C. P. U. Bach: A Probabilistic Rule-Based System to Compose Chorales

John David Walvoord

University Undergraduate Fellow, 1992-1993

Texas A&M University

Department of Computer Science

APPROVED

Fellows Advisor Walter C. Daugherity

Honors Director

Introduction

Music composition has historically been considered as an art in which great composers are not taught, but are born. Before continuing this discussion, though, it is necessary to provide a working definition of music composition within the scope of this paper. In the broadest sense, a composer would be any person who wrote any original piece of music. However, for the purposes of this discussion I will define the term classical composition to be limited to works which fit into the genre of "classical music" based on western harmony. Again, problems are presented by how one would define classical music. At the risk of over limiting the definition, I will functionally define classical music as music of the Baroque, Classical, or Romantic Periods of music; or of any period in which the work resembles the music of these periods in form and harmonic structure.

Common beliefs place classical composition as unattainable by the common man and better left to those rare musical geniuses as Bach, Mozart, or Beethoven. While almost any person who can play a guitar or piano at some time will try their hand at writing a "top 40 hit," few modern composers will attempt a classical composition without at least having or be working towards a Bachelor's degree in music. This is entirely understandable because of the great amount of music theory knowledge which is necessary to even begin a classical composition. To compose requires that one not only be thoroughly familiar with musical notation, but also that one be familiar with much additional theoretical knowledge such as understanding the relationships of harmonic progressions and the uses and effect of different cadences, understanding the principles of voice leading and counterpoint which define the "rules" of composition, and finally understanding when it is acceptable to break these rules.

As I stated earlier, there is a popular notion that one cannot learn to be a great composer, but must be born one. Wolfgang Amadeaus Mozart reportedly conceived perfect melodies in his head. However, Ludwig Van Beethoven revised his melodies continuously (Dallin 1974, 4). These two artists perhaps represent the extremes of composer methodology. However, both have the shared attribute that before either was acknowledged for his composing ability (Mozart was acknowledged quite young as a child prodigy for his performance ability) each had to spend countless hours studying the rules of counterpoint and even more hours engaged in the practice of various technical exercises. Because of this shared commonality of not only Mozart and Beethoven, but also of the vast majority of composers, perhaps great composers can be made.

If it is possible to pass along the skills necessary to produce greatness, it is sad that so little has been done to attain this result. Hindemeth asks the question: "Is it not strange that since Bach hardly any of the great composers have been outstanding teachers?" (1942, 3). Though much work has been done to analyze the work and techniques of the masters, there is still much that is not understood about how pass to this ability along to others. This is the motivational force behind this study.

Purpose

The purpose of this study, stated plainly, is to develop a computer system which is capable of composing chorales. The object is not just to compose a piece of music which is recognizable as a chorale, but to compose a chorale which may be identifiable as a good chorale. It is the reasoning of this study that, if a person can teach a computer to compose good chorales, then surely other persons could be taught similarly. Therefore, it is not the intention of this study to in any manner entertain the thought of replacing human composers with computers. A computer brings with it absolutely no outside influences and has only the knowledge that the programmer gives to it. Therefore, it is the hope that in using a computer as a student, the techniques of teaching good composition can be more easily isolated than when working with human students.

However, there is a dual purpose to this study. While it is hoped that the general body of knowledge concerning music composition can be improved by incorporating computers, it is likewise hoped that the general body of knowledge concerning the techniques of artificial intelligence can be bettered by incorporating the cognitive task of music composition.

Now that the purpose of the study has been determined, it would be appropriate to discuss the focus that has been chosen. Why should chorales be composed as opposed to fugues or sonatas? The reason for this lies in the simplicity of the chorale. Chorales represent the simplest form of classical music and therefore present themselves as the logical starting point. It would be appropriate to define a chorale at this point. A chorale is defined in *The Concise Oxford Dictionary of Music* as a metrical hymn tune characteristic of the German Reformed Church and sung in unison (Kennedy 1980, 131). An example of Bach's chorale *Freu Dich Sehr, O Meine Seele*, is shown in Figure 1. Careful study of this chorale will reveal several features that are characteristic of chorales. First of all, the focus of the chorale is on the harmony which changes with each beat. The rhythms are simple, composed almost entirely of eighth notes and quarter notes. And, the melody is primarily stepwise so that it may be sung by a congregation with relative ease.

The Role of J. S. Bach

Johann Sebastian Bach plays a significant role in the design of this system. Bach's chorales represent the culmination of chorale theory. Kostka asserts that ". . . Bach chorales epitomize the late Baroque approach to chorale writing . . ." (1989, 69). Therefore I have chosen his works as the basis around which this system is designed. His chorales are crucial to the harmonic progressions and the non-chord tones of the chorales this system generates.

Freu Dich Sehr, O Meine Seele

Fig 1. *Freu Dich Sehr, O Meine Seele* by J. S. Bach.

Limitations

Before proceeding to the description of how the system was designed, it would be beneficial to discuss the current limitations of the system and why they exist. The first limitation is that all chorales are in 4/4 time. This is a minor limitation, and is attributable to lack of time. It was judged at the time of system creation that limiting the ability of the system to 4/4 time would not really hinder the acquisition of any knowledge that could have been gained by including 3/4 time into the system's capabilities.

The second limitation that needs to be discussed is that all chorales are in the key of C. Once again, this is considered a somewhat minor limitation in that no real additional knowledge could be gained by permitting additional key signatures. Certainly, any other key signature could have been attained simply by transposing the generated chorale using an offset value and checking that no part went outside of its allowable part range. That is, sufficient checks would have to be made to prevent say a soprano part from being written too low for a soprano to sing because the part had been transposed down to another key. These considerations alone presented little difficulty and the original intent of the system was to allow this to happen. However, it was quickly realized that by not limiting the starting conditions of the chorale to the key of C, the task of starting the chorale generation process became considerably more complex. This limitation could be overcome. But, once again it was determined that no real knowledge would be lost (to the extent that this project covers) by this limitation and the time saved by imposing this constraint was considerable. If further work is attempted serious consideration will be given to removing this limitation.

The final limitation that warrants discussion is that all of the chorales composed by the system are in the major mode. This is certainly the most serious limitation and does threaten to hinder the amount of knowledge that could be gained from this project. Unfortunately, the minor mode is not nearly so straightforward as the major mode. There are additional rules that exist when working in the minor mode that govern altered scale tones which do not have to be dealt with in the major mode. Furthermore, chord progressions do not resolve in the same manner as in the major mode. Therefore, composing in the minor mode is not only more complex than composing in the major mode, but it has a fundamentally different sound. In addition, I am not entirely comfortable composing in the minor mode and do not "trust my ears" to know that what I am producing is indeed what is acceptable for the minor mode to sound like. It is primarily for this reason that I do not consider this limitation so much a system limitation as I consider it a designer limitation.

Notation

The following notational conventions will be used throughout the paper:

Chords are typed in boldface capital letters	**Am**
Pitches are typed in italicized capital letters	*C*
Absolute pitches are represented by note and octave	*C4* (middle *C*)
When not referring to any specific key signature	
Major chords	**I, IV**
Minor chords	**ii, iii**
Diminished chords	**vii°**
When referring to a specific key signature	
Major chords	**C, D, F**
Minor chords	**Am, F#m**
Diminished chords	**C°, F#°**

Approach

Before beginning the composition process the system asks the user to input some initialization parameters. First, the system asks for the filename to which it should write the chorale. Then the user is asked to enter how many phrases the chorale is to have. Each phrase will conclude with a cadence and at the beginning of each new phrase the last melody note of the previous phrase will be the starting note of the next phrase. After the user enters how many phrases the chorale should contain, the user is asked by the system to enter how many measures are in each specific phrase. These are the only parameters the user is allowed to enter, and after this is done, the system generates the chorale entirely on its own.

I will walk through the generation of a chorale that the system actually composed as the approach is explained to further illustrate the process used. For the chorale I will be using as an example, the number of phrases is two with four measures in each phrase.

In composing the chorales, this system breaks the process down into four steps:

1. Compose the melody
2. Harmonize the melody
3. Write the three remaining parts
4. Add non-chord tone eighth notes

The first three steps use only quarter notes. Eighth notes are not introduced into the chorale until the fourth step.

Compose the Melody

The process of composing the melody is itself broken down into three steps.

1. Pick an initial starting note
2. Compose the rest of the melody
3. End the melody so that an Authentic Cadence (**V** - **I** chord progression) may be generated

The starting pitch is determined by a randomly generated number with the following probabilities assigned to the following results:

Probability	**Result**
0.5	*C5*
0.25	*E5*
0.25	*G4*

Table 1. Probabilities of initial melody note

For the chorale which will be used as an example for the duration of this paper, let us assume that a random number is generated such that the initial pitch is *E5*.

After the initial pitch is determined, the rest of the melody is composed. This is done based on probabilities of moving from one note to another. A complete table of state probabilities is given in Appendix A. Since our example has given us an initial pitch of *E5* let us look at the state values for moving from an *E5* to various different pitches given in Table 2:

E5	→	*E5*	*C6*	*D6*	*E6*	*F5*	*G5*	*A5*	*B5*	*C5*	*D5*	*E4*	*F4*	*G4*	*A4*	*B4*
		.40	.07	0	0	1	.15	.10	.08	.07	1	0	0	.07	.08	.10

Table 2. State values of moving from 3rd scale degree to other pitches

This indicates, for example, that the value associated with a melody moving from the third scale degree (*E5*) up to the first scale degree (*C5*) is 0.07. In Appendix A, the 0.07 lies in the "3" column and the "↑1" row. The sum of the values for moving to each pitch do not sum to one. The reason that these values are not given as probabilities is that each of these values may be modified at a later point in order to improve the melodic generation. This process will be explained in detail later. Since it is not clear exactly what these values represent in their current form, they must now be mapped to a probability from 0 to 1. This is done with the following calculation:

$$0.4 + 0.07 + 1.0 + 0.15 + 0.10 + 0.08 + 0.07 + 1.0 + 0.07 + 0.08 + 0.10 = 3.12$$

The following probabilities result when each value is divided by the sum 3.12:

E5	→	*E5*	*C6*	*D6*	*E6*	*F5*	*G5*	*A5*	*B5*	*C5*	*D5*	*E4*	*F4*	*G4*	*A4*	*B4*
		.13	.02	0	0	.32	.05	.03	.03	.02	.32	0	0	.02	.03	.03

Table 3. Probabilities of moving from 3rd scale degree to other pitches.

These probabilities are then mapped as shown in Table 4 where the scale degree chosen is determined by the first pitch to have a mapped value greater than the random number generated. In other words, the probability distribution of Table 3 is converted to a cumulative probability distribution.

E5	→	*E5*	*C6*	*D6*	*E6*	*F5*	*G5*	*A5*	*B5*	*C5*	*D5*	*E4*	*F4*	*G4*	*A4*	*B4*
		.13	.15	—	—	.47	.52	.55	.58	.60	.92	—	—	.94	.97	1

Table 4. Mapped probabilities of moving from 3rd scale degree to other pitches

A random number is now generated to pick the next scale degree. For our example let us assume that the random number generated is 0.69. In that case *C5* will be the next pitch. The system will now check to make sure that a *C5* is in the appropriate range for a soprano. The system uses the convention established in Stefan Kostka's *Tonal Harmony* which limits the range of the soprano from *C4* to *G5*. Since *C5* is in this range, the pitch is accepted and the program is allowed to continue. Otherwise, the process is repeated until a valid pitch is chosen.

Next, the probabilities of repeating the same jump are modified to increase the chance of repetition. This is done by taking the square root of the previous state value. In this case the value associated with moving from an *E5* to a *C5* will change from 0.07 to $\sqrt{0.07} = 0.26$. This is the reason that the values in Table 2 cannot be given as probabilities initially.

This procedure is repeated for the rest of the melody with the exception of the beginning note of each new phrase. At these points, the previous note is always repeated.

Finally, the melody must be concluded so that:

1. The ending is convincing
2. An authentic cadence is set up

To do this, when there are two notes left in the melody a special function is called which determines the final two notes by choosing from acceptable endings that have been assigned a certain probability to be used. Figure 2 on page 13 shows the possible endings given the melody note on the first beat of the last measure.

The determining note of our example chorale is *C5*. From this, based on the probabilities of different endings, ending 1 from Figure 2 is chosen. Thus, our entire melody looks like Figure 3 on page 14.

Figure 2. Possible Endings.

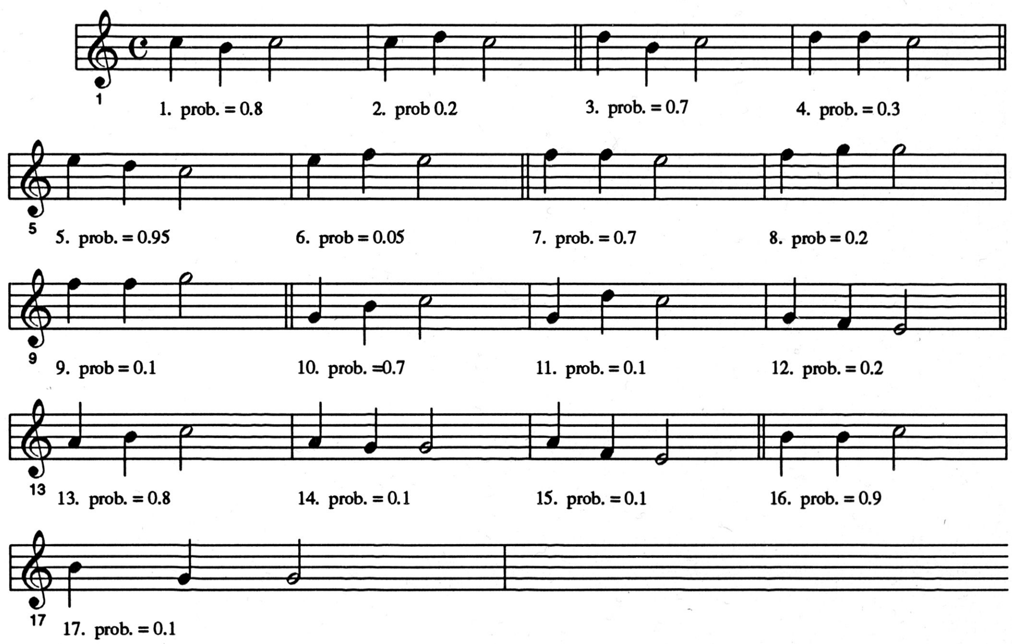

Figure 3. Chorale Melody.

Figure 4. Harmonized Melody.

Harmonization

The harmonization is always begun with a **I** chord. Following this, the current melody note is used to determine what chords are possible to move to. For example, in the chorale that is being created, the first chord would be a **I** to harmonize the *E5*. The next melody note after *E5* is a *C5*. The possible triads which have a *C5* as a member are: **I, IV, vi,** and **vii°/V**. Since the chorale is in the key of C, these chords correspond to **C, F, Am,** and **F#°**respectively. A number of Bach's chorales were analyzed harmonically to determine the probability of moving from one chord to another. These probabilities are listed in Appendix B. The previous chord in our chorale is **I** (**C**). This means that the probabilities for moving to the various chords are:

C	→	C	Dm	Em	F	G	Am	B°	A	B	D	F#	C#°	D#°	E°	F#°	G#°	A#°
		.096	.096	.250	.327	.125	.038	0	0	.038	.019	0	0	0	0	0	0	0

Table 5. Probabilities of moving from **C** to another chord.

However, only the four chords **C, F, Am** , and **F#°** have a *C* in their composition. The probabilities for changing from a **C** to one of these chords is .096, .25, .135, and .019 respectively. These probabilities must now be mapped from 0 to 1. The same method used for mapping the melody probabilities from 0 to 1 is used for the harmonizations. When this is done the following cumulative probability distribution results:

C	→	C	F	Am	F#°
		.192	.692	.962	1

Table 6. Mapped probabilities of moving from **C** to another chord.

Next, a random number is generated and the first chord whose mapped probability is greater than the random number is chosen. Suppose the random number generated is .150. This means a **C** will be chosen as the next chord. If at any time a situation arises in which there are no possible chords to choose from with a probability greater than 0, the system will back up one beat and choose another chord. This process is repeated until a cadence point is reached.

When a cadence point, which is not the end of the chorale, is reached, the system looks at the melody pitch at that point. If the melody notes are a *G*, *B*, or *D* a **G** chord is harmonized. If the melody consists of a *C* or *E* a **C** is chosen. And if the melody note is an *A* then **F** is chosen. If the probability of moving from the previous chord to the cadence chord is 0, the system backs up and picks a different chord for the previous chord. For our example chorale, an *E* is the note at the first cadence point. Therefore, it is harmonized with a **C** chord.

The last two chords of chorales are always **V-I** which constitutes an authentic cadence and gives the chorale a feeling of conclusion. If the chord preceding the authentic cadence has a probability of moving to a **V** chord of 0, then another chord whose probability of moving to a **V** chord is not 0 is chosen.

Later on the harmonization may be composed of 7th chords and altered bases, but for now the harmonizations are made only of triads. The harmonization of our chorale up to this point is shown in Figure 4 on page 14.

Compose Other Parts

The initial notes for the three lower parts are picked from various choices, given the starting melody note. After the first note for each part is determined the rest of the parts are composed using the following method:

Loop 1: Pick a bass note (*1st, 3rd, 5th,* or *7th* of chord)

  Determine necessary notes for inner parts

  Loop 2: Try all possible ways of writing those notes

    Assign a score based on counterpoint rules

  End Loop 2

End Loop 1

Choose the alternative with the best score

This process will now be explained in detail.

Pick a bass note

The second chord in our example chorale is a **C**, this means that the possible notes for the bass are *C, E, G,* and *B* (*B* is the 7th of **C**). So, first the root (*C*) is chosen.

Determine necessary notes for inner parts

In every case after the melody and the bass notes are chosen the pitches necessary to complete the triad will fit into two scenarios:

1. Two pitches which must be used
2. One pitch which must be used, and any other pitch but the *5th* of the chord.

In our example chorale since the melody note and the bass note are both the root of the chord, the third and the fifth must be used in the inner parts to form a complete triad. So, our example falls into the first scenario.

Try all possible ways of writing these notes

There are only two ways in which these notes can be written in our example.

1. *3rd* (*E*) in part 2 and *5th* (*G*) in part 3
2. *5th* (*G*) in part 3 and *3rd* (*E*) in part 2

However, for each of these ways, each pitch could possibly be written in one of two different octaves. Therefore the program must check each of these to see which alternative works best.

Assign a score based on counterpoint rules

To determine which is the best choice of many possibilities a scoring system has been incorporated into the program to pick the best alternative. The rules that are used and the penalties assigned to them are listed in Table 7.

Error	Value
1. Parallel motion among all parts	50
2. Doubling of leading tone	100
3. Omission of root	200
4. Omission of *3rd*	100
5. Parts separated by more than an octave (except for part 3 and bass)	100
6. Crossing parts (except part 2 and part 3)	200
7. Part 2 and part 3 cross	50
8. vii° not in 1st inversion	100
9. Parallel octaves	100
10. Parallel fifths	100
11. Direct octaves	70
12. Direct fifths	50
13. Not enough space between parts	100
14. Parts out of allowable range	200
15. Leaps that are too great (part 2 and part 3)	100
16. Leaps that are tritones	400

Table 7. Rules for counterpoint.

For each error found the score is summed to give a total error score. This score is further modified by what I call the "law of the nearest way." The law of the nearest way basically says that the less the parts change from one note to the next the smoother the voice leading will be and therefore the more convincing the chorale will sound. This law is only applied to parts 2 and 3. The melody and bass are excluded from this principle. For each chromatic pitch that the part changes by, 10 points are added to the error score for the alternative. For example, if part 2 changes from a *C4* to and *E4* this would constitute a score of 40 because the interval difference between *C4* and *E4* is 4 (*C#, D, D#, E*).

Choose the alternative with the best score

The best alternative is simply the one that receives the lowest score. In our example the best alternative will be the one that places a *G4* in part 2, an *E4* in part 3, and a *C3* in part 4. The rest of the chorale is then written in the same manner. The chorale that has been generated up to this point is shown in Figure 5.

Figure 5. Chorale without Non-chord Tones.

Add Non-chord Tone Eighth Notes

There are three different non-chord tones that the system knows how to add. These are:

1. Neighboring tones
2. Passing tones
3. Suspensions

Neighboring Tones

Neighboring tones are pitches that are either a scale degree up or down from two pitches that are the same. For example, in our chorale on the third and fourth beats of the fourth measure of part 1 there are two *E5*s next to each other. A neighboring tone could be added as an eight note on the *and* of the third beat as either an *F5* or a *D5*. In this case, a *D5* is chosen (see Figure 6 on page 24). To add neighboring tones, the system first checks to see in which locations it is possible to add neighboring tones. They are then added approximately as often as Bach used them in his chorales.

Passing Tones

Passing tones occur when two notes are separated by a third and an eighth note is added in between the notes so that the pitches move by step. In our example chorale, such a scenario occurs in the melody on beats 1 and 2 of the first measure. In this case a *D5* can be added in between the *E5* and the *C5* (see Figure 6). Passing tones are added by checking in which locations it is possible to add them and then adding them approximately as often as Bach does in his chorales.

Suspensions

Suspensions are the final non-chord tones that the system adds to the chorales. A suspension occurs whenever two notes move down by step from one beat to the next. An eighth note of the same pitch as the first note is inserted on the next beat which then moves down by a scale degree on the *and* of the beat. For example in our sample chorale, on beats 3 and 4 of the second measure in the melody an *F5* on beat 3 moves down to an *E5* on beat 4. This is changed so that beat 3 remains an *F5*, but beat 4 is changed to an eighth note *F5* and the *and* of beat 4 is changed to an *E5* (see Figure 6). Suspensions are added by checking in which locations it is possible to add them and then adding them approximately as often as Bach does. The final chorale with non-chord tones added is shown in Figure 6.

Figure 6. Finished Chorale.

Results

The chorale in Figure 6 and the additional chorales in Appendix C are probably the best representation of the results of this project. The chorales generated are not errorless as far as the rules of chorale writing that theory students are taught. However, these rules are not absolute and it is not difficult to find instances in the chorales of Bach where these rules are broken. Perfect chorales could have been generated easily by having the program back up and try again every time it was unable to find an errorless alternative. However, this approach seems to violate the spirit of the project. I prefer the approach that allows the rules to be violated occasionally. Very rarely are there serious violations of the rules. Perhaps the best solution would be a compromise of the two approaches where the system backs up if the errors committed are greater than a certain score.

Certain enhancements could be made to the system to make the chorales more believable. These include the addition of other non-chord tones such as anticipations, appoggiaturas, escape tones, and neighbor groups. However, I think given the time constraints on this project and the fact that very little else would be learned by the implementations of these features, the three non-chord tones added are sufficient. Furthermore modulations within the chorales would be a nice, though complex, enhancement that would probably contribute much to the knowledge learned.

The most notable aspect of the system that needs improvement that can be noticed by the untrained listener is the random sound of the melody. It is fairly obvious, that the melodies of the chorales were not written by a very talented composer, while the harmonizations and other parts, on the other hand, are quite convincing. The impression I get when listening to the chorales is that a fairly competent music theory student was given a somewhat arbitrary melody by his instructor to be harmonized as an exercise.

Analysis

The final analysis of the project must investigate exactly what was learned about the chorale writing process in the development of the project. The most significant result which could have probably been guessed before the project was ever begun is that melody composition involves cognitive processes that are much more complex than just generating melodies based on the probability that one note will move to another. More research definitely needs to be done with this. It is my belief, after completing this project, that if a computer could be taught the difference between a convincing melody and one that sounds random, then chorales capable of fooling almost anyone could be generated quite easily. In fact, if the problem of composing convincing melodies is solved I believe the step from generating chorales to more intricate musical forms, such as fugues or sonatas, is entirely possible. I feel that this is the most significant insight into the chorale writing process that was gained by this study. If nothing else has been learned, the direction which future research must be done has been discovered. However, one must not discount the amount of difficulty that teaching a computer the difference between composing a convincing melody and a random melody could cause.

Another result that stems from the previous result is that the rules of chorale writing alone serve well to generate convincing chorales. It is not difficult to see that if the system is able to tell whether or not a melody is convincing, then this same knowledge could be applied to parts two, three, and four of the chorale. This would most certainly provide the knowledge necessary to compose chorales that could most likely fool even an expert listener.

Suggestions for Future Research

As already stated, the best place to direct future research would be in the development of convincing melodies. Since this project composed chorales, the emphasis of this project was on harmony. Therefore, additional research needs to be done on the aspects of rhythm. If significant discoveries in these two areas could be made, great progress could be made in the development of systems to compose music.

Conclusion

This project was successful in two major respects:

1. Chorales were successfully generated
2. The direction for future research is now clear

A major justification for the development of this system was that the results learned by teaching a computer how to compose music could provide significant insights into the teaching of composition to students. As a result of this project, it has been demonstrated that the rules currently taught to students are directly translatable to a computer and are indeed effective. This can be seen in the chorales generated by the system. However, it has not been determined that anything learned by working with a computer will be translatable to human students. This is largely a result of gaining no new real insights into the music composition process from this study. Of course by the same token, nothing was discovered which would cast doubt on the legitimacy of the supposition. Clearly, more research needs to be done in the area of computer generated music composition.

Appendix A

State Values for Melody

Scale Degree

	1	*2*	*3*	*4*	*5*	*6*	*7*
—	.4	.4	.4	.4	.4	.4	.4
↑1	0	0	.07	.08	.10	.15	1
↑2	1	0	0	.07	.08	.10	0
↑3	.15	1	0	0	.07	.08	0
↑4	.10	.15	1	0	0	.07	0
↑5	.08	.10	.15	.08	0	0	0
↑6	.07	.08	.10	.15	1	0	0
↑7	0	.07	.08	0	.15	1	0
↓1	0	1	.07	.10	.08	.07	0
↓2	0	0	1	.15	.10	.08	0
↓3	.07	0	0	1	.15	.10	0
↓4	.08	.07	0	0	1	.15	0
↓5	.10	.08	.07	0	0	1	0
↓6	.15	.10	.08	.07	0	0	0
↓7	1	.15	.10	0	.07	0	0

The horizontal axis represents the current scale degree. The vertical axis represents the next scale degree and whether it is moved up to or down to. "—" represents no change in pitch.

Appendix B

State Probabilities for Harmonic Progression

The vertical axis represents the current chord. The horizontal axis represents the probability of moving to that chord. The table is in two parts; the second part can be thought of as existing to the right of the first table.

	I	ii	iii	IV	V	vi	vii°	V/ii	V/iii
I	.096	.096	.019	.250	.327	.135	.038	0	0
ii	.273	.045	.045	0	.409	.136	0	.045	0
iii	0	.286	0	0	0	.286	0	.143	0
IV	.300	.050	.050	0	.500	.050	.050	0	0
V	.200	.039	0	.026	.132	.092	.013	0	0
vi	.148	.074	.148	.074	.222	.037	0	0	0
vii°	1	0	0	0	0	0	0	0	0
V/ii	0	1	0	0	0	0	0	0	0
V/iii	0	0	1	0	0	0	0	0	0
V/V	0	0	0	.900	0	0	0	0	.100
V/vii°	0	0	0	.143	0	.571	0	.143	0
vii°/ii	0	.667	0	0	0	0	0	.143	0
vii°/iii	0	0	1	0	0	0	0	0	0
vii°/IV	0	0	0	1	0	0	0	0	0
vii°/V	0	0	0	0	1	0	0	0	0
vii°/vi	0	0	0	0	0	1	0	0	0
vii°/vii°	0	0	0	0	0	0	0	0	0

	V/V	V/vii°`	vii°/ii	vii°/iii	vii°/IV	vii°/V	vii°/vi	vii°/vii°
I	.038	.019	0	0	0	0	0	0
ii	0	.045	0	0	0	0	0	0
iii	0	.143	0	0	0	0	0	0
IV	0	0	0	0	0	0	0	0
V	.107	.027	.027	0	0	.013	0	.013
vi	.259	.037	0	0	0	0	0	0
vii°	0	0	0	0	0	0	0	0
V/ii	0	0	0	0	0	0	0	0
V/iii	0	0	0	0	0	0	0	0
V/V	0	0	0	0	0	0	0	0
V/vii°	0	.143	0	0	0	0	0	0
vii°/ii	0	0	0	0	0	0	0	0
vii°/iii	0	0	0	0	0	0	0	0
vii°/IV	0	0	0	0	0	0	0	0
vii°/V	0	0	0	0	0	0	0	0
vii°/vi	0	0	0	0	0	0	0	0
vii°/vii°	0	0	0	0	0	0	1	0

Appendix C

The following 5 chorales were produced consecutively by the system. No screening was done before picking the chorales. Before the chorales are given the errors that the system detected while it was composing them are listed with the appropriate measure and beat.

One

none

Two

m. 2, bt. 1-2	Parallel motion error
m. 5, bt. 4	Not enough space between part 1 and part 2

Three

m. 6, bt. 1-2	Parallel motion error
m. 6, bt. 2-3	Part 1 and part 2 separated by more than an octave

Four

none

Five

m. 1, bt. 4 to	Parallel motion error
m. 2, bt. 1	
m. 8, bt. 1-2	Tritone leap in part 2
m. 8, bt. 2-3	Part 1 and part 2 separated by more than an octave
m. 9, bt. 1-2	Parallel motion error

One

Two

Three

Four

Five

References

Dallin, L. 1974. *Techniques of Twentieth Century Composition: A Guide to the Materials of Modern Music.* 3rd ed. Dubuque, IA: WM. C. Brown.

Forkel, J. N. 1970. *Johann Sebastian Bach: His Life, Art, and Work.* New York: Da Capo.

Hindemith, P. 1942. *The Craft of Musical Composition.* Book I. 4th ed. Translated by Arthur Mendel. Mainz: Schott.

Kennedy, M. 1980. *The Concise Oxford Dictionary of Music.* 3rd ed. Oxford: Oxford Univ. Press.

Kostka, S. and Payne, D. 1989. *Tonal Harmony: With an Introduction to Twentieth-Century Music.* 2nd ed. New York: Alfred A. Knopf.

Schenker, Heinrich. 1987. *Counterpoint.* Book I. Translated by John Rothgeb and Jürgen Thym. New York: Schirmer Books.

——. 1987. *Counterpoint.* Book II. Translated by John Rothgeb and Jürgen Thym. New York: Schirmer Books.

Tanimoto, S. L. 1990. *The Elements of Artificial Intelligence: Using Common Lisp.* New York: Computer Science.

Terry, C. S. 1929. *The Four-Part Chorals of J. S. Bach: With the German Text of the Hymns and English Translations.* London: Oxford Univ. Press.