Automatic Distillation of Musical Structures: Learning the Grammar of Music

Ofir Hermesh, Ofer Shacham, Keren Mendiuk, Ben Sandbank
School of Computer Sciences, Faculty of Exact Sciences
Tel Aviv University, Tel Aviv, Israel

Abstract

This paper explores the possibility of extracting the structures underlying music sequences in an unsupervised manner using similar methods as those used for analyzing natural language. We apply ADIOS, a grammar induction algorithm shown to perform well on natural language corpora, to a corpus of Israeli folk songs, exploring several possible textual representations of music. We show that one of these representations allows the algorithm to infer rich and complex structure, which is shown to be 'harmonically-meaningful'. It is our hope that the fact that the same algorithm can infer both natural language grammar and musical structure will shed light on the similarity between the two domains.

Introduction

Recent years have seen a rising interest in the claim that music follows 'grammatical' rules, similarly to language. Following Chomsky's analysis of natural languages using the formal techniques used for mathematical languages, it was suggested that there exist formal grammars to music (Steedman, 1999). This idea is based on several similarities between music and natural language.

In recent years, research has found similarities between the cognitive processes implicated in processing linguistic and musical sequences. Deutsch (1980) showed that musical materials are processed by inferring sequence structures and alphabets in several hierarchical levels, in a similar way to context-free grammars commonly proposed for natural languages. Patel (1998) found that a common ERP (event related brain potential) component occurs during the processing of a grammatically incorrect sentence as well as when a badly structured musical phrase is encountered. From a developmental viewpoint, it was found that infants appear to learn the transition probabilities in tone sequences, in much the same way as they do in syllable sequences (Saffran et al. 1999).

Taking a different approach, other research has shown that the same conceptual tools used to describe grammars can be used to capture the structure of musical genres. For example, Steedman (1984) proposed a generative grammar for chord progression in jazz twelve-bar blues. Bell & Kippen (1992) presented a grammar for modeling Indian tabla-drums music. Steedman (1989) described a combinatorial categorical grammar for harmonic semantics of jazz music.

In this paper we focus on another aspect of the similarity between music and grammar, which has thus far received little attention, namely that of the acquisition of structure. Recently, (Solan et al, 2005) have introduced ADIOS, an unsupervised learning algorithm that is capable, given a corpus of sentences, of inferring the context free grammar underlying them. In experiments on natural language corpora ADIOS has been shown to accurately pass judgment on the grammaticality of unseen sentences and to successfully generate new grammatically well-formed sentences. In this work, we apply the same algorithm to a corpus of 146 Israeli folk songs and show that ADIOS can infer meaningful musical structures from this raw musical data. While this does not constitute evidence to support the claim that the understanding of musical structures is acquired in a similar way to language, the fact that the underlying structures in both domains can be inferred using the same statistical approach implies that they are not dissimilar.

The remainder of the paper is organized as follows. After briefly introducing the ADIOS algorithm, we turn to describe the corpus and the methods by which we convert music into sentences analogous to those of natural language, a prerequisite for using the algorithm. Then, we describe the experiments that we carried out and present their results and analysis. Finally, we present some of the structures distilled by the algorithm and analyze them from a musicological perspective.

The ADIOS algorithm

The ADIOS algorithm is an unsupervised learning algorithm that, given a set of strings, recursively distills from it hierarchically structured patterns. It relies on a statistical method for pattern extraction (MEX) and on structured generalization, two processes that have been implicated in language acquisition. It has been evaluated on artificial CFGs with thousands of rules, on natural languages as diverse as English and Chinese, and on protein data correlating sequence with function.

The input for the algorithm is a set of sentences. Each sentence is a sequence of terminals. The terminals are the atomic units of the data, and can represent words, amino acids, musical notes or whatever. The ADIOS algorithm iteratively searches for patterns and equivalence classes in the raw corpus. A pattern is a similarly structured sequence of terminals that reoccurs in the corpus. An equivalence class is a set of terminals that are interchangeable in a given position in a certain pattern. For example, given the sentences "This is a red chair", "What is a blue chair" and "I bought a green chair", the algorithm may extract the equivalence class {green, blue, red} and the pattern "a {green, blue, red} chair". During the iterative process, a newly formed pattern is regarded as a new atomic unit, and therefore can be used in the generation of new patterns and equivalence classes

together with other terminals. The output of the algorithm is a set of CFG rewrite rules, defined by the hierarchy of the patterns and equivalence classes discovered. These rules can be used in order to generate new sentences. In the given example, the algorithm can generate the sentence "I bought a red chair", which is a completely new sentence. For a detailed explanation about ADIOS algorithm, see to Solan et Al. (2005).

The Corpus

As input to the ADIOS learners, we used a corpus of 146 Israeli folk songs, 114 of these were taken from the "Speedy Composer" website (http://www.speedy.co.il/composer/). The Speedy Composer is an artificial neural network melody composer, developed by Uri Even-Chen (1999). Fifty additional songs were manually added.

Transforming music to text representation

As the ADIOS algorithm accepts a set of strings as its input, the first issue to be resolved is how music is to be converted into such a format. When compared to natural language, music representation poses several difficulties – musical data is continuous, and may be played in multiple metres and scales. In addition, more than several notes can be played simultaneously, as opposed to language where a single word is spoken at a time. To alleviate these problems, all songs were normalized to C Major/A Minor scales. Furthermore, a minimum resolution of half-beats was used, and each song was limited to having a single melody note and up to four chord notes in each half-beat. Lastly, only metres of 2/4, 4/4, ¾ and 6/8 were allowed.

However, even after the preprocessing phase there are still several possible approaches to representing music as text. Attention should be given to the amount of information encoded in the strings. By choosing to disregard some of the musical data, we can create a simpler corpus whose analysis will be easier, but on the other hand we may prevent the algorithm from inferring some of the underlying structures. Bearing this in mind, we designed three textual representations for music and experimented with them:

Bar Representation: Only the melody is included in this representation (the chords are ignored). A 'word' is defined for each bar of melody that appears in the corpus. Each word encodes the sequence of melody notes that appear in the bar- a letter for each half-beat (therefore, a sequence of two identical quarter-notes is encoded in the same way as a half-note). 'Sentences' are composed of n-bar phrases, where n is a parameter that is adjusted for best performance.

Note Representation: Again, only the melody line is included in the representation. Here, each word represents a single note, by encoding its pitch, its position in the bar and its length. Once again, every n bars constitute a sentence.

Note-Chord Representation: Both the melody and the chords are represented. In this case, each word represents a half-beat, including the pitches of the melody-note and the three chord-notes. Sentences are again composed of n-bar phrases.

Since this representation gave by far the best results, we will describe its details shortly. Figure 1 shows the first notes of a song. We look at the notes lengths and their pitches, as encoded in standard MIDI (Musical Instrument Digital Interface) representation. The first five half-beats are complete silence. So we have five words of 0-0-0-0 (one word for each half-beat). The sixth half-bit includes only a melody-note (with MIDI itch of 64), thus the sixth word is 64-0-0-0. However, in the seventh half-beat there is a melody note and a chord, so the seventh word is 76-45-48-52. We continue in the same manner and then transform each MIDI pitch to an ASCII letter. To conclude, our first words will be: aaaa aaaa aaaa aaaa aaaa kaaa wgjn wgjn xgjn. The notes in every n-bar phrase constitute a single sentence.

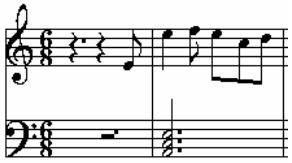


Figure 1: First notes of a sample song

Measures for Structures Acquisition Three heuristic measures were used in order to evaluate the quality of the proposed representations, based on the ability of ADIOS to learn from the corpus while using this representation. These three measures each give an estimate of the amount of structure inferred by ADIOS, though they cannot estimate the musical quality of the output, which should be analyzed manually. Only several phrases from the best representation were musically analyzed.

The three measures are: (1) Compression ratio - is defined as the ratio between the length of the corpus when encoded using the ADIOS-inferred patterns to the length of the original corpus. The more structure revealed by the algorithm, the smaller this ratio will become. (2) New sentences ratio - after each run of the algorithm we produce 1000 sentences according to the CFG that was distilled by it, and check how many of them do not appear in the training corpus. The new sentences ratio is defined as the ratio between the number of new sentences generated and the total number of generated sentences. A higher ratio implies better generalization. (3) Generalization ability - For each pattern, this is the ratio between the number of its possible instantiations that appear in the corpus and the total number of its possible instantiations. Somewhat counter-intuitively, a lower generalization ability score implies more structure was inferred by ADIOS. The generalization ability of an ADIOS learner is defined as the average generalization ability of the combined set of patterns.

Usually, the New Sentences Ratio and the Generalization Ability are strongly correlated. As ADIOS finds more generalizing and complex patterns, its output becomes more genuine.

Experiments

Several experiments were conducted for each of the representation schemes, and the quality of the results was compared.

Experiment Methods

A single ADIOS run was executed for each representation, and the three measures were assessed to determine which representation gives the best results.

In a second experiment a special "bootstrap" process was carried out, which has been shown in the past to improve the performance of ADIOS when run on a small corpus (Solan, personal communication). In this process multiple ADIOS learners are trained on the corpus and then used to generate a new bigger corpus. Then, another set of learners is trained on the new corpus and so on, leading to increased generalization.

Results and Analysis

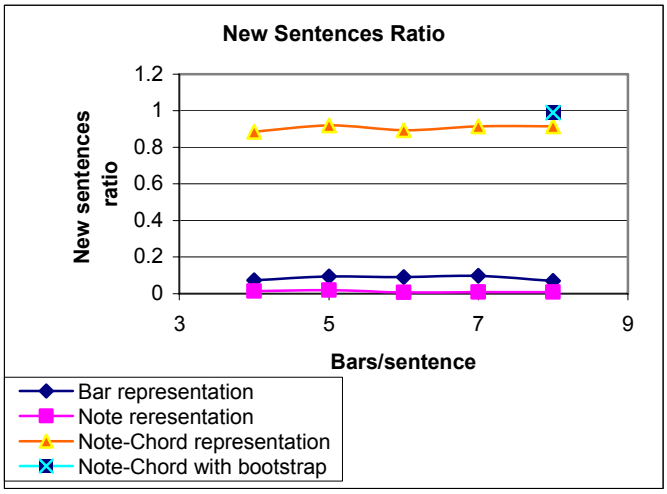
In the section, we present the results of our experiments and their analysis.

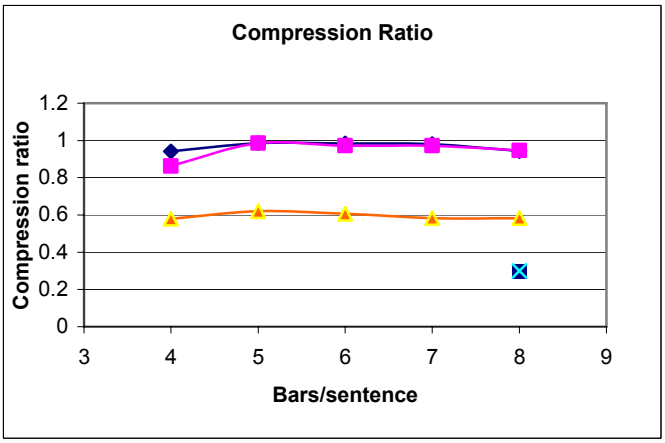
Experiments Results

In general, the best results were obtained for the Note-Chord representation. The quality of the results of the two other representations was very low, implying that ADIOS was not able to distill structures using these representations. In the following paragraph we describe the results of these experiments. The results for the three measures are also presented in Figure 2.

Bar Representation The results for this representation indicate that ADIOS could not find grammatical structures in the corpus. The generalization ability values of the different experiments were very close to 1, which implies that no real generalization took place. On closer inspection, it becomes clear that ADIOS was only able to produce patterns that recur a number of times within a single song.

One possible reason for the failure of this representation is the size of the corpus. The ratio between the corpus size and the lexicon size is 1:6, which is insufficient for ADIOS, since it almost cannot perform any alignment between the different occurrences of given patterns and form proper generalizations. Another reason may be that the representation does not give ADIOS access to the internal structures within the bars, since ADIOS can only find structures among words.





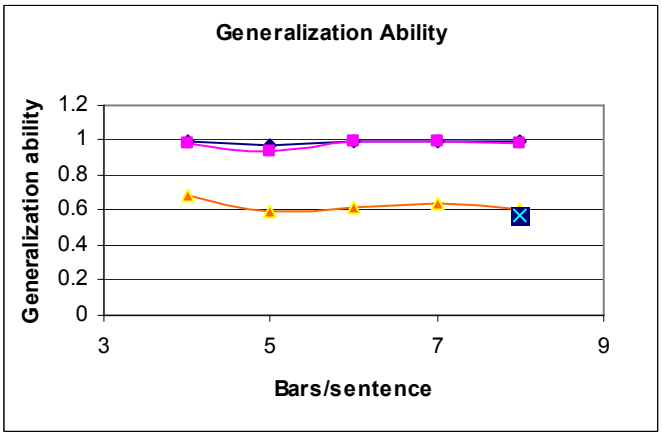


Figure 2: Comparison between the three representations, according to the three measures presented for different numbers of bars per sentence. The bootstrap experiment was performed only with 8 bars per sentence, and only this value is presented.

Note Representation The quality of the results is similar to those of the previous representation. Though the generalization ability and compression ratio are somewhat better than in the Bar representation, a closer look at the resulting 1000 musical phrases shows that most of them consist of only several long notes.
While this representation does allow access to the internal structures of each bar, this access is limited by the encoding of the in-bar position of each note. Consequently, if a sequence of notes appears at the beginning of one bar, and the same sequence reoccurs at the end of another bar, those two occurrences will appear as different sequences and could not be aligned by ADIOS.

Note-Chord Representation This representation provided by far the best results. Although the corpus is not big, ADIOS was able to reach an average generalization ability of 0.46, which means that half of the phrases that ADIOS generated were new. The compression ratio was also much better than the previous representations. Upon manual inspection, many phrases generated by the inferred CFG sounded new and harmonic.
A bootstrap run using this representation was also performed, and its results are presented in the graphs. The generalization ability is 0.38, which means that most of the phrases that ADIOS can generate are new. Almost all of the generated sentences were new and one can see that the compression ratio is low in comparison with the other experiments. This example demonstrates the benefits of the bootstrap process, as it enables ADIOS to infer patterns that are difficult to deduce from the relatively small corpus.
Figure 3 presents an example phrase generated by ADIOS in the second generation of the bootstrap run. This phrase sounds genuine and seems similar in spirit to other phrases from Israeli folk songs.

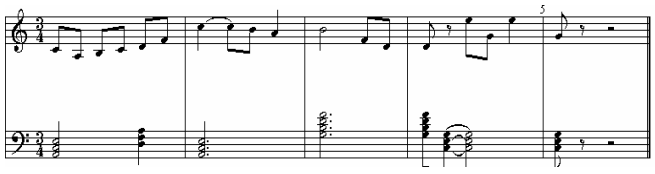


Figure 3: Example phrase generated by ADIOS using the Note-Chord representation

Musical Analysis of the Results

In this section, we provide a musical analysis of the patterns and equivalence classes inferred by ADIOS when using the Note-Chord representation with the bootstrap process.

Equivalence Classes First let us look at the simple equivalence classes, which contain terminals only. We can expect that ADIOS will group similar musical combinations in the same equivalence class. The most common equivalence class is that of the same chord with different melody notes from the chord (Recall that in this representation, each word consists of the chord notes and melody note). For example - the A-Minor chord appears with the melody notes C, E and with silence in one equivalence class. This kind of equivalence classes is quite expected in western-style music, since it is common to hear a musical phrase twice while a chord note is replaced with a different chord note.
A similar kind of equivalence class contains a seventh chord with the tonic note and with the seventh note, which sounds as "leading" to the tonic of the chord. For example, E7- Major chord with D and E as melody notes. ADIOS decides that together with a seventh chord, the melody note of the seventh note can be replaced with the tonic itself. This makes sense, as it is common to find a similar phrase in the middle of a song with a seventh-note, and then the same phrase at the end of the song, terminated by the tonic itself. This sounds like repetitive phrases, while the first phrase "leads" to the second one.
Another common equivalence class unites chords that have some notes in common. For example, A-Minor and F-Major, which have two notes in common. The melody note attached to both chords is A, which appears in both. It is expected to find such structures in a musical piece. Two chords, that are almost identical, can usually be replaced with little change to the music.
However, ADIOS also tends to find some disharmonic equivalence classes, which is not common in standard western musical theory. One equivalence class includes chords of C-Major, Bb7-Major and G7-Major, with melody notes from all of them. It seems that sometimes ADIOS is over-generalizing, but since musical combinations are very rich and diverse, it is possible that the algorithm finds structures that have some grammatical structure justification but cannot be identified easily as harmonic musical combinations.

Patterns We now turn to analyze some of the simple patterns found by the algorithm, which include only terminals. These patterns are simple to analyze but generally, they are less interesting than the equivalence classes. The most common pattern is that of a long note. In the Note-Chord representation, every half-beat is represented as a word. That means that if the music contains a note of two or three half-beats, it will result in a sequence of three identical words. It is very common to find notes that are longer than half-beat, so it becomes common to find sequences of identical words.

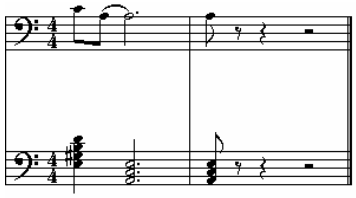


Figure 4: End-phrase pattern

Other interesting simple patterns are rare. Most of them are simply phrases that repeat themselves within a single song. Some, however, are more general and contain common musical phrases. Such a pattern is shown in Figure 4. This is a

typical pattern for an end of a musical phrase. It consist of a long tonic chord (A-Minor) preceded by dominant chord (E7-Major). The melody notes are the long tonic note of A and before it, a third (terza) note.

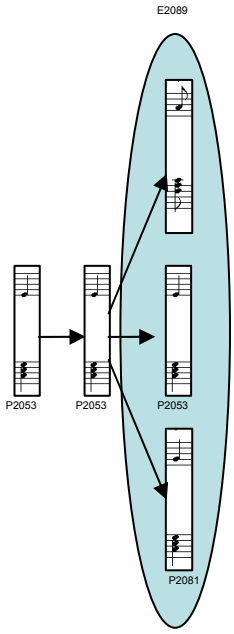


Figure 5: Pattern P2088

Complex Structures The interesting structures found by ADIOS are those composed of non-terminals, i.e. patterns and equivalence classes, as these may generate new unseen musical phrases. Here we show two complex patterns and analyze the musical structures they represent.

The first pattern we analyze is P2088, shown in Figure 5. The pattern is built from two P2053's, which is a two half-beat sequence of the C-Major chord with E melody note. Then, ADIOS can choose between another P2053 and P2081, which is the dominant of the key, or a short A note, which is probably a leading note to the next phrase. The pattern represents several structures that can end a musical phrase: (1) A long note (P2053 three times). (2) A long note leading to dominant (P2053 two times and then P2081). (3) A long note and then a short note leading to the next phrase (P2053 two times and then A).

The second pattern is P757, shown in Figure 6. This pattern always starts with a whole bar with a C-Major chord and some notes from the chord. The next bar (which is the next eight half-beats) is with F-Major chord, but there are many options for the melody. For example: F-F-A-F. The F-Major chord is the sub-dominant of the key (the note one tone below the dominant). The one-before last half-beat of this bar is chosen from E611. This can be a regular A note, which is part of the chord, or a Bb note, which leads to the next bar more smoothly. The third bar is again with a C-Major chord. The melody, however, has many options for sequences of notes from the chord. The last bar is with G7-Major chord, which is the dominant of the key. This is really a seventh chord. There are many options for the melody, built from the notes of the chord. The last note is E- leading to the next phrase. This should not be a last phrase in a sentence. This pattern can generate many nice phrases, which will have the common musical structure of 1-4-1-5 (tonic-subdominant-tonic-dominant), a structure that is very common in music in general and especially in folk songs.

Discussion

In this work, we suggested three ways of translating musical data into sentences for ADIOS. Several experiments were performed with these representations and the results were analyzed.

The results show that the Bar representation is not successful, probably since it does not provide ADIOS with information on the internal structure of the bars. The Note representation does not enable ADIOS to generalize two identical patterns in different offsets within a bar, and therefore was not successful as well. The main observation is that not only does the Note-Chord representation seem to be the best of the three; it seems to give good results in general. The success of the Note-Chord representation implies that it is possible to find grammatical structures in musical data in an unsupervised manner. The equivalence classes, patterns and complex structures that were analyzed show the similarity of ADIOS-inferred structure to known musical structures. The musical phrases generated by ADIOS usually sound harmonic and genuine, which reinforce our claim that the algorithm was able to find a meaningful context-free grammar representing the musical corpus.

Notably, ADIOS was able to find these musical structures without any knowledge about the real auditory properties of the data. The algorithm's only sources of information are the relations between the notes in the musical piece. This implies that harmonic musical structures can be inferred from raw

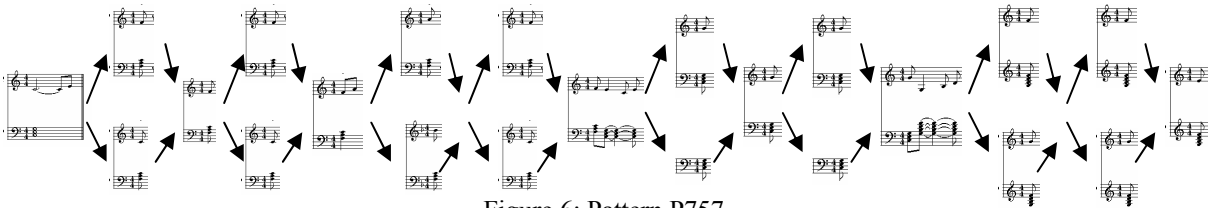


Figure 6: Pattern P757

musical data without any reference to the auditory properties of the notes played (e.g. their frequencies). It is surprising that the algorithm can learn harmonic structures only from their order of appearance in the musical piece, without any other knowledge on the notes themselves. Moreover, ADIOS has inferred these structures even though in the representation we used, the same chord when played with two different melody notes translated into two different, unrelated words.

A point that was not discussed in the text is that ADIOS seems to be able to handle only relatively short sentences (the best results in our experiments were obtained with 8- bar long sentences). When provided with longer sentences the generalization of the algorithm drops dramatically. We suspect this is the case because when it is supplied with long 'sentences', ADIOS has to infer the boundaries between musical phrases by itself, a task which it seems inept at. Moreover, while using short sentences the number of possible patterns reduces. For this reason, in this work we could only generate short phrases, and not anything approaching full songs.

Another problem when using ADIOS as a means of generating novel music is the following: In general, ADIOS performs textual replacements of parts in the sentences. It can generate a new sentence that is almost the same as a sentence in the training corpus, and just replace a couple of words (= notes). When the human ear hears a musical phrase in which several notes were replaced, it sounds as if ADIOS did not generate a new phrase at all. In this way, musical structures are different from textual structures. ADIOS is not designed in order to generate "the most original sentences", but to generate correct sentences. Similarly, ADIOS does not give the most original music, but just "correct" music. The relatively small size of our corpus is another cause to this lack of originality in the output, since ADIOS requires more data in order to generalize well.

Future work may include several extensions of the work presented here. First, the set of representations we experimented with is by far incomplete, and other representations should be designed and tested. An interesting experiment might be comparing the Note-Chord representation with a similar one, omitting the chord information, hence allowing a direct quantification of the contribution of the chords to the success of ADIOS. Second, manual subdivision of the musical pieces into sentences may lead to improved results. As explained before, in this experiment we applied arbitrary subdivision of the songs into constant-length sentences, probably breaking some of the structures to two separate sentences. A manual subdivision to musically structured sentences will be much more informative for the algorithm. We expect that a similar experiment with more complex inputs, such as classical music, will require such manual subdivision in order to give meaningful results. Third, our analysis of the harmonic structures discovered by ADIOS was based on our personal observation only. A more thorough analysis and judgments of 'harmonic-correctness' by a group of professional musicians seems in order. As the performance of ADIOS depends on the size of its input, effort should also be made to extend the corpus of songs. Similar experiments could be performed on other kinds of musical genres, e.g. classical music, where large amounts of pieces are readily available. Lastly, ADIOS may be provided with harmonic musical information by predefining proper equivalence classes which include 'harmonically compatible notes'. This is equivalent to the 'semantically-supervised' mode of operation described in Solan et Al. (2005).

In conclusion, in this work we presented a corpus of raw musical data from which rich structures were shown to be automatically distilled. Moreover, the algorithm that was used for acquiring harmonic structures from musical pieces is the same as that used for grammatical structures in natural language corpora, implying that the same statistical cues may be used for the learning of syntax in both domains.

Acknowledgement: B.S. is supported by the Yeshaya Horowitz association through the center of Complexity Science.

References

Bell, B., Kippen, J. (1992). Bol processor grammars, *Understanding music with AI: perspectives on music cognition*, pp. 366-400, MIT press.

Bod, R. (2002). A unified model of structural organization in language and music. *Journal of artificial intelligence research, 17*, 289-308.

Deutsch, D. (1980). The processing of structured and unstructured tonal sequences. *Perception & Psychology*, *28*, 381-389.

Deutsch, D., Feroe, J. (1981), The internal representation of pitch sequences in tonal music. *Psychological Review*, *88*, 503-522.

Even-Chen, U. (1999), *Artifical neural network melody composer*, http://www.sppedy.co.il/composer.

Hopcroft, J., Motwani, R., Ullman, J. (1979), *Introduction to automat theory, languages and computation*. Addison-Wesley.

Huron, D. (1997), Humdrun and Kern: selective feature encoding, *Beyond MIDI: the handbook of musical codes*, pp. 375-401, MIT press.

Patel, A., Gibson, E., Ratner, J., Besson, M., Holocomb, P. (1998). Processing syntactic relations in language and music: an event-related potential study. *Journal of Cognitive Neuroscience, 10:6*, 717-733.

Saffran, JR., Johnson, E., Aslin, N., Newport, E. (1999). Statistical learning of tone sequences by human infants and adults. *Cognition, 70*, 27-52.

Solan, Z., Horn, D., Ruppin, E., Edelman, S. (2005), Unsupervised learning of natural languages, *Procedures of the National Academy of Sciences*, *102*, 11629-11634.

Steedman, M. (1984). A generative grammar for Jazz chord sequences. *Music Perception, 2*, 52-77.

Steedman, M. (1989). Grammar, interpretation and processing from the lexicon, in W. Marslen-Wilson (ed.). *Lexical representation and process*, MIT press.

Steedman, M. (1999). The blues and the abstract truth: music and mental models. In Garnham, A., Oakhill, J. (eds), *Mental Models in Cognitive Science*, 1996.