MODELLING THE SYNTAX OF NORTH INDIAN MELODIES WITH A GENERALIZED GRAPH GRAMMAR

Christoph Finkensiep
EPFL*
christoph.finkensiep@epfl.ch

Richard Widdess
SOAS University of London
rw4@soas.ac.uk

Martin Rohrmeier
EPFL*
martin.rohrmeier@epfl.ch

ABSTRACT

Hierarchical models of music allow explanation of highly complex musical structure based on the general principle of recursive elaboration and a small set of orthogonal operations. Recent approaches to melodic elaboration have converged to a representation based on intervals, which allows the elaboration of pairs of notes. However, two problems remain: First, an interval-first representation obscures one-sided operations like neighbor notes. Second, while models of Western melody styles largely agree on stepwise operations such as neighbors and passing notes, larger intervals are either attributed to latent harmonic properties or left unexplained. This paper presents a grammar for melodies in North Indian rāga music, showing not only that recursively applied neighbor and passing note operations underlie this style as well, but that larger intervals are generated as generalized neighbors, based on the tonal hierarchy of the underlying scale structure. The notion of a generalized neighbor is not restricted to rāgas but can be transferred to other musical styles, opening new perspectives on latent structure behind melodies and music in general. The presented grammar is based on a graph representation that allows one to express elaborations on both notes and intervals, unifying and generalizing previous graph- and tree-based approaches.

1. INTRODUCTION

North Indian classical music (Hindustani music) provides valuable evidence for theories of syntactic musical organization. Like Western art music, it takes the form of aesthetic communication with an attentive and experienced audience, and is also a subject of theoretical discourse. Like most music outside the Western canon, it is normally unwritten, depending instead on memorization and improvisation. Instead of a system of chordal harmony or polyphony, Indian music comprises a solo melody against a complex background drone (of at least two pitches).

* École Polytechnique Fédérale de Lausanne

Melodic elaboration is prized as a means of musical extension and aesthetic enhancement: it operates at many levels, from the ornamentation of a single pitch, to the expansion of a phrase, to the architecture of a piece or performance. Melodic coherence is ensured by selecting one of a set of modes (rāga), each comprising a scale, a pitch hierarchy, and a set of licensed pitch transitions; any phrase that evokes a different rāga from the one selected is regarded as an error. It has been noted that Indian music resembles language in several respects [17], and a rāga could be understood as a melodic grammar, in which melodies are constructed by recursive elaboration over a hierarchically organized set of pitches.

The idea of understanding music in a hierarchical fashion goes back to Schenker [21], and has developed through the integration of impulses from generative linguistics and the theory of formal grammars since the 1980s [24, 1, 11, 20, 16]. Approaches most commonly addressed harmonic structure [23, 18, 19, 3, 5] and melody [4, 12]. Several approaches proposed simplified formalizations of Schenkerian theory and corresponding computational implementations [12, 13, 27, 9]. There is still comparably little discussion concerning the extent to which such hierarchical frameworks extend to non-Western forms of music. Narmours theory of melodic processes is explicitly directed to capture melodies outside the Western canon as well [15]. The application of Schenkerian methods to non-Western music has been discussed by Stock [25]. More recently, it has been proposed to adapt analytical tools from Schenkerian analysis and the GTTM to Indian music [14, 2].

This paper links with this discourse and proposes a generalized formal model of North Indian melodic and phrase structure. A common shortcoming in previous models of melodic elaboration is the treatment of leaps, which are usually either attributed to a latent harmonic structure that is assumed to be known [12, 10], or modelled as probabilistic intervals [4, 6] without explicit restrictions. This paper introduces a formalism for relating leaps in North Indian music to a latent tonal hierarchy that is stated explicitly. With respect to this hierarchy, leaps can be viewed as instances of generalized neighbor- and passing-note relations that take into account the stability of a pitch in a scale. As will be argued, the generalized neighbor idea applies beyond North Indian music to some degree.

A central question for elaborative models concerns the representation of the music. Since formal grammars – the standard formalism for recursive elaboration – operate on

Figure 1: Rāga Multānī with pitches in an approximate Western notation. The notated duration denotes the hierarchical level, i.e. relative stability, of each pitch; arrows indicate a constraint on the resolution direction of an unstable pitch.

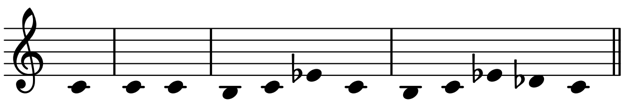


Figure 2: A short Multānī phrase and its derivation.

$d \in D_M$	1	♭2	♭3	♯4	5	♭6	7
$\delta_M(d)$	$\updownarrow$	$\downarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\downarrow$	$\updownarrow$
$\lambda_M(d)$	4	0	2	1	3	0	2

Table 1: A formal description of the rāga Multānī, showing the direction and hierarchical level of each scale degree (as shown in Figure 1). ♭2 and ♭6 are directed downwards and can therefore only be used before 1 and 5, respectively.

strings of objects, most models of musical elaboration represent music as a sequence of objects, such as notes or chords. As a consequence, these models mostly focus on melodic [4, 6, 12] or homophonic settings [8].

A desirable property of a formal grammar is that it is *context-free*, meaning that elaborations on a single object are independent from the objects around it. Systems that are based on strings of notes have problems with being context-free since some elaboration operations (such as passing notes) depend on two notes [11]. Because of this, more recent approaches have been based on strings of intervals [12, 27, 4], which allow elaborations of both single notes and pairs of notes while remaining context-free. However, in an interval grammar, notes are represented implicitly and redundantly (as part of an incoming and outgoing interval). In addition, all notes generated by elaboration are derived from two parent notes, which is unintuitive for single-sided operations. As a unification and generalization of both approaches, this paper suggests a graph-based representation in which both notes and intervals are represented explicitly, with a graph grammar describing the elaboration rules. This goes beyond descriptions of *derivations* as graphs, which is already an established practice [12, 27, 9].

2. MELODIC OPERATIONS

Melody in Indian music is based on a set of modes called rāgas. A rāga is not only a collection of pitches that may be used, it also establishes a hierarchy of stability among these pitches. Stable pitches are those that can serve as resting points, while less stable pitches tend to move towards their more stable neighbors. Some pitches in a rāga have a preferred resolution direction and must resolve to the closest pitch in that direction. An example of a rāga with its scale, tonal hierarchy, and directional constraints is shown in Figure 1. The relative stabilities indicated in Figure 1 is based on observation of normal practice in this rāga.

The melodic elaboration of a rāga is performed most completely and systematically (though not exclusively) in ālāp: a type of improvisation in which the scale and melodic features of the rāga are gradually exposed in phrases unfolding an arch-shaped trajectory, starting from the root (scale-degree 1) and reaching the octave above (or higher) before finally returning to the root (a process called *vistār* or "scalar expansion" [26]). This background structure is filled and elaborated recursively, generating a complex foreground melody. Elaboration follows mainly two principles, inserting either passing or neighbor notes.

Passing notes fill intervals that are larger than steps. They can occur close to the surface (such as the ♭2 in ♭3 ♭2 1), but can also be understood to characterize dependencies in the background (e.g., filling the top-level interval 1 - $1'$ with a 5). Two kinds of passing elaborations can be distinguished: Either a single note is introduced that subdivides the interval, potentially leaving non-step intervals that can be further elaborated; or the interval is filled with all scale notes enclosed by the interval.

Neighbor notes can be inserted before or after an existing note. While passing notes relate to both notes of an interval, neighbors are subordinate to single notes. When embellishing a note with a neighbor, a trade-off can be made between pitch proximity and stability: While unstable neighbors need to be very close to the main note's pitch, more distant neighbors can occur if they are sufficiently stable. In general, a pitch can only be perceived as a neighbor to some reference pitch if no pitch in the interval between the two is more stable than the proposed neighbor in the given mode.

Figure 2 shows the steps needed to derive a phrase using neighbors and passing notes. Starting with a single 1, the note is duplicated and elaborated twice, first with a lower neighbor 7, then with an upper neighbor ♭3. Finally, the space between ♭3 and 1 is filled with a passing ♭2.

3. MODES AND GENERALIZED NEIGHBORS

The idea of modes and generalized neighbors can be given a formal description: A *mode* M is a triple

$$\begin{aligned} M &:= (D, \delta, \lambda) \\ \delta: &\quad D \to \{\uparrow, \downarrow, \updownarrow\} \\ \lambda: &\quad D \to \mathbb{N} \end{aligned}$$

where D_M is a totally ordered set of *scale degrees*, δ_M is a function indicating the *direction* in which a scale degree is allowed to move, and λ_M returns the *hierarchical level* of a scale degree. For example, the rāga Multānī (Figure 1) would be formalized according to Table 1.

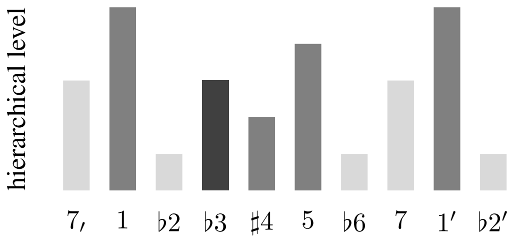


Figure 3: The upper and lower neighbors (dark) of ♭3 (black) in the Multānī rāga. Only pitches that can be reached without skipping a more stable pitch are neighbors. ♭2 is not a neighbor since it is directed downwards and can only be a neighbor to 1.

The same scale degree can be used as a *pitch* in different octaves, so pitches are indicated as scale degrees together with "′" for octaves above and "," for octaves below the default octave. The pitches of adjacent octaves are adjacent as well: $7_{,}$ is directly below 1 and $1'$ is directly above 7. As a result, a mode gives rise to a set of pitches P_M, which corresponds to $\mathbb{Z}$ while scale degrees correspond to $\mathbb{Z}_{|D|}$. For convenience, δ_M and λ_M are assumed to be defined on pitches as well and return the values of the corresponding scale degrees.

The set of pitches *between* a pitch p_1 and a pitch p_2 is the set of pitches in the open interval (p_1, p_2) that agree with the direction of the interval:

$$\Delta_M(p_1, p_2) = \begin{cases} \{p \in P_M \mid & p_1 < p < p_2 \land \delta_M(p) \neq \downarrow\} \\ & \text{if } p_1 < p_2 \\ \{p \in P_M \mid & p_1 > p > p_1 \land \delta_M(p) \neq \uparrow\} \\ & \text{if } p_1 > p_2. \end{cases}$$

The *neighbors* of a pitch $p \in P_M$ are then all pitches $n \in P_M$ that have a higher level than all pitches *between* p and n. In addition, the direction of n must agree with the direction from n to p:

$$\begin{aligned} nb_M(p) = \{n \in P_M \mid p \neq n \\ \land \forall q \in \Delta_M(n, p) : \lambda_M(q) < \lambda_M(n) \\ \land n \rightarrowtail p\}, \end{aligned}$$

where

$$n \rightarrowtail p = \begin{cases} \delta_M(n) \neq \uparrow & \text{if } p < n \\ \delta_M(n) \neq \downarrow & \text{if } p > n \\ true & \text{otherwise.} \end{cases}$$

Thus, every pitch is a neighbor to p only if it can be reached from the reference pitch without skipping a more stable pitch than the neighbor, as illustrated in Figure 3. Directed pitches can only be inserted as left neighbors since they must move towards their resolution.

When a single passing note is generated, the passing note must be a neighbor to both notes of the interval it is inserted in. However, in this case the inserted note is moving away from the first note, so the direction is not towards the reference note but towards the neighbor. A *reverse neighbor* $r \in rnb_M(p)$ is defined in analogy to a neighbor but with inverted direction:

$$\begin{aligned} rnb_M(p) = \{r \in P_M \mid p \neq r \\ \land \forall b \in \Delta_M(p, r) : \lambda_M(b) < \lambda_M(r) \\ \land p \rightarrowtail r\}, \end{aligned}$$

For example, a passing ♭2 in the sequence ♭3 ♭2 1 is a neighbor to 1 but a reverse neighbor to ♭3, as it is directed away from ♭3 and towards 1.

Finally, a *fill* is the list of all pitches between two pitches p_1 and p_2, sorted according to the direction of the interval (p_1, p_2) and restricted to pitches agreeing with that direction (as given by Δ_M).

$$fill_M(p_1, p_2) = \begin{cases} sort(\Delta_M(p_1, p_2), asc) & \text{if } p_1 < p_2 \\ sort(\Delta_M(p_1, p_2), desc) & \text{otherwise.} \end{cases}$$

4. A FORMAL GRAMMAR OF RĀGA MELODIES

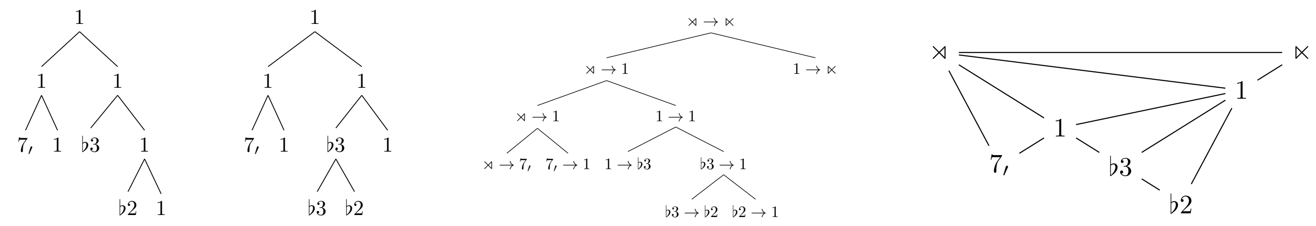
4.1 Representing Melodies as Graphs

As seen in Section 2, the two fundamental elaboration types – passing and neighbor notes – operate on two different musical entities: While neighbors elaborate single notes, passing notes fill intervals between two notes, elaborating both notes at the same time. As a consequence, two main formalisms describing hierarchical elaboration have emerged, note grammars and interval grammars.

Note grammars generate strings of notes, with derivation rules replacing single notes by several new notes. The resulting hierarchical structure is a tree of notes as shown in Figure 4a. However, elaborating single notes is problematic for passing notes, as they elaborate two notes. Not only is the resulting hierarchy ambiguous (the passing note must be attached to either its predecessor or its successor), but from a generative perspective, a passing note can only be derived from one of its parents. Thus, deciding where a passing note may be inserted becomes a context-sensitive problem.

Interval grammars [4, 10, 6, 12] solve the passing note problem (and two-sided operations in general) by elaborating pairs of notes, or intervals. Inserting a new note replaces an existing interval with two new intervals. The melody is then represented as a string of intervals with each note being represented twice, once as the second note of an interval and once as the first. To avoid this redundancy in notation, derivations are usually not given as trees (Figure 4b) but as outerplanar graphs (Figure 4c), giving each note two parents. However, for one-sided operations like neighbors, interval-based elaboration is conceptually misleading, as only one of the parent notes is considered while the other is ignored. This can lead to unwanted subordination of conceptually independent neighbors, as will be argued below.

As a unification and generalization of note- and interval-based systems, a graph-based representation of melodies is suggested here, representing notes as nodes

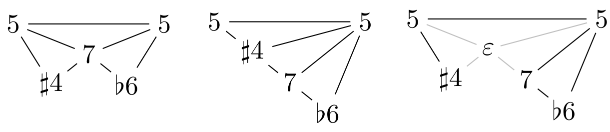


(a) Two analyses using note elaboration. The passing $\flat 2$ must be attached either to the 1 on its right or to the $\flat 3$ on its left.

(b) An analysis of the phrase using interval elaboration. The passing $\flat 2$ is generated in the interval $\flat 3 \to 1$.

(c) The same analysis as in 4b, displayed as an outerplanar graph.

Figure 4: Conventional formal analyses of the phrase in Figure 2.



(a) (b) (c)

Figure 5: Three possible derivations of $5\ \sharp 4\ 7 \flat 6\ 5$.

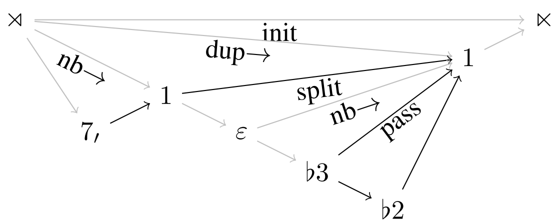


Figure 6: An analysis of the phrase in Figure 2 using the rāga grammar. Dark edges indicate the subgraph induced by removing non-note nodes ($\rtimes, \ltimes, \varepsilon$).

and note transitions as edges. Using graphs as the basis for elaboration has both conceptual and practical implications. Conceptually, graphs represent both notes and note transitions explicitly, which allows the use of both entities as a starting point for elaboration. Practically, while graphs can represent strings of objects (such as melodies) as a special case, they can easily describe much more complex structures, which potentially allows the description of elaboration operations in non-monophonic music. However, special cases such as monophony can still be defined graph-theoretically, ensuring consistency under elaboration. Thus, graphs provide a common framework for both melodic grammars and more complex formalisms.

While graphs in principle allow operations on both nodes and edges, a much simpler and more consistent system is obtained by operating only on edges, resulting in an *edge-replacement graph grammar*. All operations are then defined on edges (i.e., node transitions) with one-sided operations ignoring one node of the edge. One-sided operations still introduce an edge between the unused note and the new one in order to allow further elaboration between them. In order to express the independency between the new and the ignored note, a dummy node (written as ε) can be introduced first between any two notes. A dummy node does not generate a note and is analogous to the empty string in a conventional grammar.

Only strictly one-side operations can be performed on edges adjacent to a dummy node. This restriction expresses the independence between one-sided elaboration notes and their opposite side, and permits a more appropriate hierarchy: Suppose two one-sided neighbors are generated between two 5s, a $\sharp 4$, as a right neighbor to the first 5 and a 7 as a left neighbor to the second 5 with a passing $\flat 6$, resulting in $5\ \sharp 4\ 7 \flat 6\ 5$ (Figure 5). Without a dummy node, either 7 or $\sharp 4$ is subordinate to the other, depending on which is generated first (Figures 5a and 5b). By first introducing a dummy node, both neighbors can be derived independently (Figure 5c). Moreover, as dummy nodes are removed after the derivation, the resulting graph structure only retains edges that express elaboration dependence. Thus, dummy nodes allow the derivation to formally follow edge replacement while semantically expressing both one-sided and two-sided operations.

4.2 Formal Definition of the Grammar

A melody is formally represented as a directed linear graph with notes as nodes and transitions between notes as edges directed in time. The beginning and end of the melody are marked with the special nodes $\rtimes$ and $\ltimes$, respectively. The derivation is started from a single 1:

$$\rtimes \to 1_M \to \ltimes,$$

with 1_M indicating the root of mode M.

Derivation rules follow an edge-replacement paradigm: edges can be replaced with new subgraphs, retaining the nodes adjacent to the original edge. Some rules use only one of the adjacent nodes. In this case, a wildcard symbol ($* \in P_M \cup \{\rtimes, \ltimes, \varepsilon\}$) is used for the ignored node. The special symbol ε represents the *empty melody* and can be used to split an edge into two parts that may be elaborated independently. Only one-sided operations can be used on edges adjacent to an ε, $\rtimes$, or $\ltimes$.

For a given mode M the rāga grammar $\mathcal{G}_M^{r\bar{a}ga}$ is defined as the graph grammar $(\mathcal{T}, \mathcal{N}, \mathcal{I}, \mathcal{R})$ with

$$\mathcal{T} := \{n_1 \to n_2 \mid n_1 \in P_M \cup \{\rtimes, \varepsilon\}, n_2 \in P_M \cup \{\ltimes, \varepsilon\}\}$$
$$\mathcal{N} := \{\}$$
$$\mathcal{S} := \rtimes \to \ltimes$$

as terminals $\mathcal{T}$, non-terminals $\mathcal{N}$, and initial graph $\mathcal{S}$; and the following replacement rules $\mathcal{R}$:

initialize:
$(\rtimes \to \ltimes) \Rightarrow (\rtimes \to 1_M \to \ltimes)$

duplicate left: $\forall p \in P_M$:
$(p \to *) \Rightarrow (p \to p \to *)$

duplicate right: $\forall p \in P_M$:
$(* \to p) \Rightarrow (* \to p \to p)$

left neighbor: $\forall p \in P_M, n \in nb_M(p)$:
$(* \to p) \Rightarrow (* \to n \to p)$

right neighbor: $\forall p \in P_M, n \in nb_M(p) \wedge \delta_M(p) = \updownarrow$:
$(p \to *) \Rightarrow (p \to n \to *)$

passing: $\forall p_1, p_2 \in P_M, n \in rnb_M(p_1) \cap nb_M(p_2)$:
$(p_1 \to p_2) \Rightarrow (p_1 \to n \to p_2)$

fill: $\forall p_1, p_2 \in P_M$:
$(p_1 \to p_2) \Rightarrow (p_1 \to f_1 \to \ldots \to f_n \to p_2)$
where $f_1, \ldots, f_n = \mathit{fill}(p1, p2)$

split: $\forall p_1, p_2 \in P_M$:
$(p_1 \to p_2) \Rightarrow (p_1 \to \varepsilon \to p_2)$.

In this description, rules are given as templates that are instantiated for all (combinations of) pitches. A more elegant and efficient description is possible, if rules are considered to be functions on classes of structured symbols [7], allowing them to look inside their inputs.

Since the rāga grammar generates linear graphs, it is still possible to display derivations with outerplanar graphs. Figure 6 shows a derivation of the example phrase from Figure 2 using the rāga grammar. Each operation used to derive the phrase is written in the triangle formed by the old edge it replaces and the new edges it inserts. Later derivation graphs will omit operations and edge directions to remove visual clutter, as both are clear from the context.

In Figure 6, the ε inserted between the two 1s separates them and allows independent generation of neighbors. In particular, it would be possible to generate another right neighbor to the first 1 without subordinating it to the ♭3, or vice versa.

While the full derivation graph displays all derivation steps as they are formalized (i.e., as edge replacements), it does not distinguish one-sided and two-sided operations. Removing all non-note nodes ($\rtimes, \ltimes, \varepsilon$) and the adjacent edges induces a subgraph in which two-sided operations still use two edges while one-sided operations adjacent to non-note nodes only use one edge. The resulting graph resembles both note trees and outerplanar graphs in different regions, depending on the type of operation being used there. Thus, using ε nodes is an analytical option that reveals independencies between adjacent parts of the graph.

The graph grammar $\mathcal{G}^{rāga}$ is a special case of a graph grammar that is formally equivalent to a context-free grammar on strings of notes. Therefore, it can parse melodies efficiently. The context-free grammar can be obtained in

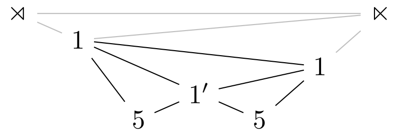


Figure 7: The spine modeling the deep structure of the octave expansion in an ālāp.

two steps: First, the graph representation is transformed to an interval representation in all parts of the grammar. Second, a set of rules is added for generating notes from intervals by taking the second note of each interval and generating empty strings (ϵ) where necessary.

In a directed linear graph, edges are totally ordered by their direction, so the graph can be transformed into a sequence of edges (e.g., $a \to b \to c$ becomes $(a,b)(b,c)$). Let e be the function that transforms linear graphs to sequences of edges. Then the context-free melody grammar $G(\mathcal{G}_\mathcal{M}) := (T, N, I, R)$ induced by a melody-graph grammar $\mathcal{G}_\mathcal{M}$ is defined as follows:

$$
\begin{aligned}
T_G :=& P_M \\
N_G :=& (P_M \cup \{\rtimes, \varepsilon\}) \times (P_M \cup \{\varepsilon, \ltimes\}) \\
S_G :=& e(\mathcal{S}_\mathcal{G}) = (\rtimes, \ltimes) \\
R_G :=& \{e(l) \Rightarrow e(r) \mid l \Rightarrow r \in \mathcal{R}_\mathcal{G}\} \cup \\
& \{(x,p) \Rightarrow p \mid p \in P_M, x \in P_M \cup \{\rtimes, \ltimes, \varepsilon\}\} \cup \\
& \{(x,n) \Rightarrow \epsilon \mid n \in \{\varepsilon, \ltimes\}, x \in P_M \cup \{\rtimes, \ltimes, \varepsilon\}\}.
\end{aligned}
$$

5. DISCUSSION

A main motivation for introducing generalized neighbors is that they allow modelling leaps in the background structure of North Indian music. Figure 7 shows the architecture of a typical *ālāp* in rāga Multānī. The melody slowly ascends from 1 to 1′ via 5 and returns back to 1 (again via 5). The upper 1′ can be seen as a very stable and distant neighbor of 1 while the 5s in between are (again stable and distant) passing notes. Each stage of this *spine* is then further elaborated by neighbors and passing notes, using increasingly less stable pitches and smaller intervals (Figure 8).

North Indian music is not the only style of music in which melodies are based on hierarchical modes. If other mainly mode-based styles also follow the elaboration principles of passing notes and neighbors, then the grammar defined in Section 4 should permit sensible analyses for these cases. Consider, for example, the melody of *Nun komm der Heiden Heiland* (based on the Dorian mode) and a phrase from the Jazz standard *Moanin'* (based on a Blues scale). Their respective derivations (shown in Figure 9) suggest plausible reductions of the surface melody in both cases. Moreover, the proposed relations between notes match the intuitions of generalized neighbors and passing notes.

A natural generalization of the mode-based approach is to consider the mode as a latent variable that can change over the course of the piece but still organizes elaboration

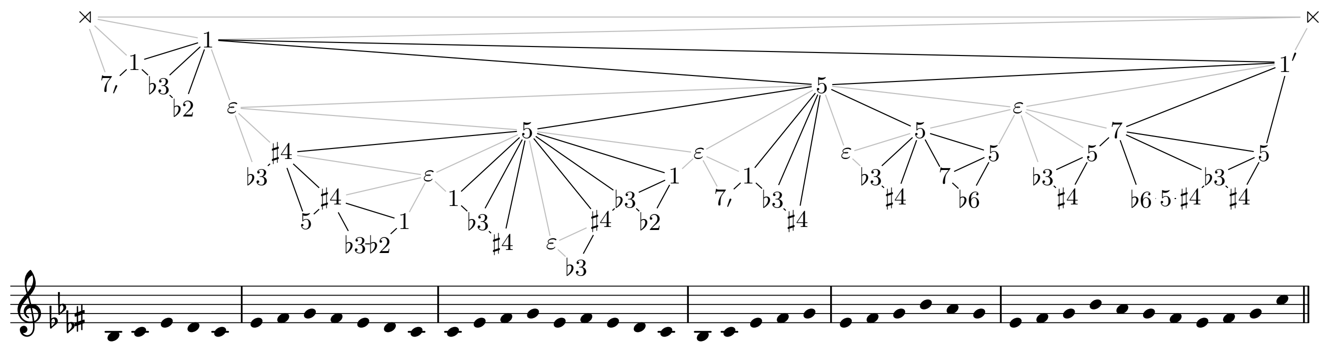


Figure 8: This example represents selected phrases, in order of performance, excerpted from the ascending part of an ālāp in rāga Multānī, recorded by the sitarist Dharambir Singh [22]. For reasons of space, one phrase has been selected for each of the stations between 1, 5 and 1′. Surface ornamentation and rhythmic durations have been omitted.

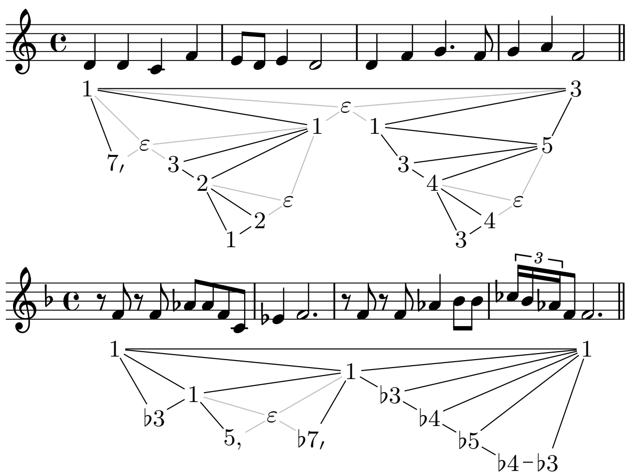


Figure 9: The first two phrases from *Nun komm der Heiden Heiland* and *Moanin'* (without repetitions), and their derivations based on a Dorian and a Blues scale, respectively.

locally. Depending on the style, this hierarchy can be constant over longer regions of the piece or change rather frequently. The latter case occurs when harmonies are considered as the latent structure, as they also define a tonal hierarchy, ranging from the root to non-chord notes.

However, there are two issues concerning generalized neighbor elaboration on harmonies. First, when the latent hierarchy changes, it is not obviously clear what should happen at the transition point. This is not an issue when these transitions are rare and elaboration across these boundaries is avoided. However, when harmonies take the role of the latent hierarchy, then transitions occur more often and elaborations frequently cut across harmonic changes. Moreover, as melodic elaboration happens on every level of reduction, it can even be considered to *generate* harmonic change in the background, such as the passing $\hat{2}$ in the *Ursatz*, generating a V harmony.

Second, not all leaps in melodies can be explained as generalized neighbors. The melody of *Take the A-Train* (Figure 10), for example, features several leaps which cannot be consistently explained as neighbors. While the ini-

Figure 10: The A part of *Take the A-Train* and a summary of its underlying lines.

tial G_4 and E_5 might be seen as neighbors to C_5, the descent to E_4 in the end is left unexplained by that. Instead, it is more plausible to assume a set of several independent lines: A higher line descends from E_5 to C_5, a lower line from G_4 to E_4, and an intermediate line that connects G_4 and C_5. Internally these lines behave according to elaboration principles (passing notes in this case), but the surface melody freely switches between the lines. This suggests that the organizing latent structure in this case is a set of implicit lines, although the elaboration and coordination of these lines might still be governed by a mode or a harmonic sequence as another layer of latent structure.

6. CONCLUSION

This paper proposed a generalized graph grammar formalism to model North Indian rāga music. We propose that passing and neighbor note elaborations are both necessary and (in their generalized form) sufficient operations of recursive rāga melody. This strengthens their status as fundamental musical principles across cultures. As the two operations are based on different objects (intervals and notes), models of elaboration should be able to represent both notes and intervals explicitly.

The notion of a generalized neighbor, based on a tonal hierarchy, shows that melodic leaps do not happen arbitrarily but can be related to a latent background structure. Understanding and modelling this background structure is necessary for a deeper understanding of melodic elaboration.

7. ACKNOWLEDGEMENTS

The research presented in this paper is generously supported by the Volkswagen Foundation and Claude Latour. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No 760081 PMSB. We thank Dharambir Singh for permission to use his recording of lp in rga Multn. We also thank the anonymous reviewers for their helpful feedback.

8. REFERENCES

[1] M. Baroni, S. Maguire, and W. Drabkin. "The concept of musical grammar". In: *Music Analysis* 2.2 (1983), pp. 175–208.

[2] D. Clarke. "North Indian Classical Music and Lerdahl and Jackendoff's Generative Theory-a Mutual Regard." In: *Music Theory Online* 23.3 (2017).

[3] W. B. De Haas, M. Rohrmeier, R. C. Veltkamp, and F. Wiering. "Modeling harmonic similarity using a generative grammar of tonal harmony". In: *Proceedings of the Tenth International Conference on Music Information Retrieval (ISMIR)*. 2009.

[4] É. Gilbert and D. Conklin. "A Probabilistic Context-Free Grammar for Melodic Reduction". In: *International Workshop on Artificial Intelligence and Music, IJCAI-07*. 2007.

[5] M. Granroth-Wilding and M. Steedman. "A robust parser-interpreter for jazz chord sequences". In: *Journal of New Music Research* 43.4 (2014), pp. 355–374.

[6] R. Groves. "Towards the Generation of Melodic Structure". In: *Proceedings of the Fourth International Workshop on Musical Metacreation*. 2016, pp. 1–8.

[7] D. Harasim, M. Rohrmeier, and T. J. O'Donnell. "A Generalized Parsing Framework for Generative Models of Harmonic Syntax". In: *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23-27, 2018*. Ed. by E. Gómez, X. Hu, E. Humphrey, and E. Benetos. 2018, pp. 152–159. ISBN: 978-2-9540351-2-3. URL: http://ismir2018.ircam.fr/doc/pdfs/258_Paper.pdf (visited on 04/12/2019).

[8] P. B. Kirlin and D. L. Thomas. "Extending a Model of Monophonic Hierarchical Music Analysis to Homophony". In: *Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR 2015, Málaga, Spain, October 26-30, 2015*. Ed. by M. Müller and F. Wiering. 2015, pp. 715–721. ISBN: 978-84-606-8853-2. URL: http://ismir2015.uma.es/articles/216_Paper.pdf (visited on 04/11/2019).

[9] P. B. Kirlin and P. E. Utgoff. "A Framework for Automated Schenkerian Analysis". In: (2008), p. 6.

[10] P. B. Kirlin and J. Yust. "Analysis of Analysis: Using Machine Learning to Evaluate the Importance of Music Parameters for Schenkerian Analysis". In: *Journal of Mathematics and Music* 10.2 (May 3, 2016), pp. 127–148. ISSN: 1745-9737, 1745-9745. DOI: 10.1080/17459737.2016.1209588. URL: https://www.tandfonline.com/doi/full/10.1080/17459737.2016.1209588 (visited on 11/26/2018).

[11] F. Lerdahl and R. S. Jackendoff. *A Generative Theory of Tonal Music*. MIT press, 1985.

[12] A. Marsden. "Representing Melodic Patterns as Networks of Elaborations". In: *Computers and the Humanities* 35.1 (Feb. 1, 2001), pp. 37–54. ISSN: 1572-8412. DOI: 10.1023/A:1002705506386. URL: https://doi.org/10.1023/A:1002705506386 (visited on 04/11/2019).

[13] A. A. Marsden. "Automatic Derivation of Musical Structure: A Tool for Research on Schenkerian Analysis." In: *International Conference on Music Information Retrieval*. 2007, pp. 55–58.

[14] S. Mukherji et al. "Generative Musical Grammar-A Minimalist Approach". In: *PhD Diss. Princeton University* (2014).

[15] E. Narmour. *The analysis and cognition of melodic complexity: The implication-realization model*. University of Chicago Press, 1992.

[16] M. Pearce and M. Rohrmeier. "Musical syntax II: empirical perspectives". In: *Springer handbook of systematic musicology*. Springer, 2018, pp. 487–505.

[17] H. S. Powers. "Language Models and Musical Analysis". In: *Ethnomusicology* 24.1 (1980), pp. 1–60. ISSN: 0014-1836. DOI: 10.2307/851308. URL: https://www.jstor.org/stable/851308 (visited on 04/11/2019).

[18] M. Rohrmeier. "Towards a generative syntax of tonal harmony". In: *Journal of Mathematics and Music* 5.1 (2011), pp. 35–53.

[19] M. Rohrmeier and M. Neuwirth. "Towards a syntax of the Classical cadence". In: *What is a Cadence?: Theoretical and Analytical Perspectives on Cadences in the Classical Repertoire*. Ed. by M. Neuwirth and P. Bergé. Leuven University Press, 2015, pp. 287–338.

[20] M. Rohrmeier and M. Pearce. "Musical syntax I: Theoretical perspectives". In: *Springer handbook of systematic musicology*. Springer, 2018, pp. 473–486.

[21] H. Schenker. *Der freie Satz Satz (=Neue musikalische Theorien und Phantasien 3)*. Ed. by J. Oswald. Vol. 3. Universal Edition, 1935.

[22] D. Singh. *Selected Alap Phrases in Raga Multani*. June 2019. DOI: 10.5281/zenodo.3258808. URL: https://doi.org/10.5281/zenodo.3258808.

[23] M. Steedman. "The blues and the abstract truth: Music and mental models". In: *Mental models in cognitive science*. 1996, pp. 305–318.

[24] M. J. Steedman. "A generative grammar for jazz chord sequences". In: *Music Perception* 2.1 (1984), pp. 52–77.

[25] J. Stock. "The Application of Schenkerian Analysis to Ethnomusicology: Problems and Possibilities". In: *Music Analysis* 12.2 (1993), pp. 215–240. ISSN: 0262-5245. DOI: 10.2307/854273. URL: https://www.jstor.org/stable/854273 (visited on 06/25/2019).

[26] D. Widdess. "Aspects of form in North Indian lp and dhrupad". In: *Music and tradition: essays presented to Laurence Picken*. Ed. by D. Widdess and R. Wolpert. 1981.

[27] J. Yust. "Voice-Leading Transformation and Generative Theories of Tonal Structure". In: *Music Theory Online* 21.4 (Dec. 1, 2015). URL: http://www.mtosmt.org/issues/mto.15.21.4/mto.15.21.4.yust.html (visited on 11/26/2018).