 7 10))))
      (12(A-)((4(B-(3 7 10)))(4(E-(4 7 10)))
        (4(A-(4 7 11))))     )

3.2  output from step4
     ((16(A-)((4(F(3 7 10)))(4(B-(3 7 10)))
                (4(E-(4 7 10)))(4(A-(4 7 11))))     )
```

```
step4(L4,L5) ; input Lk; output L5; L5:= nil;
  while L4 not empty do
    (n k seg):= first(L4);
    L4:= tail(L4);
    if L4 not empty then do
      (n' k' seg'):= first(L4);
      s:= set of all keys of k' that are five half
         steps below some member of k;
      If s not empty and n < p2 then do
        L5:= append(L5,list(list(n, s, seg),list
           (n', s, seg')));
        L4:= tail(L4)
      od else
        L5:= append(L5, list(list(n, k, seg)))
    od else L5:= append(L5, list(list(n, k, seg)))
  od;
  L5:= combine(L5);
end step k
```

FUNCTIONAL ANALYSIS OF CHORDS

Having completed the identification of key centers we are ready to associate with each chord its functional meaning. This is accomplished by applying a harmonic theory to each of the chords of the song. If the chord belongs to a segment with two or more possible keys then it will be analyzed in more than one way. The theory that we shall use is embodied in Table 5. The x stands for the key in which the chords are being analyzed. The analysis program instantiates the table by substituting for x the key of the current segment. It then looks up each chord of the segment in the instantiated table. If a chord cannot be found in the table then it is assigned the role of transition. The algorithm is not difficult and is omitted. An example of the output is given in Figure 4.

TABLE 5: TABLE OF CHORD FUNCTIONS

CHORD	FUNCTION
x(4 7 9)	tonic of x
x+7(4 7 10)	dominant of x
x+2(3 7 0)	subdominant of x
x(4 7 11)	tonic of x
x(4 7)	tonic of x
x+9(4 7 10)	transition
x+4(3 7 10)	substitute for x(4 7 11)
x+9(3 7 10)	substitute for x(4 7 11)
x(4 7 11)	dominant of x+5
x+5(4 7 11)	tonic relief
x+7(3 7 10)	subdominant of x+5
x+4(3 7 10)	transition
x+2(4 7 10)	substitute for x+2(3 7 10)
x+3(3 7 10)	substitute for x+7(4 7 10)
x+1(7 10)	transition
x+1(4 7 10)	substitute for x+7(4 7 10)
x+6(3 6 9)	substitute for x+7(4 7 10)
x+3(3 6 9)	substitute for x+7(4 7 10)

FIGURE 4: HOW HIGH THE MOON (analysis summary)

```
  KEY LENGTH   CHORD        FUNCTION
 1 G     3    B(5 7 10)  TRANSITION
 2 G     4    G(4 7 11)  TONIC OF G
 3 G     4    G(4 7 9)   TONIC OF G
 4 F     4    G(3 7 10)  SUBDOMINANT OF F
 5 F     4    G-(3 7 10) SUBSTITUTE DOMINANT OF F
 6 F     4    F(4 7 11)  TONIC OF F
 7 F     4    F(4 7 9)   TONIC OF F
 8 E-    4    E(4 7 10)  SUBSTITUTE DOMINANT OF E-
 9 E-    4    E-(4 7 11) TONIC OF E-
10 E-    2    E-(4 7 11) TONIC OF E-
11 E-    2    C(3 7 10)  SUBSTITUTE TONIC OF E-
12 G     2    A(3 6 10)  TRANSITION
13 G     2    D(4 7 10)  DOMINANT OF G
14 G     4    G(4 7 9)   TONIC OF G
15 G     4    B(5 7 10)  TRANSITION
16 G     2    G(4 7 11)  TONIC OF G
17 G     2    A(3 7 10)  SUBDOMINANT OF G
18       2    C(3 7 11)  UNANALYZED
```

THE FITTING OF MELODIC MATERIAL TO A SPECIFIED HARMONIC BACKGROUND

We now indicate how the analysis can be used to fit melodic material to a given harmonic background. We use motifs to represent melodic material. In music a motif refers to a small snatch of melody. Sequences of motifs are woven together to form a melody. Rather than constantly inventing new motifs, the musician modifies old ones to fit new harmonic situations. This is accomplished by imposing on the motif a scale based on the key of the current section of music. Examples and discussions of motifs may be found in Coker [1] Chapter 2.

We use the notion of an abstract motif. An abstract motif is like a motif except that is is void of rhythmic and harmonic structure. When these motifs are fit into a specified rhythmic and harmonic context they become motifs in the usual sense. The harmonic context is determined by the key centers and the chord names produced by the key analysis program. We will show how abstract motifs may be instantiated with respect to harmony. To illustrate the process we will fit instantiations of a motif to the first eight measures of HOW HIGH THE MOON.

Before we can understand how instantiations are computed, we must understand the use of scales. It was mentioned above that a scale was a sequence of notes from which melodies are constructed. Most are familiar with the sound of C major scale. This scale corresponds exactly to the white keys on the piano. If the reader refers to Table 1, starts at the first C in the table, and proceeds up through the lettered notes to the next C in the table, then he will have identified C major scale. If we compute the half steps associated with each of the notes we get the list (0 2 4 5 T 9 11 12). This suggests that we represent C major scale similar to the notation with which represent chords, i.e., C(0 2 4 5 7 9 10). We call C the root of the scale and (0 2 4 5 7 9 10) the type. D(0 *2 4* 5 *'(9* 11 1?) will be D major scale and so on. Besides the major types of scales there are many others in general use. Gome common ones are dorian, mixolydian and whole tone. They are defined by the scale types (0 2 3 5 7 9 10 12), (0 *2 4* 5 7 9 10 12) and (0 2 4 6 8 10 12) respectively. Each type of chord is associated with a set of scale types that are harmonically consistent with the chord. Coker lists these associations. We give in Table 6 a list of the scales commonly referred to and their set of compatible chords.

TABLE 6: SCALES

NAME OF SCALE	TYPE OF SCALE	COMPATIBLE CHORDS
major	x(0 2 4 5 7 9 11)	x(4 7) x(4 7 11) x(4 7 9)
dorian	x(0 2 3 5 7 9 10)	x(3 7 10)
locrian	x(0 1 3 5 6 8 10)	x(4 7 10)
mixolydian	x(0 2 4 5 7 9 10)	x(4 7 10)
whole tone	x(0 2 4 6 8 10)	x(4 7 10) (4 6 10)(4 8 10)

diminished	x(0 2 3 5 6 8 9 11)	x(3 7 10) x(3 6 9)
		x(3 6 10) x-1(4 8 10)
		x-1(4 7 10)

We specify a motif as a sequence of displacements measured in scale steps. For instance (2 5 7 8 7 5 1 7 8 7 8 9) represents a motif of 12 notes. If we specify a scale say C major then we can identify each number in the abstract motif with a particular note. This is accomplished by associating each number of the motif with a location in the scale type. To compute a note associated with a number in the abstract motif we use the number as an index into the scale. This obtains for us the number of half steps the desired note is above the first note of the scale. Using C major scale with the above example we obtain (D G B C B G DB C B C D), If on the other hand we specify G mixolydian we get (A D F G F D A F G F G A) which is harmonically very different from the first case.

The instantiation of a motif should retain its pitch throughout a segment. When a new segment is encountered however the motif is shifted in pitch so that it is now expressed relative to the new key center. There is a problem however in following this strategy. When a new chord is encountered the harmonic quality of the music changes. This usually requires that other notes of the original scale be altered. Additionally notes of the motif that belong to the chord should be emphasized. This is accomplished by accent, rhythm and phrasing.

To accommodate the changing harmonic quality of the music, musicians are taught to "think chords". Each chord that is sounded is associated with a set of compatible scales that are built on the root of the chord. The musician selects one of these scales. By restricting the notes he plays to those of the scale selected he avoids creating harmonic conflict. For instance the G(l 3 4 6 7 9 10 12) scale is used with the *G(4* 6 10) chord. This is not entirely satisfactory for our purpose since the motif is expressed relative to the key of the entire segment which is likely to be different than G. In order to maintain the pitch of the motif and at the same time follow the usual musical practice, we modify the scale that the musician would use as follows: Suppose that x is the key and y is the scale that is suitable. We create a new scale which is a permutation of y and has x as the root note. This new scale then becomes the basis for instantiating the motif. For example if the key is C then the scale G(0 1 3 4 6 7 9 10) would be modified to C(0 1 2 4 5 7 8 10 11). These two scales contain exactly the same notes, just the root is different. Under some conditions x will not belong to the desired scale. In such cases we require the lowest note of the permuted scale to be above x but as close to x as possible. For computational reasons we keep x as the root of the scale even though in this case the scale does not contain x.

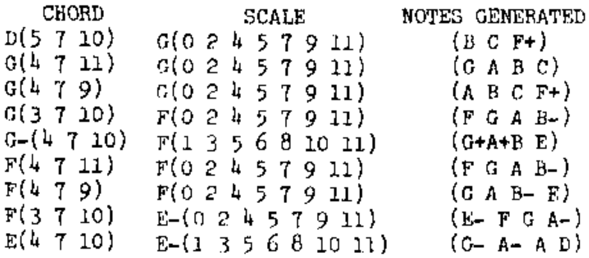
We have applied these concepts to generate an improvisation for the first eight bars of HOW HIGH THE MOON. The rhythm was specified before hand in an ad hoc manner. The result is illustrated in Figure 5. The scales were computed on the basis of Table 6. The motif was repeated when necessary. From such a simple technique the result is bound to be less than interesting music. The music is however consistent with the harmonic structure of the song.

DISCUSSION

A feature of this work is that it takes a non-grammatical approach to harmonic analysis. Some confusion may arise at this point because of the use of a grammar for the recognition of chord types. However the chord grammar is only used for the initial naming of chords. The recognition of key centers and functional analysis of chords are noticeably non-grammatical.

FIGURE 5: HOW HIGH THE MOON (improvised line)

rhythm: straight four beats to the measure



CHORD	SCALE	NOTES GENERATED
D(5 7 10)	G(0 2 4 5 7 9 11)	(B C F+)
G(4 7 11)	G(0 2 4 5 7 9 11)	(G A B C)
G(4 7 9)	G(0 2 4 5 7 9 11)	(A B C F+)
G(3 7 10)	F(0 2 4 5 7 9 11)	(F G A B-)
G-(4 7 10)	F(1 3 5 6 8 10 11)	(G+A+B E)
F(4 7 11)	F(0 2 4 5 7 9 11)	(F G A B-)
F(4 7 9)	F(0 2 4 5 7 9 11)	(G A B- E)
F(3 7 10)	E-(0 2 4 5 7 9 11)	(E- F G A-)
E(4 7 10)	E-(1 3 5 6 8 10 11)	(G- A- A D)

Consider for contrast Winograd's study [5] on the analysis of classical music. His program embodies a simplified version of Forte's harmonic theory [2]. The theory is represented as a systemic grammar. The form of the input to Winograd's program is similar to the input to the program described in this paper. His examples however are entirely from Bach and Schubert.

Winograd reports dealing with the same sorts of ambiguities as described in this study. Because of these ambiguities, WinOgrad's parser uses a set of heuristics to reduce the number of possible parsings to a managable size. When two different parsing are found, a preference is given to the simpler of the two. The program reported here resolves ambiguities in favor of the analysis which has fewer key changes.

In the long run the question of whether grammatical approaches are superior to non-grammatical ones may depend on how much harmonic structure can be derived from local properties of the music. If global properties are essential to harmonic analysis, then a grammar may provide a useful means of representing the global organization. However in the case of Jazz music, a musician can usually identify the key on the basis of two or three consecutive chords. If this statement can be verified experimentally then there is strong reason to believe that jazz encodes so much harmony in the local structure of the music that'global considerations can be ignored.

The assumption is made in this study that it is meaningful to represent melodic material as abstract motifs devoid of rhythmic and harmonic content. This assumption can be regarded as an approximation at best and must stand the trial of future investigation. Moorer *[k]* reports a study (probably originally from Lundin [3])which demonstrates that subjects can determine on the basis of relative pitch when two consecutive notes will achieve melodic resolution. This suggests that harmonic principles (i.e., relative pitch) may effect melodic features in essential ways. Such evidence must be weighed against the fact that when harmony and rhythm are removed from certain melodic material, patterns emerge that suggest that melody has a certain independence. For instance if the first nine notes of the allegro to Beethoven's fifth are expressed as an abstract motif we obtain (0 2 *k* 3 2 1 0 10). This is the same abstract motif obtained from the first nine notes of the Piano Concerto in C minor even though these two pieces are rhythmically and harmonically quite dissimilar.

CONCLUSIONS

The jazz musician has two problems. The first is the creation of melodic material. The second is the problem of fitting the melodic material into a particular harmonic structure. The work described in this paper concentrates on this second problem. To solve the problem we utilized a harmonic theory which allowed the identification of key centers and the assignment of functional meaning to chards. From this we constructed scales that were compatible with the analyzed harmonic structure. These scales were then used to translate abstract melodic material into sequences of notes that sound compatible with the background chords.

The work described here is just a beginning. For instance the theory used does not apply to songs with minor tonality and needs to be enlarged. The method for fitting motifs to harmonic structure is more an illustration than a reliable technique. The use of rhythm was not treated at all nor was the problem of developing a library of useful motifs. In spite of these shortcomings the methods described do deal quantitatively with a number of practical problems facing jazz musicians such as the identification of keycenter, the identification of chords and the construction of scales that are harmonically compatible with the identified keys.

ACKNOWLEDGEMENT

The author would like to thank Professor Nancy Martin for her encouragement and helpful suggestions related to this work.

BIBLIOGRAPHY

[1] Coker, J., IMPROVISING JAZZ, Prentice-Hall, Englewood Cliffs, N.J.

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

[3] Lundin, R., An Objective Psychology of Music, Ronald Press.

[4] Moorer, J. A., "Music and Computer Composition", Communications of the ACM 15,2 (Feb. 1972), 104.

[5] Winograd, T., "Linguistics and Computer Analysis of Tonal Harmony", Journal of Music Theory, 12 (1969), 1968.