A Concatenation Operation to Derive Autosegmental Graphs

Adam Jardine and **Jeffrey Heinz**
University of Delaware
{ajardine,heinz}@udel.edu

Abstract

Autosegmental phonology represents words with graph structures. This paper introduces a way of reasoning about autosegmental graphs as strings of concatenated graph primitives. The main result shows that the sets of autosegmental graphs so generated obey two important, putatively universal, constraints in phonological theory provided that the graph primitives also obey these constraints. These constraints are the Obligatory Contour Principle and the No Crossing Constraint. Thus, these constraints can be understood as being derived from a finite basis under concatenation. This contrasts with (and complements) earlier analyses of autosegmental representations, where these constraints were presented as axioms of the grammatical system. Empirically motivated examples are provided.

1 Introduction

Autosegmental phonology represents words with graph structures. This paper provides a new way of defining the set of valid autosegmental representations through concatenating a finite set of graph primitives with particular properties. This 'bottom-up' approach to formalizing autosegmental representations (henceforth APRs) contrasts with the 'top-down', axiomatic approach of previous formalizations of APRs (Goldsmith, 1976; Bird and Klein, 1990; Coleman and Local, 1991; Kornai, 1995). However, we show that APR graphs constructed in the way we define hold to these axioms. One advantage to this perspective is that it brings out the *string-like* quality of APRs, in that they can be generated by the concatenation of a finite set of primitives. Furthermore, it shows that two putatively universal constraints, the Obligatory Contour Principle and the No Crossing Constraint (see below), are guaranteed to hold of autosegmental representations provided the graph primitives also obey these constraints. In other words, concatenation preserves these properties. Finally, the empirical generalization that languages may exhibit unbounded spreading but not unbounded contours is naturally expressed by this finite set of primitives, as spreading is derivable through concatenation but the only available contours are those found in the set of graph primitives. In short, important properties of autosegmental representations of words can be understood as being derived from a finite basis under concatenation.

Goldsmith (1976) originally defined APRs as *graphs*. Likewise, this paper models APRs using graphs representing both the associations and precedence relations of APRs. We apply established graph-theoretic methods to APRs, in particular *graph concatenation*, as defined by Engelfriet and Vereijken (1997). Engelfriet and Vereijken (1997) generate all graphs from concatenation and sum operations and a finite set of primitives. What is proposed here is a much weaker version of this idea, using concatenation only to build a specific class of graphs from a set of primitives. In doing so, it is shown how the properties of structures in the generated class derive from the operation and the primitives.

As detailed in the next section, there are several properties that most researchers agree are essential to APRs. One is that their composite autosegments

Proceedings of the 14th Meeting on the Mathematics of Language (MoL 14), pages 139–151,
Chicago, USA, July 25–26, 2015.

are divided up into disjoint strings called *tiers*, with associations linking autosegments on different tiers. Second, the No-Crossing Constraint (NCC) (Goldsmith, 1976; Hammond, 1988; Coleman and Local, 1991) states that these associations cannot 'cross'; i.e., they must respect the precedence relations on each tier. Finally, the Obligatory Contour Prinicple (OCP) (Leben, 1973) states that on the melody tier adjacent autosegments cannot be identical.

Formal treatments of these properties, starting with Goldsmith (1976), state these properties as axioms. For example, Bird and Klein (1990) provide a model-theoretic definition of APRs given a particular interpretation of association as overlap, and state axioms restricting the overlap relation. More recently, Jardine (2014) axiomatizes the NCC and one-to-one association in monadic-second order logic. Kornai (1995)'s treatment defines concatenation operations similar to the one given here, but his definition of APRs as *bistrings* does not derive from these operations. As a result, key properties like the NCC must be specified as axioms.

Instead, the current paper shows that the NCC and OCP can be derived by a concatenation operation alone, given a well-defined set of primitives. This paper is structured as follows. §2 details the set of properties phonologists deem important for APRs. §3 gives the relevant mathematical preliminaries, and §4 defines APRs as graphs and how the properties in §2 can be formalized as axioms. §5 defines a concatenation operation over graphs, and §6 proves how APR graphs derived using this concatenation operation obey the relevant axioms from §4. §7 then shows how to describe some common natural language phenomena using concatenation, as well as some phenomena that raise issues for concatenation. §8 reviews the advantages of viewing APRs through concatenation and discusses future work, and §9 concludes.

2 Basics of Autosegmental Phonology

Autosegmental phonology (AP) (Goldsmith, 1976; Goldsmith, 1979; Clements, 1976; McCarthy, 1979; McCarthy, 1985) has been a widely adopted theory of phonological representations in which phonological units, called *autosegments*, appear on one of some finite set of strings, or *tiers*, and related to autosegments on other tiers by *association*. Such *autosegmental representations* (APRs) are usually depicted with the tiers as vertically separated strings of symbols and the association relation shown as lines drawn between autosegments, as in (1) below.

(1)

The core insight APRs express is that a single autosegment on one tier may be associated to multiple autosegments on another tier, as in (1). For purposes of exposition, this paper focuses on two-tiered APRs: a *melody tier*, which carries featural information, and a *timing tier*, which represents how features on the melody tier are pronounced in the linear speech stream. For example, in tonal phonology, APRs often comprise a melody tier over the symbols $\{\mathrm{H}, \mathrm{L}\}$ for high and low tones and a timing tier over $\{\mu\}$ for morae (the timing unit most commonly associated with tone). The APR in (1c) thus represents a high-toned mora followed by a falling tone mora.

Thus, the insights of autosegmental phonology can be studied minimally with two-tier APRs, and so this paper focuses on two-tier APRs. However, in practice, APRs often use more than two tiers. As we explain at the appropriate points throughout the paper, the concepts discussed here can be straightforwardly applied to AP graphs with multiple tiers.

Two principles have been seen as crucial to constraining the theory of APRs. One is the No Crossing Constraint (NCC) (Goldsmith, 1976; Hammond, 1988; Coleman and Local, 1991), which states that if autosegment a is associated to autosegment y, no autosegment b which follows a on its tier may be associated to an autosegment x which precedes y. An example APR violating the NCC is given in (2a). The other principle is the Obligatory Contour Principle (OCP), which states that on each tier, adjacent autosegments must be different (Leben, 1973; McCarthy, 1986). The APR in (2b) violates the OCP.

(2)

Formal definitions of the NCC and OCP will be given in the following section, after we have defined

APRs explicitly in terms of graphs. The NCC is usually considered to be inviolable, where the OCP is considered violable by some authors (Odden, 1986). This paper treats the OCP as an inviolable principle, although this point is returned to in §8.

It is often, but not always, assumed that the sets of autosegments which are allowed to appear on each tier are disjoint. This assumption is usually adhered to in tonal and featural APRs, but not always in morphological APRs in which separate tiers represent separate morphemes (a la McCarthy (1979)). Here, we assume that the sets of elements allowed to appear on each tier are disjoint, and leave theories of APRs which allow a particular autosegment to appear on multiple tiers for future work.

3 Preliminaries

Let $\mathbb{N}$ represent the natural numbers. Given a set X of elements, a *partition* P is a set $\{X_0, X_1, ...X_n\}$ of nonempty subsets or *blocks* of X such that X is the union of these blocks and for each $X_i, X_j \in P$, $X_i \cap X_j = \emptyset$. P induces an equivalence relation $\sim_P$ over X such that for all $x, y \in X$, $x \sim_P y$ iff for some $X_i \in P$, $x \in X_i$ and $y \in X_i$. We also say $\sim_P$ *partitions* X into P. A partition P is said to *refine* another partition P' iff every block of P' is a union of blocks of P. We also say $\sim_P$ is then *finer* than $\sim_{P'}$. If R is a relation on X then let $\sim_R$ denote the *finest equivalence relation on X containing R*.

If Σ is a finite alphabet of symbols, then Σ^* denotes the set of all strings over that alphabet, including the empty string λ. We consider here alphabets structured by partitions. We refer to a partition $T = \{T_0, T_1, ..., T_n\}$ of Σ as a *tier partition* over Σ, and refer to some block T_i in T as a *tier alphabet*.

A *labeled mixed graph* is a tuple $\langle V, E, A, \ell \rangle$ where V is a set of *nodes*, E is the set of *undirected* edges, A is the set of *directed* edges (or *arcs*), and $\ell : V \to \Sigma$ is a total labeling function assigning each node in V a label in an alphabet Σ. For elements of the set V we will use early elements in $\mathbb{N}$. An undirected edge is a set $\{x, y\}$ of cardinality 2 of nodes $x, y \in V$, and a directed edge is a 2-tuple (x, y) of nodes in V. When not obvious from context, the elements of a graph G will be marked with subscripts; e.g., V_G. Let G_λ, the *empty graph*, refer to the graph $\langle \emptyset, \emptyset, \emptyset, \emptyset \rangle$.

Unless otherwise noted, all graphs in this paper are labeled mixed graphs, and thus will simply be referred to as *graphs*. All graphs are also assumed to be *simple* graphs without multiple edges; $\{x, y\} \in E$ implies $(x, y) \notin A$, and $(x, y) \in A$ implies $\{x, y\} \notin E$. Let $GR(\Sigma)$ denote the union of $\{G_\lambda\}$ with all graphs whose labels are in Σ.

A graph H is a *subgraph* of a graph G if $V_H \subseteq V_G$, $E_H \subseteq E_G$, $A_H \subseteq A_G$, and $\ell_H \subseteq \ell_G$. A subgraph H of G is an *induced subgraph* if for some subset X of V_G, $V_H = X$ and for all $x, y \in X$, $\{x, y\} \in E_G$ iff $\{x, y\} \in E_H$ and $(x, y) \in A_G$ iff $(x, y) \in A_H$. In other words, H has exactly the edges in G that appear between the nodes in X. We say X *induces* H and also write $G[X]$ for H. By a partition of G we refer to some set $\{G[V_0], G[V_1], ..., G[V_n]\}$ where $\{V_0, V_1, ..., V_n\}$ is a partition of V.

4 APRs as graphs

Here we define *autosegmental graphs* (APGs), or explicit graph representations of APRs. In this section, the set of valid APGs is defined axiomatically based on the phonological principles discussed in §2. In §6.2 we show that these principles can all be derived from graph concatenation. For an APG G, A represents the ordering relation on each tier, and E represents the association relations between them.[1]

We first define the tiers as subgraphs of G that are *string graphs* for which A represents the successor relation (Engelfriet and Hoogeboom, 2001).

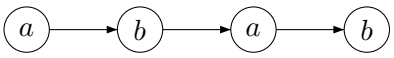

Figure 1: A string graph

Let $\preccurlyeq$ be the reflexive, transitive closure of A. That is, for any $x, y \in V$, if $x \preccurlyeq y$ then either $x = y$ or there is a directed path from x to y.

Definition 1 *A graph is a* string graph *if $E = \emptyset$ and its relation $\preccurlyeq$ is a total order on V.*

[1] It should be noted that linguists often leave the precedence relation on each tier as implicit or otherwise distinct from the model of associations (see Coleman and Local (1991) for an overview). However, with precedence directly in the graph, an APG represents all of the information in an APR, and thus this information can be studied by established graph-theoretic techniques, such as the graph concatenation considered here.

Let $\sim_A$ be the smallest equivalence relation that results from the symmetric closure of $\preccurlyeq$. The first axiom says $\sim_A$ partitions V into two tiers.

Axiom 1 *V is partitioned by $\sim_A$ into at most two sets V_0, V_1 such that $G[V_0]$ and $G[V_1]$ are string graphs. V_0 and V_1 are the* tiers *of G.*

The second axiom, related to Axiom 1, is that the partition of G into tiers respects some partition of Σ.

Axiom 2 *There is some tier partition $T = \{T_0, T_1\}$ over Σ such that ℓ forms a morphism from $\sim_T$ to $\sim_A$ such that $\ell(x) \sim_T \ell(y)$* iff *$x \sim_A y$.*

Axiom 2 corresponds to the principle discussed in §2 that each kind of autosegment may only appear on a particular tier. Note that a tier in G thus corresponds to a tier alphabet in T. For notational brevity, we mark this with matching subscripts; e.g., V_0 is the subset of V s.t. for all $v \in V_0$, $\ell(v) \in T_0$.

Axiom 3 governs the general form of associations.

Axiom 3 *For all $\{x, y\} \in E$, $x \not\sim_A y$.*

This simply states that the undirected edges, which again represent associations, must have one end in each tier. Thus, as noted by Coleman and Local (1991), the set of associations between two tiers in an APG forms a bipartite undirected graph $\langle V, E, \ell \rangle$ where the two parts are the tiers V_0 and V_1.

Having defined the structure of APGs in Axioms 1 through 3, we now define the NCC and OCP.

Axiom 4 (NCC) *For all $u, v, x, y \in V$, if $\{u, x\}, \{v, y\} \in E$ and $u \preccurlyeq v$, then $x \preccurlyeq y$.*

Similar axioms have also been defined by Bird and Klein (1990), Kornai (1995), and Jardine (2014).

Finally, Axiom 5 defines the OCP. Recall that the OCP only holds at the *melodic* level, so we choose only one of the tiers V_m for the OCP to hold.

Axiom 5 (OCP) *For one tier V_m, for all $x, y \in V_m$, $(x, y) \in A$ implies $\ell(x) \neq \ell(y)$.*

This concludes the axioms for APGs. For an alphabet Σ and tier partition $T = \{T_m, T_t\}$ over Σ, let $APG(\Sigma, T)$ denote the *class* of APGs obeying the tier partition T of Σ, where for each $G \in APG(\Sigma, T)$, ℓ maps elements in the tier V_m adhering to Axiom 5 to T_m.[2] §6 shows how to derive these axioms from the concatenation, as defined in the following section, of an alphabet of graph primitives with certain properties.

These axioms can be extended to graphs with more than two tiers. Instead of binary partitions, Σ and V could be partitioned into $\{T_0, T_1, ..., T_n\}$ and $\{V_0, V_1, ..., V_n\}$, respectively. In this case, Axiom 3 would specify a *single* tier in which all undirected edges must have one end. Axiom 5 would then hold for all tiers besides this tier. This results in 'paddle-wheel' APRs, like those defined by Pulleyblank (1986). Theories of feature geometry (Archangeli and Pulleyblank, 1994; Clements and Hume, 1995; Sagey, 1986) could also be accommodated for by positing additional structure on T. This, however, shall be left for future work.

5 Concatenation

This section defines a concatenation operation ($\circ$) based on that of Engelfriet and Vereijken (1997). Engelfriet and Vereijken (1997)'s operation merges nodes of graphs with specified beginning and end points; here, we use the tier structure to determine how the graphs are concatenated. We thus define $G_1 \circ G_2$ for two graphs G_1, G_2 in $GR(\Sigma)$ given a tier partition $T = \{T_m, T_t\}$ over Σ. The basic idea is to connect, if they exist, the last node of the first graph and the first node of the second graph for each tier. Such 'end nodes' with identical labels in the T_m tier alphabet are merged, whereas end nodes with labels in the timing tier alphabet and nodes with nonidentical labels in the melody tier alphabet are connected via a directed edge. As shown in §6.2 and §7, it is this 'merging' that derives both the OCP and spreading for APGs constructed this way.

As the concatenation operation is defined over graphs in $GR(\Sigma)$, it is at first very general and not of any phonological interest. However, we show in §6 that concatenation can be used to define a set of APGs that follow the axioms in §4, as shown in §6.2.

5.1 Definition

We assume that G_1 and G_2 are disjoint (i.e., that V_1 and V_2 are disjoint sets)—if G_2 is not disjoint with G_1, then we replace it with a graph isomorphic to G_2 that is disjoint with G_1.

We use two partial functions $\texttt{first} : GR(\Sigma) \times$

[2] Note that E is not required to be nonempty; that is, APGs with no association lines are allowed. This can model, for example, underlying APRs with unassociated melodies.

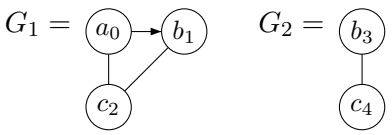

Figure 2: Two graphs in $GR(\Sigma)$

$T \to \mathbb{N}$ and $\texttt{last} : GR(\Sigma) \times T \to \mathbb{N}$ which pick out the first and last nodes on a particular tier in a graph.[3] Recall that V_i is the subset of V s.t. for all $v \in V_i$, $\ell(v) \in T_i$. Formally, $\texttt{first}(G, T_i) \stackrel{\text{def}}{=} v \in V_i$ s.t. $\forall v' \in V_i$, $v \preccurlyeq v'$ if such a v exists; otherwise it is undefined. Similarly, $\texttt{last}(G, T_i) \stackrel{\text{def}}{=} v \in V_i$ s.t. $\forall v' \in V_i$, $v' \preccurlyeq v$ if such a v exists; otherwise it is undefined. We shall sometimes refer to $\texttt{first}(G, T_i)$ (resp. $\texttt{last}(G, T_i)$) as the *first (last) node of G for tier alphabet T_i*.

Example 1 *Consider the alphabet $\Sigma = \{a, b, c\}$ and the tier partition $T = \{T_m = \{a, b\}, T_t = \{c\}\}$. Take graphs G_1 and G_2 where $V_1 = \{0, 1, 2\}$ and $V_2 = \{3, 4\}$ with edges and labeling as in Figure 2 Node indices are given as subscripts on the node labels.* $\texttt{last}(G_1, T_m) = 1$*, and* $\texttt{first}(G_2, T_t) = \texttt{last}(G_2, T_t) = 4$.

The concatenation operation combines the graphs, either merging or drawing arcs between the first and last nodes on each tier, depending on their labels. The operation can be broken down into multiple steps as follows. First, we define the graph $G_{1,2}$ as the pairwise union of G_1 and G_2. We denote $V_1 \cup V_2$ with $V_{1,2}$ and so on.

$$G_{1,2} = \langle \underbrace{V_1 \cup V_2}_{V_{1,2}}, \underbrace{E_1 \cup E_2}_{E_{1,2}}, \underbrace{A_1 \cup A_2}_{A_{1,2}}, \underbrace{\ell_1 \cup \ell_2}_{\ell_{1,2}} \rangle \quad (3)$$

Next, two binary relations over the nodes of $G_{1,2}$ are defined. R pairs the last element in G_1 and the first element in G_2 of each tier. R_{ID} is a restriction on R to pairs who share identical labels, excluding nodes whose labels are in T_t.

$$R \stackrel{\text{def}}{=} \{ \quad (v, v') \in V_{1,2} \times V_{1,2} \mid \quad (4)$$
$$v = \texttt{last}(G_1, T_i),$$
$$v' = \texttt{first}(G_2, T_i),$$
$$\text{for some } T_i \in T \quad \}$$

[3]That these are partial functions will be most useful for dealing with graphs with no nodes on a particular tier; for example the empty graph, which is discussed below.

$$R_{ID} \stackrel{\text{def}}{=} \{ \quad (v, v') \in R \mid \quad (5)$$
$$\ell(v) = \ell(v'), \ell(v) \notin T_t \quad \}$$

We also often refer to the complement of R_{ID} with respect to R; $R_{\overline{ID}} \stackrel{\text{def}}{=} R - R_{ID}$.

We can then use Engelfriet and Vereijken (1997)'s merging operation which reduces a graph G with any relation $R \subseteq V \times V$ over its nodes. Informally, nodes which stand in the relation are merged; everything else stays the same. Given any such relation R, we consider $\sim_R$, the finest equivalence relation on V containing R. In the usual way, let $[v]_{\sim_R} = \{v' | v \sim_R v'\}$. Here, we use $\sim_{R_{ID}}$, which assigns each node its own equivalence class, except for pairs $(v, v') \in R_{ID}$ of last and first nodes with identical labels, which are lumped together.

Example 2 *Continuing with G_1 and G_2 from Example 1, $G_{1,2}$ is given in Figure 3a. For $G_{1,2}$, $R = \{(1, 3), (2, 4)\}$, $R_{ID} = \{(1, 3)\}$, and so $R_{\overline{ID}} = \{(2, 4)\}$. The equivalence classes for $\sim_{R_{ID}}$, $\{\{0\}, \{1, 3\}, \{2\}, \{4\}\}$, are shown in Figure 3b.*

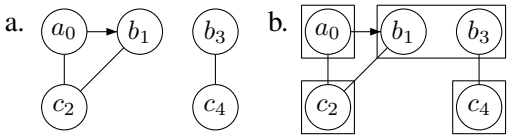

Figure 3: (a) $G_{1,2}$ and (b) $\sim_{R_{ID}}$ in $G_{1,2}$

Given a graph G and a relation $R \subseteq V \times V$, Engelfriet and Vereijken (1997) define $V/R = \{[v]_R | v \in V\}$. This simply 'merges' the nodes of V based on the equivalence relation $\sim_R$. G/R can then be defined as the graph reduced by this merged set of nodes; $\langle V/R, E, A, \ell \rangle$.

The final step is to add precedence arcs to connect $R_{\overline{ID}}$, the unmerged last and first nodes in $G_{1,2}/R_{ID}$. Again, $R_{\overline{ID}}$ is the pairs of last/first nodes on the melody tier that are not identical and the last/first pair on the timing tier, which are never merged.

Definition 2 ***(Concatenation of APGs).*** *The* concatenation $G_1 \circ G_2$ *of graphs G_1 and G_2 in $GR(\Sigma)$ is:*

$$G_1 \circ G_2 = \langle V_{1,2}/R_{ID}, E_{1,2}, A_{1,2} \cup R_{\overline{ID}}, \ell_{1,2} \rangle$$

Example 3 *The concatenation of G_1 and G_2 is given in Figure 4. The node numbered $1, 3$ represents the nodes from Fig. 3 which have been*

merged. Node also the added directed edge $(2, 4)$ *from* $R_{\overline{ID}}$ *in Example 2.*

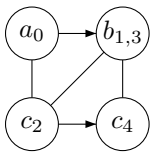

Figure 4: $G_1 \circ G_2$

Technically, the resulting set $V_{1,2}/R_{ID}$ is a set of *sets* of nodes representing the equivalence classes of $\sim_{R_{ID}}$; for example, $V_{1,2}/R_{ID}$ in Example 3 is $\{\{0\}, \{1, 3\}, \{2\}, \{4\}\}$. Represented strictly in this way, successive concatenations will yield sets of sets of sets of nodes, ad infinitum. For example, concatenating a third graph, such as G_3 in Figure 5 below, to $G_1 \circ G_2$ would further merge node $\{1, 3\}$ with node 5 in G_3. Strictly speaking, the resulting node is $\{\{1, 3\}, \{5\}\}$. For clarity, we instead represent each node in this case as the union of the elements of each member of its equivalence class, e.g. $\{1, 3, 5\}$ for the concatenation $(G_1 \circ G_2) \circ G_3$ in Figure 5. This convenient renaming 'flattens out' the nested sets. It does not result in any loss of generality because union is associative. Also, it will be useful later when showing concatenation is associative for the particular class of graphs described in §6.

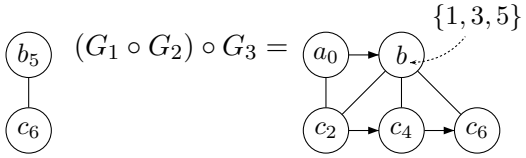

Figure 5: Concatenating a third graph G_3

Importantly, the relations R and R_{ID} do not depend on a binary partition over Σ; they only require that one partition T_t for the timing tier be specified. Thus, while the examples given here focus on two tiers, this operation is defined for graphs representing APRs with multiple melody tiers.

5.2 Properties

This section proves two important properties of concatentation, that G_λ is the identity for $\circ$, and that for any tier in both G_1 and G_2, $G_1 \circ G_2$ contains a string graph corresponding to those tiers.

Theorem 1 *G_λ is the identity element for the $\circ$ operation. That is, for any $G \in GR(\Sigma)$, $G \circ G_\lambda = G_\lambda \circ G = G$.*

Proof: Let $G = \langle V, E, A, \ell \rangle$. We first consider $G_\lambda \circ G$. Recall that the concatenation of two graphs is a modification of their disjoint union. From the properties of the union operation, we know that the disjoint union of G_λ and G is G. Note that $\texttt{first}(G_\lambda, T_i)$ and $\texttt{last}(G_\lambda, T_i)$ are undefined for all $T_i \in T$, because the set of nodes is empty in G_λ. Thus, $R = \emptyset$, and so $R_{ID} = R_{\overline{ID}} = \emptyset$. Because $R_{ID} = \emptyset$, $V/R = V$, because the smallest equivalence relation containing $\emptyset$ is $=$. Thus,

$$G_\lambda \circ G = \langle \quad (V/R) = V, (E \cup \emptyset) = E, \\ (A \cup \emptyset) = A, (\ell \cup \emptyset) = \ell \quad \rangle = G$$

$G \circ G_\lambda = G$ is similarly derived. □

The next lemma shows that concatenation preserves the string graph properties of any tiers in G_1 and G_2. This is important for showing the associativity of concatenation under certain graph classes, as will be discussed in §6.

Lemma 1 *Let U_i and V_i denote the set of all nodes in G_1 and G_2, respectively, with labels in some member T_i of T. If $G_1[U_i]$ and $G_2[V_i]$ are string graphs, or if U_i is empty and $G_2[V_i]$ is a string graph, or if $G_1[U_i]$ is a string graph and V_i is empty, then $(G_1 \circ G_2)[W_i]$ is a string graph, where W_i is the set of all nodes in $G_1 \circ G_2$ whose labels are in T_i. Furthermore, for any T_i, if $v = \texttt{first}(G_1, T_i)$, then $\texttt{first}(G_1 \circ G_2, T_i)$ is the unique node in $G_1 \circ G_2$ which contains v, and likewise for $\texttt{last}(G_2, T_i)$.*

Proof: This follows immediately from the definition of concatenation if $G_1[U_i]$ is a string graph and V_i is empty, because then $\texttt{first}(G_2, T_i)$ will be undefined and no member of U_i will appear in R, and thus all will appear in $G_1 \circ G_2$ unmodified and with no new arcs associated with them. Thus, $G_1[U_i] = (G_1 \circ G_2)[W_i]$ and so both are string graphs. The proof for the case in which U_i is empty and $G_2[V_i]$ is a string graph is very similar.

For the final case, recall that a graph G is a string graph iff its set of arcs A forms a total order $\preceq$ on its nodes V. For the case $G_1[U_i]$ and $G_2[V_i]$ are string graphs and $v_1 = \texttt{last}(G_1, T_i)$ and

$v_2 = \texttt{first}(G_2, T_i)$, then (v_1, v_2) appears in either R_{ID} or $R_{\overline{ID}}$. If the pair is in R_{ID} , v_1 and v_2 are merged into a node $v_{1,2}$ and no new arcs will be introduced to the set A_i of the arcs in $(G_1 \circ G_2)[W_i]$. So for the arc (v_1', v_1) from $G_1[U_i]$ and (v_2, v_2') from $G_2[V_i]$, the corresponding arcs in A_i are $(v_1', v_{1,2})$ and $(v_{1,2}, v_2')$, respectively, which maintains the total orders of both U_i and V_i. If $(v_1, v_2) \in R_{\overline{ID}}$, then (v_1', v_1), (v_1, v_2), and (v_2, v_2') are all in A_i, which also mantains the total order.

That for $v = \texttt{first}(G_1, T_i)$, $\texttt{first}(G_1 \circ G_2, T_i)$ is the unique node which contains v follows directly from the fact that the total order on U_i is maintained. Likewise for $v = \texttt{last}(G_2, T_i)$ and V_i. □

These properties allow us to treat sets of graphs parallel to sets of strings, as the next section shows.

6 APGs derived from concatenation

6.1 Alphabets of graph primitives

As Engelfriet and Vereijken (1997) observe, given a concatenation operation a class of graphs can be seen as an interpretation of a set of strings, where each symbol in the string corresponds to a graph primitive. We now define an APG graph primitive.

Definition 3 *Over an alphabet Σ and tier partition $T = \{T_t, T_m\}$, an* APG graph primitive *is a graph $G \in GR(\Sigma)$ which has the following properties:*

a. V_t is a singleton set $\{v_t\}$
b. $G[V_m]$ is a string graph or is empty
c. All $e \in E$ are of the form $\{v_m, v_t\}$, $v_m \in V_m$

We can then treat a finite set of primitives like an alphabet of symbols:

Definition 4 *An* alphabet of graph primitives *over $GR(\Sigma)$ is a finite set Γ of symbols and a naming function $g : \Gamma \to GR(\Sigma)$.*

An alphabet of APG graph primitives is thus Γ for which for all $\gamma \in \Gamma$, $g(\gamma)$ satisfies Definition 3.

Example 4 *As in the previous examples, consider the alphabet $\Sigma = \{a, b, c\}$ and the tier partition $T = \{T_m = \{a, b\}, T_t = \{c\}\}$. The alphabet of graph primitives $\Gamma = \{a, b, d\}$, where its naming function g is defined as in Figure 6, is an alphabet of APG graph primitives.*

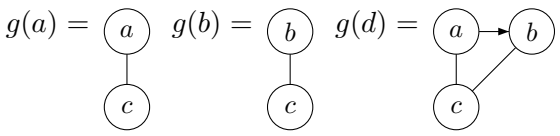

Figure 6: An example Γ and g

The strings in Γ^* thus represent a class of graphs, which we will call $APG(\Gamma)$. We define $APG(\Gamma)$by extending g to strings in Γ^*.

Definition 5 *For an alphabet of graph primitives Γ with naming function g, extend g to strings in Γ^* as follows. For $w \in \Gamma^*$, $g(w) \stackrel{def}{=}$*

- G_λ *if* $w = \lambda$
- $g(u) \circ g(\gamma)$ *if* $w = u\gamma$, $u \in \Gamma^*, \gamma \in \Gamma$

$APG(\Gamma)$ *is thus* $\{g(w) | w \in \Gamma^*\}$.

6.2 Derived properties

We now show that if Γ is an alphabet of APG graph primitives, then $APG(\Gamma)$ has a number of desirable properties. The following assumes Γ is an alphabet of APG graph primitives. First, we prove the following theorem stating that all graphs in $APG(\Gamma)$ follow Axioms 1 through 3 from §4 regarding the general structure of APGs.

Theorem 2 *For any $G \in APG(\Gamma)$, G satisfies Axiom 1 (that $\sim_A$ partitions V into at most two sets V_0 and V_1 such that $G[V_0]$ and $G[V_1]$ are string graphs), Axiom 2 (that the tiers of G correspond to the partition T), and Axiom 3 (that the ends of all undirected edges are between different tiers).*

Proof: That G satisfies Axioms 1 and 2 follows directly from parts (a) and (b) of Definition 3 and the fact that concatenation only adds arcs between nodes whose labels are in the same $T_i \in T$. That $G[V_0]$ and $G[V_1]$ are string graphs follows from parts (a) and (b) of Definition 3 and Lemma 1.

That G follows Axiom 3 follows directly from Part (c) of Definition 3 and the fact that concatenation adds no new undirected edges to E. □

Next, concatenation is associative over $APG(\Gamma)$. The following lemma allows one to prove Theorem 3 (associativity) below.

Lemma 2 *For any $u, v \in \Gamma^*$ denote $g(u), g(v) \in APG(\Gamma)$ with G_u and G_v respectively. Then for any $\gamma \in \Gamma$, $G_u \circ (G_v \circ G_\gamma) = (G_u \circ G_v) \circ G_\gamma$.*

Proof: Let $G = \langle V, E, A, \ell \rangle$ denote $(G_u \circ G_v) \circ G_\gamma$ and $G' = \langle V', E', A', \ell' \rangle$ denote $G_u \circ (G_v \circ G_\gamma)$. That $E = E'$ and $\ell = \ell'$ follow from Definition 2 of concatenation and associativity of union.

To show $V = V'$, there are seven relevent cases to consider. Let V_u, V_v, and V_γ denote the sets of nodes for G_u, G_v, and G_γ, respectively, and let v_u denote a node in V_u, etc. As merging is accomplished through grouping nodes into equivalence classes,all nodes in V or V' thus correspond to either **Cases 1–3** $\{v_u\}$, $\{v_v\}$, $\{v_\gamma\}$, **Cases 4–6** $\{v_u, v_v\}$, $\{v_v, v_\gamma\}$, $\{v_u, v_\gamma\}$, or **Case 7** $\{v_u, v_v, v_\gamma\}$ (recall from §5 we do not distinguish between nodes representing sets and nodes representing sets of sets).

As per the definition of concatenation, $V = ((V_u \cup V_v)/R_{ID\text{-}u,v} \cup V_\gamma)/R_{ID}$, where $R_{ID\text{-}u,v}$ is defined over $V_u \cup V_v$ and R_{ID} over $(V_u \cup V_v)/R_{ID\text{-}u,v}$. Likewise, $V' = (V_u \cup (V_v \cup V_\gamma)/R_{ID\text{-}v,\gamma})/R'_{ID}$.

Cases 1–3. We first establish that when $v \in V$ corresponds to a singleton set that $v \in V'$. Consider the case when $v \in V$ corresponds to $\{v_v\}$, when v_v has not been merged. For V, this is exactly the case in which there is no $(v_u, v_v) \in R_{ID\text{-}u,v}$ nor a $(\{v_v\}, v_\gamma) \in R_{ID}$. We show that this entails that there is neither a $(v_u, \{v_v\}) \in R'_{ID}$ nor a $(v_v, v_\gamma) \in R_{ID\text{-}v,\gamma}$, and so $\{v_v\} \in V'$. There is no $(\{v_v\}, v_\gamma) \in R_{ID}$ either when $\{v_v\}$ is not the last node in $G_u \circ G_v$ for any T_i or there is no v_γ with which it can merge. If $\{v_v\}$ is not the last node in $G_u \circ G_v$ for any T_i, then v_v is not the last node in G_v for any T_i, as by Theorem 2 and Lemma 1 the last node for T_i in $G_u \circ G_v$ must be the unique set which includes the last node for T_i in G_v. If there is no v_γ to merge with $\{v_v\}$, then there is no v_γ to merge with v_v either. Thus, there cannot be a $(v_v, v_\gamma) \in R_{ID\text{-}v,\gamma}$. Similarly, it follows that there is no $(v_u, \{v_v\}) \in R'_{ID}$. If there is no $(v_u, v_v) \in R_{ID\text{-}u,v}$, then either v_v is not the first node in G_v or there is no v_u with which it can merge. Thus, either $\{v_v\}$ is not the first node in $G_v \circ G_\gamma$ (again by Lemma 1) or there is no node v_u to merge with $\{v_v\}$, and so there is no $(v_u, \{v_v\}) \in R'_{ID}$. As there is neither a $(v_u, \{v_v\}) \in R'_{ID}$ nor a $(v_v, v_\gamma) \in R_{ID\text{-}v,\gamma}$, then $\{v_v\} \in V'$. The proofs that $v \in V$ implies $v \in V'$ when v corresponds to $\{v_u\}$ and $\{v_\gamma\}$ are very similar. The proofs that $v' \in V'$ implies $v' \in V$ for all three cases are identical.

The remaining cases deal with merged nodes. **Cases 4–6.** Consider the case in which $v \in V$ is $\{v_u, v_v\}$ corresponding to merged nodes from V_u and V_v. This is the case in which $(v_u, v_v) \in R_{ID\text{-}u,v}$ but there is no $(\{v_u, v_v\}, v_\gamma) \in R_{ID}$ for any v_γ. As before, if $\{v_u, v_v\}$ cannot be merged with some v_γ, then there is no v_γ to merge with v_v. Furthermore, if $(v_u, v_v) \in R_{ID\text{-}u,v}$, then v_u is the last node in G_u and v_v is the first node in G_v for some T_i. By Lemma 1, then $\{v_v\}$ is the first node in $G_v \circ G_\gamma$ for T_i, and so $(v_u, \{v_v\}) \in R'_{ID}$. Thus, $\{v_u, v_v\} \in V'$. The proof that $v \in V$ implies $v \in V'$ when $v = \{v_v, v_\gamma\}$ is very similar, as it is for $v = \{v_u, v_\gamma\}$. The latter is a special case in which V_v has no nodes for some T_i, but v_u and v_γ are compatible to merge. **Case 7.** For $v = \{v_u, v_v, v_\gamma\}$, both $(v_u, v_v) \in R_{ID\text{-}u,v}$ and $(\{v_u, v_v\}, v_\gamma) \in R_{ID}$. If $(\{v_u, v_v\}, v_\gamma) \in R_{ID}$, then through Lemma 1 $(v_v, v_\gamma) \in R_{ID\text{-}v,\gamma}$ and then $(v_u, \{v_v, v_\gamma\}) \in R'_{ID}$, so $\{v_u, v_v, v_\gamma\} \in V'$. The proofs that $v' \in V'$ implies $v' \in V$ for these cases are identical.

That $A = A'$ is very similar to the proof for $V = V'$. Let A_i denotes the set of arcs in $g(\gamma_i)$. Then $A = (A_u \cup A_v \cup R_{\overline{ID-u,v}}) \cup A_\gamma \cup R_{\overline{ID}}$, where $R_{\overline{ID-u,v}}$ is defined over $(V_u \cup V_v)/R_{ID\text{-}u,v}$ and $R_{\overline{ID}}$ is defined over $((V_u \cup V_v)/R_{ID\text{-}u,v} \cup V_\gamma)/R_{ID}$, and $V_u \cup V_v$ $A' = (A_u \cup (A_v \cup A_\gamma \cup R_{\overline{ID-v,\gamma}}) \cup R'_{\overline{ID}})$, where $R_{\overline{ID-v,\gamma}}$ and $R'_{\overline{ID}}$ are defined parallel to $R_{\overline{ID-u,v}}$ and $R_{\overline{ID}}$. As union is associative, it is sufficient to show that every pair $(v_u, v_v) \in R_{\overline{ID-u,v}}$ has a corresponding pair $(v_u, \{v_v\})$ or $(v_u, \{v_v, v_\gamma\})$ in $R'_{\overline{ID}}$ and vice versa, and that every pair $(\{v_v\}, v_\gamma)$ or $(\{v_u, v_v\}, v_\gamma)$ in $R_{\overline{ID}}$ has a corresponding pair $(v_v, v_\gamma) \in R_{\overline{ID-v,\gamma}}$, and vice versa. Both of these follow from the fact that $V = V'$ and Lemma 1 in the same way as merging nodes above. □

Next it is shown that graph concatenation is associative over arbitrary graphs in $APG(\Gamma)$ with the same kind of inductive argument which establishes concatenation is associative over strings.

Theorem 3 *The $\circ$ operation is associative over graphs in $APG(\Gamma)$. For any $u, v, w \in \Gamma^*$ denote $g(u), g(v), g(w) \in APG(\Gamma)$ with G_u, G_v, G_w re-*

spectively. Then $G_u \circ (G_v \circ G_w) = (G_u \circ G_v) \circ G_w$.

Proof: The proof is by induction on the size of w. For the base case, when $w = \lambda$, $G_w = G_\lambda$. Then $G_u \circ (G_v \circ G_w) = G_u \circ (G_v \circ G_\lambda)$, which equals $G_u \circ G_v$ by Theorem 1. It follows, again by Theorem 1, that $(G_u \circ G_v) \circ G_\lambda$. Hence the base case is proved.

Next we assume the inductive hypothesis that associativity holds for strings of length n and we consider any $w \in \Gamma^*$ of length $n+1$. Clearly there exists $x \in \Gamma^*$ of length n and $\gamma \in \Gamma$ so that $w = x\gamma$. Then $G_u \circ (G_v \circ G_w) = G_u \circ (G_v \circ (G_x \circ G_\gamma))$. By Lemma 2, this equals $G_u \circ ((G_v \circ G_x) \circ G_\gamma)$, which again by Lemma 2, equals $G_u \circ (G_v \circ G_x)) \circ G_\gamma$. Then, by the induction hypothesis, we have $((G_u \circ G_v) \circ G_x) \circ G_\gamma$, which again by the induction hypothesis, yields $(G_u \circ G_v) \circ (G_x \circ G_\gamma)$. This is of course is $(G_u \circ G_v) \circ G_w$. □

The next theorem states that any $G \in APG(\Gamma)$ follows the NCC.

Theorem 4 *For any* $G \in APG(\Gamma)$*,* G *satisfies the NCC (Axiom 4).*

Proof: The proof is by recursion on the length of $w \in \Gamma^*$. G_λ trivially satisfies the NCC because it has no nodes. For $g(\gamma)$ for any $\gamma \in \Gamma$, Definition 4 states that there is only one node v_t in V_t and this node must be one of the endpoints for each edge in E. Thus for any two edges $\{x, y\}$ and $\{x', y'\}$ in $g(\gamma)$ where $x \preccurlyeq x'$, it must be the case that $y = y' = v_t$, because directed edges only occur between nodes in tier V_m. Thus, any $g(\gamma)$ satisfies the NCC.

Next we assume it holds for $w \in \Gamma^*$ of length n and consider any $w \in \Gamma^*, \gamma \in \Gamma$. Then $g(w\gamma)$ satisfies the NCC because the graph concatenation operation does not add any undirected edges and because, by Lemma 1 concatenation preserves the order of each tier in $g(w)$ and $g(\gamma)$. □

The final theorem states that any $G \in APG(\Gamma)$ follows the OCP if the graph primitives do.

Theorem 5 *If* $g(\gamma)$ *for all* $\gamma \in \Gamma$ *satisfy the OCP (Axiom 5), then for any* $G \in APG(\Gamma)$*,* G *satisfies Axiom 5.*

Proof: The proof is again by recursion on the length of $w \in \Gamma^*$. The OCP is trivially satisfied for G_λ since it contains no nodes or arcs. The case when $|w| = 1$ is given as the condition of the theorem.

Assume that every $w \in \Gamma^*$ of length n satisfies the OCP. Now consider $G = g(w\gamma)$ with w of length n and $\gamma \in \Gamma$. To see that $G_u \circ G_\gamma$ satisfies the OCP, recall from Definition 2 of graph concatenation that the set of arcs for $G_1 \circ G_2$ is equal to $A_{1,2} \cup R_{\overline{ID}}$; i.e., the union of A_1 and A_2 and $R_{\overline{ID}}$. By definition $R_{\overline{ID}}$ only includes pairs of nodes (x, y) s.t. $\ell(x) \neq \ell(y)$, so if G_1 satisfies the OCP and G_2 satisfies the OCP $R_{\overline{ID}}$ will not add any arcs on V_m which violate the OCP (recall that the OCP only holds for tier V_m), and so $G_1 \circ G_2$ will also satisfy the OCP. □

Thus, the merging part of the concatenation preserves the OCP. One may wonder why the OCP is built in to the concatenation operation this way, instead of using string-like concatenation and then invoking a constraint that merges adjacent, like nodes in the resulting graph. Such a method, though, cannot capture violations of the OCP—all would be merged. The next section shows that the concatenation operation defined here can capture violations by concatenating OCP-violating graph primitives.

This section has thus proved the important properties of $APG(\Gamma)$. We now show how such an $APG(\Gamma)$ can be used to model autosegmental phenomena in natural language phonology.

7 Analysis of natural language phenomena

In this section we examine the extent to which the analysis presented here accounts for common and uncommon phenomena in phonological theory. The first two subsections examine spreading and contour tones, respectively, and demonstrate how both phenomena can be effectively represented with a $APG(\Gamma)$ for some Γ. It is also shown that the empirical generalization that there are only finitely many contour tones present in any given language is an automatic consequence of the finite alphabet Γ and the concatenation operation.

The third subsection addresses the few cases where OCP violations may be necessary to properly describe the language. It is sketched out how these cases could be accounted for by using special graph primitives or a second concatenation operation. Similarly, the fourth subsection addresses

underpecification and floating tones. We conclude that these concepts can be represented in this approach. The caveat is that it is also observed as a consequence that gapped structures are also permitted. Again, we note that such gapped structures are also permitted with axioms given in §4 approaches above, and we discuss how a different concatenation operation may address this.

7.1 Spreading

The 'merging' of nodes on the melody tier models autosegmental spreading, in which one melody unit is associated to more than one timing tier unit. A classic example is Mende (Leben, 1973). Mende nouns separate into tone categories, three of which are shown in Table 1. The first rows show words whose syllables are all high-toned, the second rows show words whose syllables are all low-toned, and the third rows show words whose syllables start high and end low. In the following [á] transcribes a high tone, [à] a low tone, [â] a falling tone.

Monosyllables		Disyllables	
kɔ́	'war'	pɛ́lɛ́	'house'
kpà	'debt'	bɛ̀lɛ̀	'pants'
mbû	'owl'	ngílà	'dog'

Trisyllables	
háwámá	'waist'
kpàkàlì	'three-legged chair'
félàmà	'junction'

Table 1: Mende word tone

An autosegmental analysis for this pattern is that a set number of melodies spread left-to-right over the tone-bearing units (TBUs; we assume that for Mende the TBU is the syllable, σ) of a word, as in Table 2. For example, the falling tone words [mbû] 'owl' and [félàmà] 'junction' have an HL melody. In this case, the H associates to the first syllable of the word, and the L associates to all remaining syllables.

H kɔ́'war'	H háwámá 'waist'
HL mbû 'owl'	HL félàmà 'junction'

Table 2: APRs for four Mende words

The APRs in Table 2 can be generated with the alphabet of APG graph primitives Γ given in Figure 7. The alphabet is $\Sigma = \{\mathrm{H}, \mathrm{L}, \sigma\}$ and the tier partition $T = \{T_t, T_m\}$ where $T_t = \{\sigma\}$ and $T_m = \{\mathrm{H}, \mathrm{L}\}$. Note that for these APGs, we abstract away from consonants and vowels and focus on the TBU, σ.

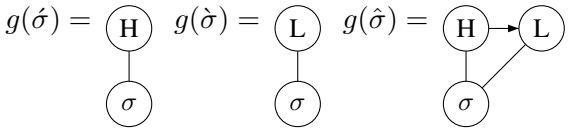

Figure 7: Γ and g for Mende

The APGs corresponding to the trisyllabic forms are thus $g(\acute{\sigma}\acute{\sigma}\acute{\sigma})$ and $g(\acute{\sigma}\grave{\sigma}\grave{\sigma})$, as in Figure 8.

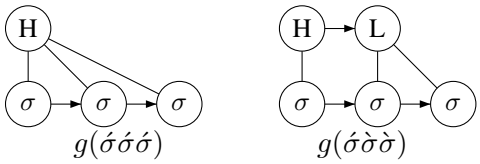

Figure 8: APGs for Mende APRs in Table 2

These spreading effects are achieved by, for example in $g(\acute{\sigma}\acute{\sigma}\acute{\sigma})$, the like H nodes from each $g(\acute{\sigma})$ merging during concatenation, resulting in a single H associated to multiple σ nodes (which are not merged, because $\sigma \in T_t$). Note that given Σ, T, Γ and g, we are able to generate APGs directly from the linear string of toned syllables.

7.2 Contours

Concatenation allows for unbounded spreading, as a single node on the melody tier may 'merge' any number of times. In contrast, concatenation does *not* allow for unbounded contours, as timing tier nodes do not merge. Figure 9 shows how concatenation obtains APGs corresponding to the APRs for the Mende words [mbû] 'owl' and [nyàhâ] 'woman'.

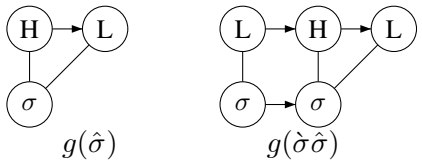

Figure 9: APGs for Mende contours

Importantly, any set of graphs is going to have a bound on the number of melody units a contour can have, which follows directly from the fact that Γ is finite, that each element of Γ has exactly one node on V_t, and so concatenation never creates new contours. Thus, for the example Γ we have been using for Mende, the graph in Figure 10 is *not* in $APG(\Gamma)$.

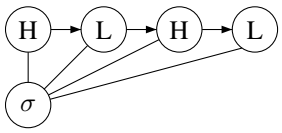

Figure 10: A graph not in $APG(\Gamma)$

While this is a natural property of graphs in $APG(\Gamma)$, the axiomatic approach to defining APRs requires a further axiom stating that for any language, the number of contours must be bound by some n. To our knowledge, the only explicit formalizations of such a constraint are by Jardine (2014) and Yli-Jyrä (2013) (the latter requiring that $n = 2$).

7.3 Violations of the OCP

As discussed in Odden (1986) and Meyers (1997), the OCP may not be an absolute universal. For example, Odden lists the contrasting APRs in Figure 11 for two nouns in Kishambaa (Odden, 1986, Fig. 13):

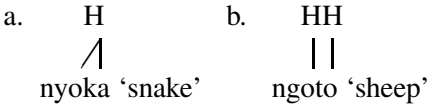

Figure 11: OCP violating forms in Kishambaa

This is partially motivated by the different surface pronunciation of the two forms: the first, Figure 11 (a) ‘snake’ is pronounced with two level H tones, nyóká, and 11 (b) ‘sheep’ is pronounced with a H followed by a downstepped H; ngó!tó.

The corresponding graphs for these APRs, assuming the mora as the TBU, are given in Figure 12. Figure 12 (a) corresponds to Figure 11 (a), and Figure 12 (b) to Figure 11 (b).

Given an alphabet of graph primitives obeying the OCP, as the Γ for Mende in Figure 7, Figure 12 (a) is in $APG(\Gamma)$, but Figure 12 (b) is not, because it does not obey the OCP. Thus, Kishambaa is not describable with such a graph set $APG(\Gamma)$.

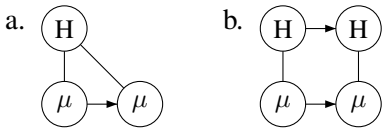

Figure 12: APGs for Kishambaa forms

There are at least two solutions to admitting graphs like in Kishambaa. One is to introduce OCP-violating graph primitives, as in Figure 13.

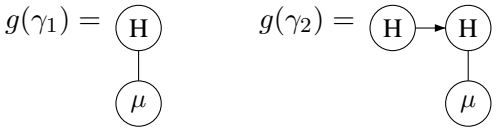

Figure 13: A Γ for Kishambaa

Given this alphabet of graph primitives, the spreading Kishambaa graph in Figure 12 (a) is $g(\gamma_1\gamma_1)$, and the OCP-violating (b) is $g(\gamma_1\gamma_2)$. The graph primitives follow the linear pronunciation of the morae; $g(\gamma_1\gamma_1)$ represents a sequence HH of two high-toned morae, and $g(\gamma_1\gamma_2)$ a sequence H!H of a high followed by a downstepped high.

Another option is to define a second concatenation operation, in which there is no merging and directed edges are drawn between all last/first pairs. Spreading Kishambaa graph in Figure 12 (a) would be concatenated by the operation defined in this paper, and the OCP-violating Figure 12 (b) would be concatenated by this second no-merging operation. We shall leave it up to future work to compare the theoretical and empirical benefits of these approaches to OCP violations.

7.4 Underspecification and floating tones

Some graph primitives in Γ may not have any nodes in V_m; these represent *underspecified* timing units.

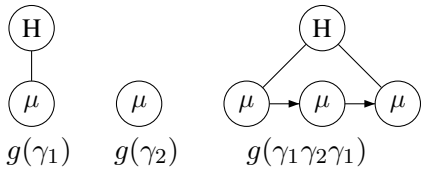

Figure 14: APGs with underspecification

However, such underspecified graph primitives

can give rise to 'gapped structures' via concatenation, as in $g(\gamma_1\gamma_2\gamma_1)$ in Figure 14. This can be seen as an unwelcome consequence as some researchers have argued against gapped structures (Archangeli and Pulleyblank, 1994). One solution could be to use a second concatenation operation which does not merge nodes, instead only drawing directed edges between the end nodes on each tier. This appears identical to the operation proposed in §7.3 for dealing with OCP violations. Again, studying additional concatenation operations will be left for future work.

Finally, graph primitives with more melody tier nodes than timing tier nodes can be used to generate floating tones, as in Figure 15.

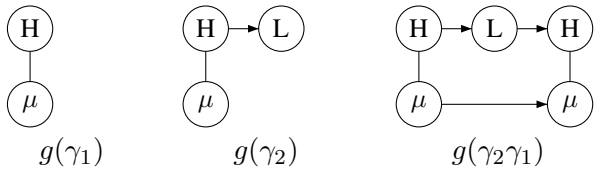

Figure 15: Generating APGs with floating tones

8 Dicussion

The examples in the previous section show several advantages to considering APRs through concatenation. One, as seen in Mende, simple cases allow direct translation of strings into APRs. Second, concatenation allows for unbounded spreading, as a single node on the melody tier may 'merge' any number of times. However, concatenation does *not* allow for unbounded contours, as timing tier nodes do not merge in this way. Thus, the number of contours is bounded by the number of graph primitives. This reflects the fact that languages exhibit unbounded spreading, but no language (to our knowledge) has an unbounded number of contour segments.

There are several avenues for future work. It was already mentioned that the set of valid autosegmental representations may be expanded by allowing a second concatenation operation. Also, while we have shown that every element of $APG(\Gamma)$ obeys the axioms in §4, it remains to be shown that for every graph which obeys those axioms, there is a finite alphabet which generates it.

Future work can also study the nature of transformations from underlying APGs with one alphabet to surface APGs with another (for instance it is known surface APGs can admit more contours than underlying APGs through association rules).

Another line of development concerns extending the analysis to feature geometry (Clements and Hume, 1995; Sagey, 1986), in which association lines also link featural autosegments and 'organizational' nodes, such as PLACE. Deriving a set of such operations would require more complex primitives and additional marking on the tier partition T, to denote timing tier nodes, organizational nodes, and melody nodes. The concatenation operation would then need to be revised to be sensitive to this marking. A more serious challenge would be adopting a concatenation-based framework for autosegmental morphology, which as mentioned in §2, disposes of the requirement that autosegments of a particular type must appear on a particular tier.

9 Conclusion

In this paper we addressed the question of what is the set of valid autosegmental representations looks like. In contrast to previous research, which explored this question axiomatically, we showed that the autosegmental representations can be generated recursively and constructively from a finite set of graph primitives, a concatenation operation, and an identity element for concatenation, much in the same way that strings can be so generated. Hence, the theory of free monoids may be fruitfully applied to APRs.

The advantages we wish to highlight are as follows. First, we proved that provided the finite set of primitives obey the NCC and the OCP, the autosegmental representations will as well. Second, we showed it also follows naturally from the nature of the alphabet and concatenation that new contour tones cannot be generated ad infinitum. Finally, this method makes clear the stringlike nature of autosegmental representations, and that their properties can be viewed as a consequence of this nature.

Acknowledgments

The authors would like to thank three reviewers for their insightful comments and suggestions. The first author acknowledges support from a University of Delaware Graduate Research Fellowship.

References

Diana Archangeli and Douglas Pulleyblank. 1994. *Grounded Phonology*. Cambridge: MIT Press.

Steven Bird and E. Klein. 1990. Phonological events. *Journal of Linguistics*, 26:33–56.

G. N. Clements and Elizabeth V. Hume. 1995. The internal organization of speech sounds. In John Goldsmith, editor, *The handbook of phonological theory*, pages 245–306. Oxford: Blackwell.

G. N. Clements. 1976. *Vowel Harmony in Nonlinear Generative Phonology: An Autosegmental Model*. Bloomington: Indiana University Linguistics Club Publications.

John Coleman and John Local. 1991. The "No Crossing Constraint" in autosegmental phonology. *Linguistics and Philosophy*, 14:295–338.

Joost Engelfriet and Hendrik Jan Hoogeboom. 2001. MSO definable string transductions and two-way finite-state transducers. *ACM Transations on Computational Logic*, 2:216–254, April.

Joost Engelfriet and Jan Joris Vereijken. 1997. Context-free graph grammars and concatenation of graphs. *Acta Informatica*, 34:773–803.

John Goldsmith. 1976. *Autosegmental Phonology*. Ph.D. thesis, Massachussets Institute of Technology.

John Goldsmith. 1979. *Autosegmental Phonology*. Gardland Press.

Michael Hammond. 1988. On deriving the Well-Formedness Condition. *Linguistic Inquiry*, 19(2):319–325.

Adam Jardine. 2014. Logic and the generative power of Autosegmental Phonology. In John Kingston, Claire Moore-Cantwell, Joe Pater, and Robert Staubs, editors, *Supplemental proceedings of the 2013 Meeting on Phonology (UMass Amherst)*, Proceedings of the Annual Meetings on Phonology. LSA.

András Kornai. 1995. *Formal Phonology*. Garland Publication.

W. R. Leben. 1973. *Suprasegmental phonology*. Ph.D. thesis, Massachussets Institute of Technology.

John J. McCarthy. 1979. *Formal Problems in Semitic Phonology and Morphology*. Ph.D. thesis, Massachusetts Institute of Technology.

John J. McCarthy. 1985. *Formal Problems in Semitic Phonology and Morphology*. New York: Garland.

John J. McCarthy. 1986. OCP effects: gemination and antigemination. *Linguistic Inquiry*, 17:207–263.

Scott Meyers. 1997. OCP effects in Optimality Theory. *Natural Language & Linguistic Theory*, 15(4):847–892.

David Odden. 1986. On the role of the Obligatory Contour Principle in phonological theory. *Language*, 62(2):353–383.

Douglas Pulleyblank. 1986. *Tone in Lexical Phonology*. Dordrecht: D. Reidel.

Elizabeth Sagey. 1986. *The Representation of Features and Relations in Non-Linear Phonology*. Ph.D. thesis, Massachusetts Institute of Technology.

Moira Yip. 2002. *Tone*. Cambridge University Press.

Anssi Yli-Jyrä. 2013. On finite-state tonology with autosegmental representations. In *Proceedings of the 11th International Conference on Finite State Methods and Natural Language Processing*, pages 90–98. Association for Computational Linguistics.