# HARMONIC, MELODIC, AND FUNCTIONAL AUTOMATIC ANALYSIS

*Plácido R. Illescas, David Rizo, José M. Iñesta*
Dept. Lenguajes y Sistemas Informáticos
Universidad de Alicante, E-03071 Alicante, Spain
placidoroman@gmail.com {drizo,inesta}@dlsi.ua.es

## ABSTRACT

This work is an effort towards the development of a system for the automation of traditional harmonic analysis of polyphonic scores. A number of stages have been designed in this procedure: melodic analysis of harmonic and non-harmonic tones, vertical harmonic analysis, tonality, and tonal functions. All these informations are represented as a weighted directed acyclic graph. The best possible analysis is the path that maximizes the sum of weights in the graph, obtained through a dynamic programming algorithm. The feasibility of the proposed approach has been tested on J. S. Bach's harmonized chorales. The results so far are encouraging.

## 1. INTRODUCTION

The musical analysis is a means to better understand the thought of the composer when composing a piece. A musician must perform a good musical analysis to execute a correct interpretation of a work. The player, as a communication channel from the composer to the listener must know the spirit the creator wanted to transmit in his composition. The musical analysis tries to discover that spirit.

The melodic, harmonic, and tonal function analyses are the basic elements in order to achieve an optimal musical analysis. The first one shows the stylistic characteristics of a note from a contrapunctal point of view. On the other hand, the harmony is the basis that maintains the whole musical structure on tonic chords, giving names to the different tonalities the piece travels through. Finally, the tonal and harmonic analyses evidence the chord functions in the musical work.

Harmony is an interrelation among chords. Each chord, in its relation with its neighbor chords has a tonal function, bringing stability or relaxation, tension or instability tension. The function determines the tension level the chord provides given its harmonic environment, conditioning the way the player must perform the chord: e.g. a sixth degree functioning as subdominant cannot be played the same way as a tonic. Traditional harmonic analysis representation (or Roman number analysis) with respect to the root of the chord, as taught in music theory courses, has the advantage of revealing the relation of a chord with respect to those surrounding it, this way permitting to faithfully interpret the idea of the composer.

Besides the interpretation, there are many applications of this automatic analysis to diverse areas of music: education, score reduction, pitch spelling, harmonic comparison of works [10], etc.

The automatic tonal analysis has been tackled under different approaches and objectives. Some works use grammars to solve the problem [3, 14], or an expert system [6]. There are probabilistic models like that in [11] and others based on preference rules or scoring techniques [9, 13]. The work in [4] tries to solve the problem using neural networks. Maybe the best effort so far, from our point of view is that of Taube [12], who solves the problem by means of model matching. A more comprehensive review of these works can be found in [1].

The are several motivations that lead us to develop the system described in this paper. Firstly, one of our aims, not reached yet, is to be able to analyze automatically a monody to integrate the functional analysis into other MIR systems we are working on. Other target is to build an e-learning system that helps in tonal harmony teaching. Finally, the analysis model can be used to help in automatic composition. Although none of the works listed above fulfills our requirements, there are many points our system shares with those previous systems. We have tried to solve some of the problems found in them.

Our objective is not only to obtain a high percentage of correct analyses, but also to describe in a human-readable way the reasons why the system has chosen an analysis as the best one.

The system described in this paper uses both a rule engine approach and a ranking method, along with a dynamic programming algorithm, to describe analytically a musical piece.

The complexity and nature of the variables that must be utilized to perform a musical analysis are different among genres. However, the general scholastic method can be explained using the harmonized chorals of Bach. The presented work is devoted to test the ability of the proposed approach and has been founded on the baroque period music rules and tested using some of those Bach chorals. With some variations of the rules, the current system could be also extended to other genres of the western tonal music.

## 2. METHODOLOGY

In order to analyze a musical piece the system performs the following steps.

Firstly, a melodic analysis is performed. The notes are tagged either as *harmonic tones* for those belonging to the current harmony at each time, or as *non-harmonic tones* for those ornamental notes. There can be notes with different possible melodic analysis that will be disambiguated later.

Secondly, a vertical analysis of the piece is performed: after segmenting each bar into a number of time windows, all possible chords are obtained from the notes in each individual window.

The third step chooses which of the 24 different tonalities are candidates to be the central key in each window, given the accidentals of the notes involved.

The fourth step has all the possible chords and tonalities for each window as input. From these data, a weighted acyclic directed graph (wDAG) organized by layers is built. Each layer represents a window. The nodes of the graph correspond to chords with tonal functions in a tonality. The edges of the graph contain the valid progressions between the nodes in successive layers, weighted according to the importance of those progressions in order to establish a tonality, e.g. perfect cadences are scored higher than half cadences.

The final step uses a dynamic-programming approach to compute the best path along the graph, discovering the best tonality and tonal function sequence. The output of this final step is the Roman numeral analysis along with a tonality segmentation. Having this harmonic information, those notes still having a multiple melodic analysis are disambiguated.

Next, each of these steps are described with more technical details.

### 2.1. Melodic analysis

The melodic analysis tags each note as harmonic or non-harmonic-tone (NHT) based only on the melodic information, leaving aside any harmonic vertical information. Since a given note can have multiple different analysis, a confidence value is also computed. The NHT tags are given based on the type of ornament: passing tone, neighbor, etc.

The *JBoss Drools* [1] rule engine framework has been used to implement a series of rules written by the authors based on the music theory applicable to the baroque period. Those rules are based on meter, pulse, duration, and pitch interval information.

Before detailing the rules some terms must be defined.

**Definition 2.1** $rd(n_a) = duration(n_a)/duration(beat)$

The relative duration function determines the ratio between the duration of a note, $n_a$, and the duration of a beat.

[1] http://labs.jboss.com/jbossrules

**Definition 2.2** $ratio(n_i) = \frac{rd(n_i)}{rd(n_{i-1})} \times \frac{rd(n_i)}{rd(n_{i+1})}$

The ratio function is used to compare the relative duration of a note with its next and previous notes.

**Definition 2.3** $pitchName(n_a)$

It defines the position of the name of the note $n_a$ in the ordered set $\{C, D, E, F, G, A, B\}$. It does not include any accidentals.

**Definition 2.4** $pitchClass(n_a)$

Order of the note name including its accidental in within the octave. (e.g. $pitch(E\flat 4) = 3$).

**Definition 2.5** $pitchInterval(n_a, n_b) = d.s$

It computes the pitch interval between two notes $n_a$ and $n_b$, where $d = pitchName(n_b) - pitchName(n_a) + 1$ and $s = pitchClass(n_b) - pitchClass(n_a)$ specified with a resolution of two decimals. A positive value indicates an ascending interval. For example, the unison interval $pitchInterval(n_a, n_a) = 1.00$ and the tritone or augmented fourth is 4.06.

**Definition 2.6** $prevI(n_i)$

The previous interval is the interval between a note $n_i$ and its predecessor. $prevI(n_i) = pitchInterval(n_{i-1}, n_i)$

**Definition 2.7** $nextI(n_i)$

The next interval is the interval between a note $n_i$ and its successor. $nextI(n_i) = pitchInterval(n_i, n_{i+1})$

**Definition 2.8** $subbeat(n_i)$

It is a boolean function that is true when the note onset does not match the exact position of the beat, otherwise is false.

**Definition 2.9** $strong(n_i)$

For quaternary meters a note is strong when its onset is located in the first or third beat of the measure. In ternary meters, it is strong if and only if it onsets in the first beat. For the compound meters, this function can be computed from these two situations.

**Definition 2.10** $tied(n_i)$

It is a boolean function that is true whenever the note $n_i$ is tied from the previous note using a prolongation tie and false otherwise.

A NHT is tagged that way depending on its location in strong or weak beats, its duration, and the intervals it defines from its previous note and to the next one. There are 38 different rules that can be summarized in a simplified version in Table 1 [2].

[2] The whole set of complete rules have been stored in http://grfia.dlsi.ua.es/cm .

| NHT kind | rules |
|---|---|
| appoggiatura | $strong \wedge prevI = 1.00$ $\wedge\, nextI \in \{-2.01, -2.02, +2.01\}$ |
| suspension | $strong \wedge tied \wedge prevI = 1.00 \wedge$ $(nextI \in \{-2.01, -2.02, +2.01\})$ |
| passing tone | $(\neg strong) \wedge (ratio \leq 1)$ $\wedge\, ((prevI \in \{-2.01, -2.02\})$ $\wedge\, (nextI \in \{+2.01, +2.02\}))$ $\vee\, ((prevI \in \{+2.01, +2.02\})$ $\wedge\, (nextI \in \{-2.01, -2.02\}))$ |
| neighbor tone | $\neg strong \wedge (ratio \leq 1) \wedge$ $((prevI \in \{+2.01, +2.02\} \wedge$ $nextI \in \{+2.01, +2.02\})$ $\vee\, (prevI \in \{-2.01, -2.02\} \wedge$ $nextI \in \{-2.01, -2.02\}))$ |

**Table 1**. Summary of melodic rules for a note $n_i$. The parameter $n_i$ of the functions has been omitted to reduce space. Actually, a passing tone starting in a weak beat or sub-beat is allowed using a low confidence.

Each rule in the whole set assigns a confidence to the harmonic tag selected from the following catalog: '$s$' for *sure*, '$h$' for *high*, '$m$' for *medium*, and '$l$' for *low*. This confidence value will be useful in next steps of the procedure.

The result of this process is a list of possible melodic analysis for each note as shown in Figure 1.

### 2.2. Chord extraction

For each of the bars, the duration of the shortest figure or rest is selected as the time resolution for that bar. Then, the bar is divided into windows of that duration for all the voices in the score. This way, the musical piece is sliced into a sequence $W$ of windows. For each window $w \in W$, a set $S_w$ of all notes sounding in it and a set $C_w$ of chords from all combinations of notes in $S_w$ are built.

Two adjacent notes of a chord $c_i \in C_w$ must hold the condition (1):

$$\forall\ note_{j-1}, note_j \in c_i\ , \quad pitchInterval(note_{j-1}, note_j) \in \{3.s, 5.s\} \tag{1}$$

A backtracking scheme computes all combinations of note names without octave in $S_w$ leaving out those orderings containing adjacent notes not holding that condition. The result of the backtracking leaves in $C_w$ the set of valid chords. See Figure 2 for an example.

Note that the backtracking process reorders all notes in the chords. This means that, even if the original positions of the notes suggest an inversion, this process deletes it and returns the root note of the chord to the lowest note position.

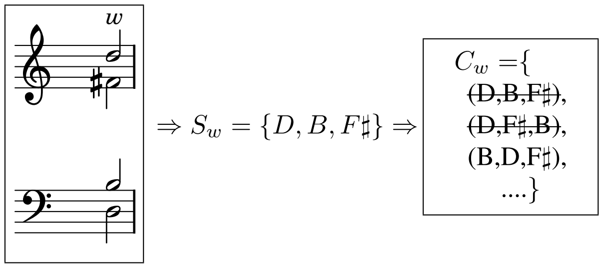


**Figure 2**. Chord extraction example. From the notes in a window (left), the set of note names is built (center) and all the combinations are computed (right) filtering out those not holding the condition (1) (struck out combinations).

### 2.3. Accidental analysis

The accidental analysis filters out from the 24 possible tonalities those that are not feasible given the accidentals of a set of notes. This filtering is performed for each window. The following definitions will be used to describe how this step of the analysis works.

**Definition 2.11** $mode(k) \in \{Maj, min\}$

The mode of a tonality is inherently in a mode, major or minor.

**Definition 2.12** $expectedSem(g, m)$

The values of this function are detailed in table 2 as the set of valid semitone intervals from the first degree of a scale to the specified degree $g$ given the mode $m$ of a tonality.

**Definition 2.13** $actualSem(n_i, k)$

Its value corresponds to the actual number of semitones from the tonic of $k$ to $n_i$.

**Definition 2.14** $isDiatonic(n_i, k)$

This boolean function is computed as

$$actualSem(n_i, k) \in expectedSem(deg(n_i, k), mode(k))$$

We introduce here the set $K_w$ of all valid tonalities for the set of notes in a given window, $S_w$. A tonality $k \in K_w$ if $isDiatonic(n_i, k)$ is true $\forall n_i \in S_w$.

| $expectedSem$ | I | II | III | IV | V | VI | VII |
|---|---|---|---|---|---|---|---|
| $Maj$ | 0 | 2,(1) | 4 | 5 | 7 | 9 | 11 |
| $min$ | 0 | 2,(1) | 3,(4) | 5 | 7 | 9,8 | 11,10 |

**Table 2**. Diatonic scales for the definition (2.12). The cells contain all the valid semitone differences from the tonic of the specified degree, the (1) value represents the Neapolitan, the (4) represents the Picardy ending.

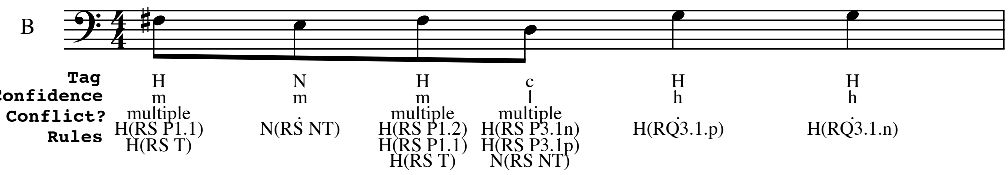


**Figure 1**. Melodic analysis example. 'H' is the tag for *harmonic*, 'N' for NHT, and 'c' for conflicts (ambiguities) during the analysis. Also the confidences and the utilized rules are shown for each note. The rule names belong to the whole set of rules.

### 2.4. Tonal functions

Music is based on a continuous flow of tensions and relaxations, of stability and stability. In tonal music, this is performed by adding dissonances to the tensions that are resolved to successive chords without dissonances (i.e., chords with tonic function). When passing from musical instability to stability, a transitory function named subdominant is included. Under our point of view, all notes, all chords, all music fulfill this principle. This way, a relation between tonal functions based on tensions and relaxations is established. Table 3 shows the relation between chord degrees and tonal functions. The degree of the chord is extracted using the function $g$ defined below in definition 2.16.

**Definition 2.15** $deg(n_i, k)$

It is defined as the degree of a note $n_i$ given a tonality $k$.

**Definition 2.16** $g_{c_i,k} = deg(root(c_i), k)$

The degree of a chord $c_i$ is equal to the degree of the root of the chord.

| Chord degrees | Tonal function |
|---|---|
| I | T |
| II | S |
| III | S,T,D |
| IV | S |
| V | D,T |
| VI | S,T |
| VII | S,D |

**Table 3**. Possible tonal functions for each degree. 'T': tonic, 'D': dominant, 'S': subdominant.

The V degree has a tonic function to allow half cadences that move the tonal relaxation to the dominant region. For the I, II, IV, and VII (altered in the minor mode) chord only one relation with a given tonal function is possible. III and VI can assume different relations depending on the harmonic neighbours.

#### Mediant chord

The III degree chord is the one with more possibilities of analysis. Depending on the tonality mode, the chord will have a different tonal function.

The mediant chord in a major tonality is minor perfect, where the fifth of the chord is the leading note of the tonality. A complete chord (including the fifth altered in minor mode) can be considered as dominant with a high level of confidence when the next chord is a tonic function. When the mediant chord is incomplete (it is a triad and its fifth is missing), if the previous chord is a dominant one, it can be tagged as tonic with a high level of confidence because the tension generated by the previous dominant resolves to that tonic.

In addition to the dominant and tonic functions, the third degree can behave as subdominant. To the best of our knowledge, this treatment is not found in the music theory literature, however, we attribute this function to it because we consider the tonal functions as levels of tension and musical stability. From this point of view, and considering three levels of tension, when the third degree is located between a tonic and a dominant, it should be given a subdominant transitory function.

#### VI chord

As stated above, the VI chord can be assigned two tonal functions depending on its harmonic surroundings. The VI degree will be a subdominant function when the previous chord is a tonic or a subdominant. In this case, the next chord has no impact in this decision. Otherwise, if the previous chord is a dominant function, this VI chord is considered as tonic.

### 2.5. Weighted Acyclic Directed Graph

As outlined in section 2.2, the piece is segmented in $|W|$ time windows and, from each window, $w \in W$, the sets of chords $C_w$ and valid tonalities $K_w$ are built.

Now, following the principles exposed in 2.4, the set $F_{w,k}(c_i)$ of all feasible tonal functions ($f$) for all $k \in K_w$ and all $c_i \in C_w$ is computed.

The wDAG is constructed using these data.

#### 2.5.1. Graph construction

Let's define the wDAG as $G = (V, E, D)$, where $V$ is the set of nodes and each $v \in V$ is labeled with the values provided by $F_{w,k}(c_i)$. Thus, we will use the labels (tonal functions) $f$ to represent the nodes, instead of $v$. $E \subseteq V \times V$ is the set of edges, and $D$ is the set of weights that are computed by a weight function $d : E \to \mathbb{R}$. The vertices are partitioned into $|W|$ disjoint subsets $V_i$, $1 \leq i \leq |W|$, in a way that, if $(f_a, f_b) \in E$, then $f_a \in V_i$ and $f_b \in V_{i+1}$. Each subset $V_i$ includes the nodes from a given window $w$. This way, the graph is structured as a sequence of layers, $V_i$, representing the course of time along the score. The definition of the weight function $d$ is explained below in the cadences section.

Figure 3 shows an extract of a graph for the tenth bar of the Bach's choral #25 [5](see score in Figure 4).

### Cadences

Cadences are points of relaxation that have the function of reaffirming the tonality. A cadence takes place when it concludes with the tonic function, and a half cadence when the target is a dominant or subdominant. Cadences and half cadences are summarized in Table 4.

| Cadence | Tonal funct. | Degrees |
|---|---|---|
| Perfect authentic | D→T | V→{I,i} |
| Imperfect | D→T | vii→I, vii$_d$→i |
| Interrupted | D→T | V→{vi, VI} |
| Plagal | S→T | {IV, iv, VI, vi, ii$_d$, ii} → {I,i} |
| Dominant Hc | T→D, S→D | {I,i,II,ii,III,iii,IV,iv,VI,vi} →{V,v,vii,vii$_d$} |
| Subdominant Hc | T→S, S→S | {I,i,ii,III,iii,IV,iv,VI,vi} →{II,IV,iv,VI,vi} |

**Table 4**. Cadences. 'Hc' stands for half cadence. The degrees in lower case represent minor chords. Subindex $d$ is applied to diminished chords.

The sequence of cadences and half cadences is a hint towards the recognition of a tonality in a given point. A sequence of chords analyzed from two different tonalities yields two different cadence / half-cadence successions. Weight information is introduced in the graph for the edge linking nodes $f_a$ and $f_b$ in adjacent layers by means of the weight function $d(f_a, f_b)$. The values for this function are displayed in Table 5, along with the relations between $f_a$ and $f_b$ in order to select the weights. The more conclusive a cadence is, the higher the weight is assigned to it. Each sequence of tonal functions will produce a different weight value. The path with the highest sum of weights will be selected as the most suitable one.

The values in Table 5 have been established empirically. A negative value is assigned to reflect the tonal regression D → S. Tonality changes are allowed only when a cadence in a new tonality is found. Not feasible progressions (those specified in section 2.4) are weighted with $-\infty$.

| From ($f_a$) | To ($f_b$) | Weight |
|---|---|---|
| T | D | 26 |
| T | S | 75 |
| T | T | 1 |
| S | D | 100 |
| S | T | 145 |
| S | S | 1 |
| D | S | -101 |
| D | D | 1 |
| D | T | |
| Perfect V triad + minor 7th | {I,i} | 2500 |
| Perfect V triad + minor 7th | {VI,vi} | 2100 |
| Perfect V triad | {I,i} | 1900 |
| Dim. vii triad + minor 7th<br>Dim. vii triad + dim. 7th | I<br>i | 2300 |
| Dim. vii triad | I | 1600 |
| Perfect iii triad + minor 7th<br>Aug. triad + with major 7th | I<br>i | 1550 |
| Perfect iii triad<br>Aug. III triad | I<br>i | 1500 |

**Table 5**. Relations between tonal functions within the same tonality and their corresponding weights. The weights for the relation D → T depend on the chords forming these tonal functions.

#### 2.5.2. Best path computation

Once the graph is constructed, the selection of the best path is reduced to the classical problem of the computation of the best path in a graph using dynamic programming [2]. The nodes visited in this path are taken for the best analysis. Those nodes appear filled in the example in Figure 3. The result, including tonal functions, degrees, and tonality changes is added as a new "Analysis" staff in the score (see figure 4, lowest staff).

### 2.6. Melodic analysis disambiguation

After the best path computation, the system has selected which chord, tonal function, and tonality are the best for each window. The first step of the analysis (the melodic analysis step) left some notes with conflicting rules unable to classify them in harmonic or non-harmonic tones. Now, having solved the harmony of the whole piece, those notes are tagged as NHT when they do not belong to the current chord at each moment.

## 3. EXPERIMENTS

As mentioned above, to test the system the harmonized chorals from J.S.Bach have been uused. The input of the

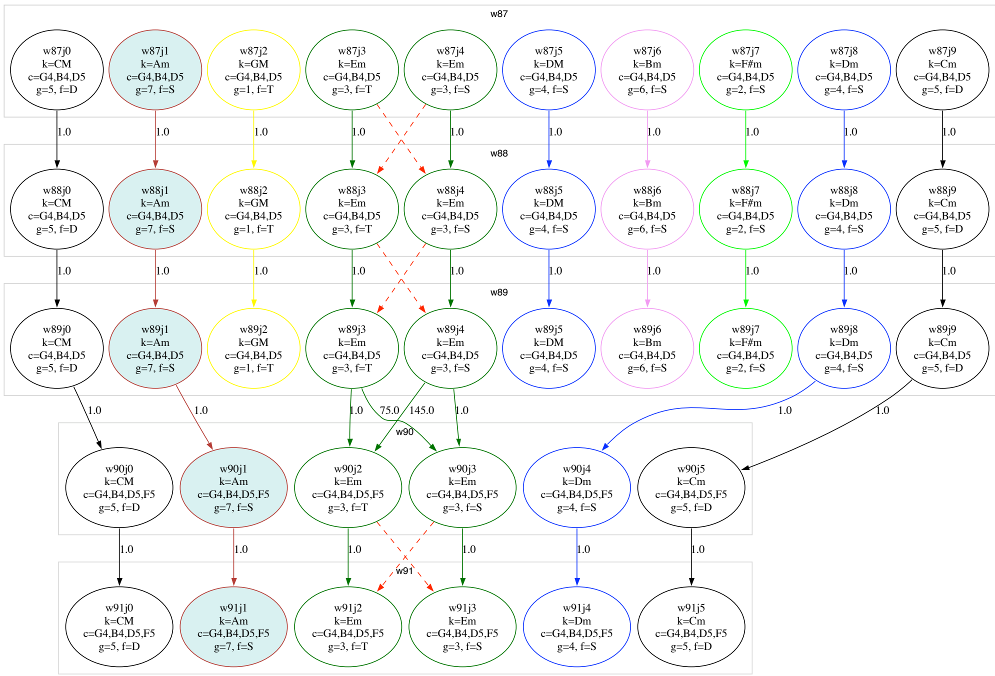


**Figure 3**. Graph example. Edges with weight 0 and those that imply a tonality change are not shown for clarity. Those edges with a $-\infty$ value are dashed lines.

system was transcriptions of the chorals in MusicXML format found at the Humdrum Kern Scores Site [3]. Due to the lack of a digitally annotated corpus of these chorals, we focused the analysis of results on the chorals: choral #25, BWV-0269, BWV-0367, BWV-0400, BWV 2-6. We did so in order to be able to compare our results to those of a relevant work named MTW (Music Theory Workbench) [12]. A full report of the obtained results can be downloaded from http://grfia.dlsi.ua.es/cm.

Since sometimes two different analyses can be both valid we cannot give success percentages in order to compare to those reported by MTW. For our point of view, the MTW fails in some tonal function progressions and seems to make mistakes when analyzing alternative tonalities by not solving the chord cadence (e.g. BWV 2-6 at bar 3, beats 2, 3, and 4). Our system corrects those errors by means of the cadence scoring. However, we must correct some errors the MTW does not make. Anyway, our system has failed only in two chords in the analysis of choral #25.

An extract of the output of the system can be found in the figure 4.

The system performs the analysis of each of the works in a few seconds, running in a Apple G4 at 1.33 GHz with 1.5 GB RAM.

## 4. DISCUSSION AND CONCLUSIONS

This paper has described a system to analyze automatically a score from the melodic and harmonic points of view, providing the tagging of each note as harmonic or not harmonic, the tonality changes and the Roman numeral analysis of each chord along with its tonal function.

This system works with polyphonic scores requiring both the correct names and accidentals of notes and meter information as inputs. Nowadays, MusicXML scores provide those informations. Other related works in the literature are able to work without meter [13], but our system is deeply rooted on that information that can be supplied by other subsystems, like [8] for meter or the ones reviewed in [7] for pitch spelling, if less structured information sources, like MIDI files, are utilized.

The presented system is comparable in results to those reported by MTW [12], but it has the advantage to be ready to work with monodic melodies only adding more possibilities of analysis at each layer of the graph.

We are working on it to compact the output so that the chords that have been split by the windowing process can merge again. Currently the system does not show the inversion of chords, the implementation of this feature is straightforward and is being developed. We have defined the conditions for the melodic tags: double neighbor tone, cambiatta, escape tone, and fux, and currently we are implementing them. That may correct some of the erroneous analysis the system outputs nowadays. Other important line of work is improve the merging of the melodic analysis with the harmonic steps of the process.

Modulation points are currently displaced in a $\pm$one chord distance. This is due to the dynamic programming algorithm. Hopefully, this problem will be solved by the use of a harmonic rhythm and a modulation subsystem.

Other work we are performing now is the construction

[3] http://kern.humdrum.net/

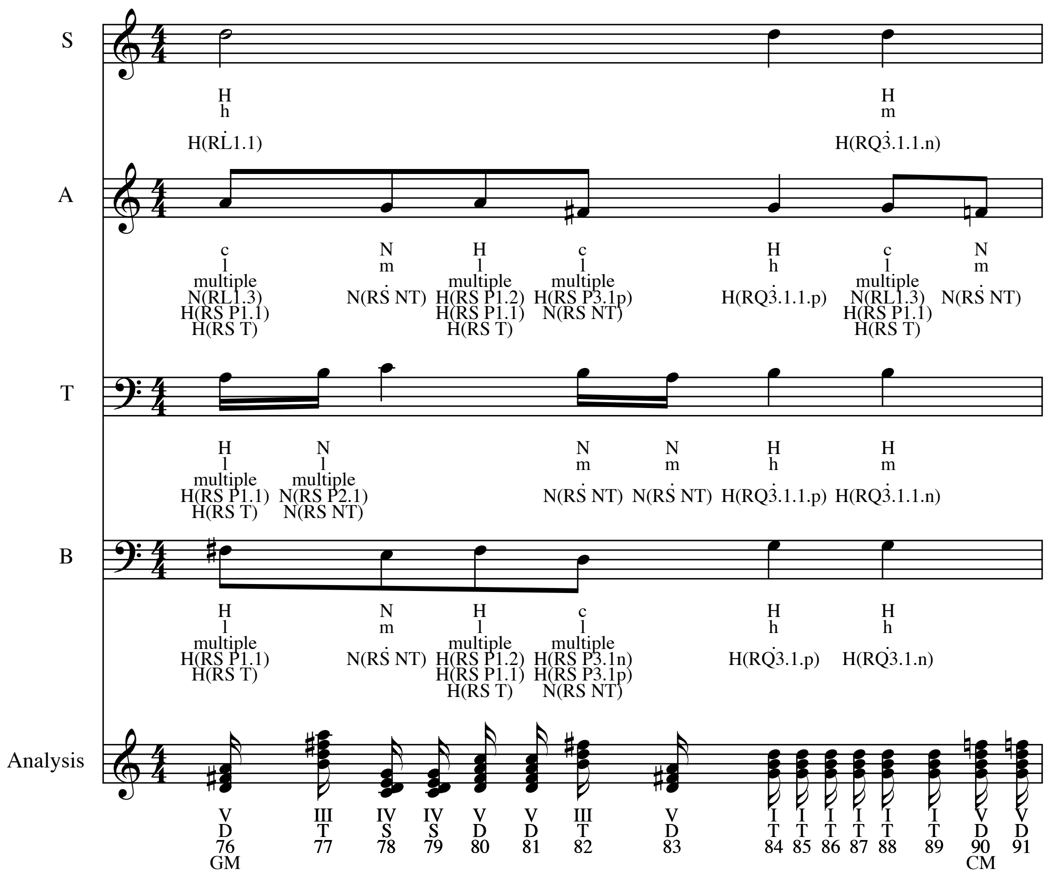


**Figure 4**. Analysis output. The first row under the 'Analysis' staff shows the tonal degree, the second one is for the tonal function, the numbers in the third row indicate the window number, and the tonality is shown in the fourth.

of a larger corpus of tagged pieces to be able to learn the scoring of the tonal function progressions from that corpus.

The next process to be tackled is the definition of new rules so that the system can be used in other genres. The romantic period seems to be the hardest one to be solved in academic music.

## 5. REFERENCES


[1] Jérôme Barthelemy. Figured bass and tonality recognition. In *ISMIR*, 2001.

[2] G. Brassard and P.Bratley. *Fundamentals of Algorithmics*. Prentice-Hall, 1997.

[3] Marc Chemillier. Toward a formal study of jazz chord sequences generated by steedman's grammar. *Soft Comput.*, 8(9):617–622, 2004.

[4] D. Hornel and T. Ragg. Learning musical structure and style by recognition. In *Proc. of the ICMC*, 1996.

[5] J.S.Bach. *389 Chorales-Jesange*. Breitkopf.

[6] H.J Maxwell. *An Expert System for Harmonic Analysis of Tonal Music*. MIT Press, Cambridge, MA, USA, 1992.

[7] David Meredith and Geraint A. Wiggins. Comparing pitch spelling algorithms. In *Proc. of ISMIR*, pages 280–287, 2005.

[8] Benoit Meudic. Automatic meter extraction from midi files. In *Proc. of JIM, Marseille*, 2002.

[9] Bryan Pardo and William P. Birmingham. Algorithms for chordal analysis. *Comput. Music J.*, 26(2):27–49, 2002.

[10] Anna Pienimäki and Kjell Lemström. Clustering symbolic music using paradigmatic and surface level analyses. In *Proc. of ISMIR*, pages 262–265, 2004.

[11] Christopher Raphael and Joshua Stoddard. Functional harmonic analysis using probabilistic models. *Comput. Music J.*, 28(3):45–52, 2004.

[12] Heinrich Taube. Automatic tonal analysis: Toward the implementation of a music theory workbench. *Comput. Music J.*, 23(4):18–32, 1999.

[13] David Temperley and Daniel Sleator. Modeling meter and harmony: A preference-rule approach. *Comput. Music J.*, 23(1):10–27, 1999.

[14] Terry Winograd. *Linguistics and the computer analysis of tonal harmony*. MIT Press, Cambridge, MA, USA, 1992.