

# Logic and the Generative Power of Autosegmental Phonology\*

Adam Jardine  
*University of Delaware*

## 1 Introduction

The current study examines the generative power of Autosegmental Phonology (Goldsmith, 1976, 1979, 1990) in the framework of Formal Language Theory, with which we can study the computational complexity of phonological phenomena and formalisms independent of specific theoretical frameworks in phonology. A methodology for a model-theoretic study of autosegmental phonology with monadic second-order logic is introduced. Monadic second order logic is chosen because it provides a mathematically rigorous way of studying autosegmental formalisms, and its complexity is well understood. The preliminary conclusion is that autosegmental diagrams which conform to the well-formedness constraints defined here likely describe at most *regular* sets of strings.

The structure of this paper is as follows. Section 2 motivates the current work and reviews related research. Section 3 details the proposal, and Section 4 concludes, outlining directions for further work.

## 2 Background

**2.1 Formal studies of phonology** This paper takes the view that it is important to have a formal understanding of phonological models in order to have precise knowledge of what they predict. This is not a new view; Potts & Pullum (2002) and de Lacy (2011), for example, propose explicit descriptions of the content of OT constraints, and Graf (2010a,b) compares the descriptive power of Sound Pattern of English (henceforth SPE; Chomsky & Halle, 1968)–style rewrite rule formalisms and those of Government Phonology (Kaye et al., 1990).

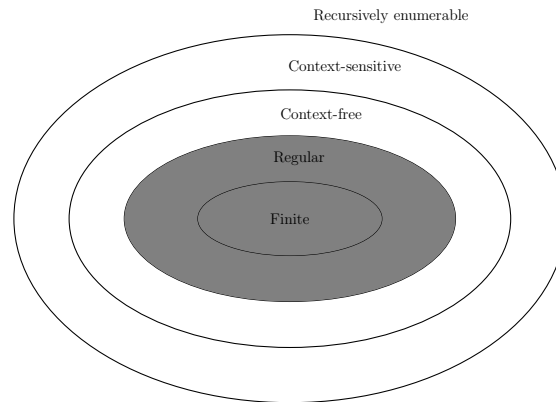
It is important, of course, to have some way of measuring the ‘descriptive power’ of phonological formalisms. To this end, this paper utilizes a Formal Language Theoretic (FLT) perspective, which classifies patterns of symbols according to their complexity. FLT is apposite for studying the complexity of linguistic formalisms because a) it provides well-studied, mathematically rigorous method of studying complexity and b) its application to natural language patterns has already yielded results, as to be detailed below. The core of FLT is a nested complexity hierarchy of classes of patterns called the *Chomsky hierarchy*, after its original development by Chomsky (1956; for a more recent discussion for the hierarchy its application to natural language, see Partee et al., 1993). This hierarchy is shown in Figure 1.

The regions represented by the concentric circles in Figure 1 are of increasing complexity as one moves out from the center. For example, the *regular* region, shaded in gray in Figure 1, is strictly less complex than the *context-free* region, represented by the next largest circle. To give a concrete examples of what this means, the pattern in (1) is a regular pattern (1a) of ‘a’s and ‘b’s, whereas (1b) is context-free:

- |     |    |           |    |           |
|-----|----|-----------|----|-----------|
| (1) | a. | $\lambda$ | b. | $\lambda$ |
|     |    | ab        |    | ab        |
|     |    | abab      |    | aabb      |
|     |    | ababab    |    | aaabbb    |
|     |    | abababab  |    | aaaabbbb  |
|     |    | ...       |    | ...       |

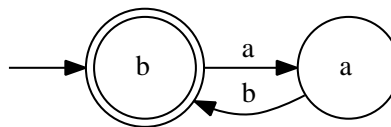
---

\* The author would first like to thank Jeffrey Heinz and James Rogers for their ideas and advice on this project. Many thanks also go to an interested and thought-provoking audience at the Phonology 2013 conference. All errors are my own.



**Figure 1:** The Chomsky Hierarchy

The pattern in (1a) can be described as any number (including zero— $\lambda$  above represents a string of length zero) of repetitions of the sequence ‘ab’; (1b) is any sequence of ‘a’s followed by a sequence of ‘b’s of the same length (this is often referred to as  $a^n b^n$ ). The reason (1a) is considered to be less complex than (1b) is that a grammar formalism powerful enough to generate (1b) will also be able to generate (1a), but not vice versa. ‘Grammar formalism’ can mean any number of methods for describing patterns, and part of FLT deals with how grammar formalisms relate to each other. One commonly referred-to formalism is the *finite-state machine*, an abstract machine that can check whether or not a string (or tuple of strings) is in a pattern. A *finite-state acceptor* (henceforth FSA), which checks strings one at a time, for (1a) is given in Figure 2.



**Figure 2:** FSA for (1a)

FSAs, like the one in Figure 2, check strings by reading them left-to-right, one symbol at a time, with the symbols triggering transitions from state to state. When the string ends, the FSA accepts the string if it is in a final state (marked with double circles), otherwise it rejects the string. Taking the string ‘abab’ through Figure 2, the machine starts on the state labelled *b* (the unlabelled arrow marks it as a start state), takes the ‘a’ transition on the first ‘a’ of the string, then the ‘b’ transition back to ‘b’, then repeats these two steps, then ends on the *b* state. Because *b* is a final state, the FSA accepts the string—and it will do so for all the strings in (1a) and *only* the strings in (1a).

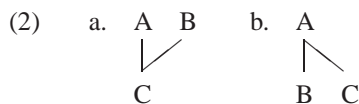
A key property of FSAs, and finite-state machines in general, is that they have a finite memory, represented by their states. For example, the FSA in Figure 2 can only ‘remember’ whether or not it has just seen an ‘a’ or a ‘b’. This property restricts the kinds of patterns it can describe—no FSA can describe the pattern in (1b), because it would have to remember exactly how many ‘a’s it has seen in order to check that there are the same number of ‘b’s. Because there can be *any* number of ‘a’s, there is no bound on the memory required to check a string in pattern (1), and thus it cannot be done in a finite number of states. Thus, an FSA can describe the regular pattern in (1a), but not the context-free one in (1b). In fact, FSAs describe exactly the regular patterns, and nothing more complex.

An important observation regarding natural language phonology from the FLT perspective, originally attributable to Johnson (1972) and Kaplan & Kay (1994), is that its computational complexity is likely to be at most regular. This can be thought of intuitively by taking the regular repeating ‘ab’ pattern in (1a) and replacing the ‘a’s with ‘C’s and the ‘b’s with ‘V’s—we suddenly get a familiar phonological pattern.

In contrast, most phonologists would likely agree that nothing like the context-free  $a^n b^n$  pattern in (1b) is attested in natural language phonology (although it is in syntax, in the form of nested embedding sentences like “The dog the cat the rat bit chased ran away”). More concretely, Johnson (1972) and Kaplan & Kay (1994) proved that, with the restriction that they cannot apply to their own output, SPE-style rewrite rules cannot have a pattern like (1b) in either their output or their input. Since SPE is generally considered powerful enough to capture any phonological alternation, it is reasonable to presume that phonology is regular. More recent research has pushed this hypothesis further, arguing that phonology is *sub*-regular (Heinz, 2009, 2010; Graf, 2010b; Heinz & Lai, 2013; Rogers et al., 2013).

Given these hypotheses, we likewise expect that Autosegmental Phonology (AP) should not be able to produce non-regular strings. The current proposal explores the complexity of AP. This is not the first work to examine the complexity of AP; as surveyed in the next section, this territory has been travelled before. However, like Kaplan & Kay (1994), Heinz (2009), and others, the work exploring the computational complexity of AP uses an automata-theoretic perspective, which equates regularity with the ability to be described with finite-state machines. For the reasons to be outlined below, the current paper departs from this work and, like Potts & Pullum (2002) and Graf (2010a,b), utilizes a model-theoretic perspective.

**2.2 Computational studies of autosegmental phonology** Autosegmental Phonology (henceforth AP; Goldsmith, 1976, 1979, 1990) is a representational theory of phonology in which phonological units on multiple autonomous tiers may be associated to each other in a many-to-one fashion. The usual graphical convention is to represent tiers as vertically separated strings of symbols, with the association relation shown as lines drawn between symbols on different tiers, as in (2) below.



The late 1980s and early 1990s AP saw a flurry of work examining AP from a formal perspective. Some work focused on the semantic interpretation of association relations (Sagey, 1986, 1988; Bird & Klein, 1990), while others discussed the formal groundwork for computational implementation of AP (Kay, 1987; Kornai, 1991; Wiebe, 1992; Bird & Ellison, 1994; Kornai, 1995). This latter work is highly relevant to the current paper, as while the computational complexity of AP was not its primary focus, the topic featured heavily in the discussion, with the aforementioned researchers disagreeing as to whether or not AP could be managed in a finite-state framework. It is thus important to first summarize this work.

The first to attempt the problem was Kay (1987), who devised a scheme for representing the nonconcatenative morphology of Arabic with a four-tape finite-state *transducer*, which checks not single strings but tuples of strings. In brief, his proposal assigns each autosegmental tier—the consonantal root morpheme, the vowel inflection morpheme, and the CV tier to which the former two are associated—its own ‘tape’, a string of symbols readable by a finite-state machine. In this way, the machine is able to check these three strings and a string on the fourth ‘output’ tape representing the linearized surface string. While Kay’s proposal was restricted to Arabic morphology, Wiebe (1992) generalized the multi-tape method to AP proper, but noted that these multi-tape machines are formally more powerful than standard finite-state transducers, which by definition have only two tapes. Wiebe argued that this power is necessary, and as such autosegmental phonology is necessarily not finite-state and non-regular.

In contrast, Kornai (1991, 1995) and Bird & Ellison (1994) each developed formal implementations of AP they claimed are finite-state. Kornai (1991, 1995) developed a linearized representation of autosegmental ‘bistrings’ (pairs of associated tiers and their association), representing movements through a two-tape ‘biautomaton’, a two-tape finite state machine which advances through the tapes based on associations between symbols on the tapes. As these biautomata are not finite-state transducers in the strict sense, Kornai (1991, 1995) tackles the question of their generative capacity by giving an algebraic definition of regularity for sets of bistrings. He argues that by this metric, his implementation is indeed regular, and is thus in some sense ‘finite-state’. Bird & Ellison (1994) question this result, arguing that Kornai’s representations cannot handle some autosegmental diagrams, and further state that the ‘the association mechanism for coded autosegmental representations is not finite state’ (p. 83). They instead argue for their declarative model of phonology, which represents autosegmental information with special finite-state acceptors whose states are

‘synchronized’. However, again, it is unclear how these acceptors compare to standard finite-state acceptors, and as such it is hard to evaluate their claims regarding complexity.

It is now clear that an important reason for the differing results in the above work regarding the complexity of AP is that all use non-standard variations of the finite-state machine in order to deal with the non-linearity. As such, it is hard to directly compare the computational complexity to results measuring the computational complexity of phonology with standard finite-state acceptors and transducers, which deal with linear strings and pairs of strings, respectively. The current proposal deals these problems in two ways. First, it defines AP in a logical framework independent of automata; this avoids the problems implementing non-linear representations in finite-state machines. Second, with logic, it is able to define a linear *yield* of an autosegmental diagram, which is simply the surface string resulting from the diagram. Thus, the yields of a set of autosegmental diagrams make up a pattern like those in (1), and thus are straightforwardly comparable to other patterns in the usual formal language-theoretic sense. Section 3 details how this is done, but first a brief introduction to Monadic Second Order Logic is necessary; this is the concern of the following subsection.

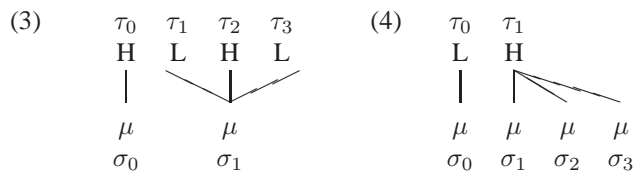
**2.3 Monadic Second Order Logic** Monadic Second Order Logic (MSO) is a boolean logic utilizing the usual operators ( $\neg, \wedge, \vee$ , from which  $\rightarrow$  and  $\leftrightarrow$  can also be derived) and quantification ( $\forall, \exists$ ) of both variables ( $x$ ) and sets of variables ( $X$ ). When applied to strings, these variables refer to positions in the string, and monadic predicates can evaluate whether or not a position belongs to a set of variables ( $X(x)$ ) or has a particular symbol value ( $a(x)$ , where  $a$  is a symbol in the alphabet). A classic FLT result is that sets of strings definable with MSO predicates and a precedence relation between positions are exactly regular (Büchi, 1960). Thus, MSO is an apt methodology for studying phonological formalisms; not only is it mathematically explicit and rigorous, but its complexity is well-known.

### 3 Autosegmental Phonology in Monadic Second-Order Logic

The following defines AP in MSO in order to measure its computational complexity. It is important to note that this is not the first model-theoretic treatment of AP; Bird & Klein (1990) and Kornai (1991, 1995) view AP from a model-theoretic perspective. However, in addition to some implementational differences discussed below, this proposal is unique in that it defines a yield of autosegmental diagrams in order to discuss their generative power. This section is structured as follows. Section 3.1 describes how to build models of AP diagrams to which MSO predicates can refer; Section 3.2 builds a set of such predicates; Section 3.3 uses these predicates to define axioms restricting the set of well-formed AP models; Section 3.4 defines the string yields of AP models; and Section 3.5 concludes with a conjecture that these axioms restrict us to sets of AP diagrams whose yields are regular.

**3.1 Autosegmental diagrams as word models** In autosegmental diagrams, in the most general sense, there can be a one-to-many relationship between autosegments on one tier and on autosegments on another. As examples, we consider autosegmental diagrams in which segmental tone-bearing units (TBUs) (whose positions are marked with  $\sigma$ s) are associated with some tones on a tonal tier ( $\tau$ s). For the purposes of simplicity, for now the segmental alphabet  $\Sigma$  will only be  $\{\mu\}$ , as the mora is a phonological structural unit commonly analyzed as a TBU. The tonal alphabet  $T$  will be  $\{H, L\}$ . The following examples are artificial illustrations of how the formalism works; however, they are inspired by real-world examples such as the classic case of tonal melodies associating to vowels in Mende (Leben, 1973, 1978).

In (3), multiple positions on the tonal tier (specifically,  $\tau_1, \tau_2$ , and  $\tau_3$ ) are associated to a single position on the segmental tier ( $\sigma_1$ ). In (4), shows another possibility, with multiple positions on the segmental tier ( $\sigma_1, \sigma_2$ , and  $\sigma_3$ ) associated to one position on the tonal tier ( $\tau_1$ ).



We can treat such diagrams as the association between two word models  $\mathcal{W}$ , each a tuple with a set

of positions  $W$  (canonically an initial segment of the natural numbers  $\mathbb{N}$ ), a linear less-than-or-equal-to precedence relation  $\triangleleft^*$  (i.e., it is both reflexive and transitive), an immediate association relation  $\triangle$  (which is neither reflexive nor transitive), and a set of predicates defining the symbol assigned to each position.

$$(5) \quad \begin{aligned} \mathcal{W}_T &= \langle W_T, \triangleleft^*, P_T \rangle \\ \mathcal{W}_\Sigma &= \langle W_\Sigma, \triangleleft^*, P_\Sigma \rangle \\ \mathcal{A} &= \langle \mathcal{W}_T, \mathcal{W}_\Sigma, \triangle \rangle \end{aligned}$$

For example, (3) above corresponds to the following model:

$$(6) \quad \begin{aligned} \mathcal{W}_T &= \langle \{0, 1, 2, 3\}_{W_T}, \{(0 \triangleleft^* 0), (0 \triangleleft^* 1), (0 \triangleleft^* 2), \dots, (3 \triangleleft^* 3)\}_{\triangleleft^*}, \{0, 2\}_H, \{1, 3\}_L \rangle \\ \mathcal{W}_\Sigma &= \langle \{0, 1\}_{W_\Sigma}, \{(0 \triangleleft^* 0), (0 \triangleleft^* 1), (1 \triangleleft^* 1)\}_{\triangleleft^*}, \{0, 1\}_\mu \rangle \\ \mathcal{A} &= \langle \mathcal{W}_T, \mathcal{W}_\Sigma, \{(0 \triangle 0), (1 \triangle 1), (2 \triangle 1), (3 \triangle 1)\}_\triangle \rangle \end{aligned}$$

The association  $\mathcal{A}$  is thus like the bistring models of Kornai (1991, 1995), which were three-tuples made up of two autosegment tiers and their association relation. Here, because each tier is defined with MSO and the precedence relation  $\triangleleft^*$ , we know from Büchi (1960)'s result that the set of possible strings on each tier is a *regular* stringset. Crucially, no precedence is defined *between* the tiers (unlike Bird & Klein, 1990); without this restriction, it is not clear how powerful the formalism will be.

As of yet, the association relation  $\triangle$  is yet unrestricted, and can describe any number of associations between the tiers. As a first step, we want to at a bare minimum be able to restrict the possible models to those that conform to the No-Crossing Constraint part of Goldsmith (1976)'s Well-Formedness Condition (WFC), as well as directional, One-to-One Association, which makes explicit the constraint that tonemes can only associate 'after' all preceding tonemes have been associated ('after' meaning in the direction of association, which can apply either right-to-left or left-to-right):

- (7) 1. *The No-Crossing Constraint.* Association lines do not cross.
2. *One-to-One Association.* Autosegments associate one-to-(at least) one, right-to-left or left-to-right.

We will show that with these two constraints, it appears that the yields of AP diagrams will be at most regular. What follows uses MSO to define these constraints as axioms which restrict us to only those models whose association relation  $\triangle$  follows the constraints in (7).

**3.2 Definitions** The following define propositions to be used in the two well-formedness axioms. The first goal is to first extend the association relation  $\triangle$ , which is defined as a relation between single positions, to a relation between sets on the tonal tier and positions on the segmental tier. This allows the axioms to refer to many-to-one associations between tonal autosegments and segments. The final goal is to build, using MSO, the machinery necessary to define the two axioms, the No-Crossing Constraint and One-to-One Association, which restrict us to the well-formed autosegmental representations. Note that all are built out of more basic MSO statements; as such, they do not expand the power of the formalism.

The first show that irreflexive precedence,  $\triangleleft^+$ , and immediate precedence,  $\triangleleft$ , can be defined from  $\triangleleft^*$ .

$$(8) \quad x \triangleleft^+ y \stackrel{\text{def}}{=} (x \triangleleft^* y) \wedge (\neg \exists z) [(x \triangleleft^* z) \wedge (z \triangleleft^* y)]$$

$$(9) \quad x \triangleleft y \stackrel{\text{def}}{=} (x \triangleleft^+ y) \wedge (\neg \exists z) [(x \triangleleft^+ z) \wedge (z \triangleleft^+ y)]$$

The next definitions extend precedence to sets of variables. In order to do this, we need to be able to pick out the first and last variables in a set, as well as sets containing these variables:

$$(10) \quad \text{First}(X, x) \stackrel{\text{def}}{=} X(x) \wedge (\forall z) [X(z) \rightarrow (x \triangleleft^* z)]$$

$$(11) \quad \text{Last}(X, x) \stackrel{\text{def}}{=} X(x) \wedge (\forall z) [X(z) \rightarrow (z \triangleleft^* x)]$$

$$(12) \quad \text{First}(X, Y) \stackrel{\text{def}}{=} (\exists x) [\text{First}(Y, x) \wedge X(x)]$$

$$(13) \text{ Last}(X, Y) \stackrel{\text{def}}{=} (\exists x) [\text{Last}(Y, x) \wedge X(x)]$$

One set  $X$  precedes ( $\triangleleft^+$ ) another set  $Y$  if all members of  $X$  precede all members of  $Y$ . A set  $X$  *immediately* precedes another set  $Y$  if it both precedes  $Y$  and the last member of  $X$  immediately precedes the first member of  $Y$ .

$$(14) X \triangleleft^+ Y \stackrel{\text{def}}{=} (\forall x, y) [X(x) \wedge Y(y) \rightarrow x \triangleleft^+ y]$$

$$(15) X \triangleleft Y \stackrel{\text{def}}{=} (X \triangleleft^+ Y) \wedge (\forall x, y) [(\text{Last}(X, x) \wedge \text{First}(Y, y)) \rightarrow x \triangleleft y]$$

Next, we define *contiguous* sets, as we are not interested in arbitrary sets of positions. A contiguous set of positions is one in which there are two endpoints in between which all positions in the set lie.

$$(16) \text{Contig}(X) \stackrel{\text{def}}{=} \exists(x, y)(\forall z) [(X(x) \wedge X(y)) \wedge ((x \triangleleft^* z) \wedge (z \triangleleft^* y) \leftrightarrow X(z))]$$

We will also want to isolate *maximal* associated sets, which requires a notion of a subset relation between sets. We are also going to want to pick out unary sets.

$$(17) X \subseteq Y \stackrel{\text{def}}{=} (\forall x) [X(x) \rightarrow Y(x)]$$

$$(18) \text{Unary}(X) \stackrel{\text{def}}{=} (\forall x, y) [(X(x) \wedge X(y)) \rightarrow (x \approx y)]$$

We can now begin to define association between sets of symbols on the tonal tier and individual positions on the segmental tier. First, it is necessary to define a set that is *potentially* associated to a position. A set of tonal autosegments  $X$  is potentially associated with a TBU  $y$  if all members of the set are associated with it.

$$(19) \text{Pot}(X, y) \stackrel{\text{def}}{=} (X \subseteq W_T) \wedge (y \in W_\Sigma) \wedge (\forall x) [X(x) \rightarrow x \triangle y]$$

We then define association between a set of positions and a single position as only true if the set is potentially associated with that position, contiguous, and *maximal* (i.e., if  $X \triangle y$ , all sets that are also potentially associated with  $y$  are subsets of  $X$ ).

$$(20) X \triangle y \stackrel{\text{def}}{=} (\forall x)[X(x) \rightarrow (x \triangle y)] \wedge \text{Contig}(X) \wedge (\forall Z) [\text{Pot}(Z, y) \rightarrow (Z \subseteq X)]$$

Note that association is in this way asymmetrical; this explicitly defines the association of a set of tonal phonemes to a single TBU, but not the other way around. In Section 3.4, which defines the yield, the reasons for this are made clear.

It is also important to talk about different kinds of association. It will be important to distinguish between one-to-one and one-to-many associations.

$$(21) \text{One-to-one}(X, x) \stackrel{\text{def}}{=} [(X \triangle x) \wedge \text{Unary}(X) \wedge (\forall y) [(X \triangle y) \rightarrow (x \approx y)]]$$

Note that we *never* want to count many-to-many associations as a single association. This will actually constitute our first axiom, as it always needs to be true.

**Axiom 1** *Many-to-many associations are disallowed.*

$$(\forall X, x) [(X \triangle x) \rightarrow (\text{Unary}(X) \vee (\forall y) [(X \triangle y) \rightarrow (x \approx y)])]$$

We can now define our well-formedness axioms.

**3.3 Well-formedness axioms** The axiomatic definitions of the constraints on AP representations given in (7) are as follows:

**Axiom 2** *The No-Crossing Constraint (=7a).* If  $v$  is associated with  $x$ , and  $w$  with  $y$ ,  $v \triangleleft^* w$  if and only if  $x \triangleleft^* y$ .

$$(\forall v, w, x, y) [(v \triangle x) \wedge (w \triangle y) \rightarrow ((v \triangleleft^* w) \leftrightarrow (x \triangleleft^* y))]$$

An axiom like Axiom 2 above appears in Bird & Klein (1990), however theirs refers to precedence *between* tiers; i.e., it refers to tonal symbols preceding or following segmental symbols. We consider this a crucial difference between the work here and that of Bird & Klein (1990); for one, it is not clear whether there *should* be a precedence relation between tiers. Not only does it not appear necessary, but, as already mentioned, it may increase the formalism beyond the expressive power that we want.

**Axiom 3** *One-to-One Association (either left to right or right to left) (7b).* For all associated  $X$  and  $x$ , if there is a  $y$  that precedes (or follows)  $x$ , there must be some  $Y$  associated with it, and  $X$  must be unary unless  $x$  or  $X$  is the last (first) symbol in the segmental tier.

$$\begin{aligned} (\forall X, x) & [(X \triangle x) \rightarrow (\forall y) [(y \triangleleft x) \rightarrow (\exists Y) [Y \triangle y]] \wedge \\ & (\text{One-to-one}(X, x) \vee (\text{Last}(W_\Sigma, x) \vee \text{Last}(X, W_\Sigma)))] \\ \vee (\forall X, x) & [(X \triangle x) \rightarrow (\forall y) [(x \triangleleft y) \rightarrow (\exists Y) [Y \triangle y]] \wedge \\ & (\text{One-to-one}(X, x) \vee (\text{First}(W_\Sigma, x) \vee \text{First}(X, W_\Sigma)))] \end{aligned}$$

As both right-to-left and left-to-right association are attested, this axiom must come in the form of a conjunction, with individual languages being able to choose from one side of the conjunction.

Let us confirm how the example in (3) conforms to these two axioms. Example (3) and its model (6) are repeated below in (22):

$$(22) \quad \begin{array}{cccc} \tau_0 & \tau_1 & \tau_2 & \tau_3 \\ \text{H} & \text{L} & \text{H} & \text{L} \\ | & & \diagdown & \diagup \\ \mu & & \mu & \\ \sigma_0 & & \sigma_1 & \end{array} \quad \begin{aligned} \mathcal{W}_T &= \langle \{0, 1, 2, 3\}_{W_T}, \{(0 \triangleleft^* 0), \dots, (3 \triangleleft^* 3)\}_{\triangleleft^*}, \{0, 2\}_{\text{H}}, \{1, 3\}_{\text{L}} \rangle \\ \mathcal{W}_\Sigma &= \langle \{0, 1\}_{W_\Sigma}, \{(0 \triangleleft^* 0), (0 \triangleleft^* 1), (1 \triangleleft^* 1)\}_{\triangleleft^*}, \{0, 1\}_{\mu} \rangle \\ \mathcal{W}_A &= \langle W_T, W_\Sigma, \{(0 \triangle 0), (1 \triangle 1), (2 \triangle 1), (3 \triangle 1)\}_{\triangle} \rangle \end{aligned}$$

The reader can confirm that the two sets on the T tier associated with a position on the  $\Sigma$  tier are  $\{\tau_0\} \triangle \sigma_0$  and  $\{\tau_1, \tau_2, \tau_3\} \triangle \sigma_1$ . Note that they are contiguous and maximal. These two set-position relations also conform to the two axioms; the model obeys Axiom 2 because  $\tau_0$  precedes  $\tau_1, \tau_2$ , and  $\tau_3$  and  $\sigma_0$  precedes  $\sigma_1$  on the  $\Sigma$  tier, and it obeys Axiom 3 because  $\sigma_0$ , which precedes  $\sigma_1$ , has a set (namely  $\{\tau_0\}$ ) associated with it. Also, only  $\sigma_1$ , which is the last symbol on its tier, is associated with more than one tone. The reader can also confirm that the other example, (4), also obeys these two axioms.

**3.4 Yield** We can then represent a linearized *yield* of the diagram by creating a new alphabet based on the associated sets of segments. Returning to the example in (3)/(22), recall that the associated sets are  $\{\tau_0\} \triangle \sigma_0$  and  $\{\tau_1, \tau_2, \tau_3\} \triangle \sigma_1$ ; replacing the positions with their respective symbols, we can write the associations as  $(\{\text{H}\}, \mu)$  and  $(\{\text{LHL}\}, \mu)$ . In the other example (4), repeated below in (23) the associated sets are  $(\{\text{L}\}, \mu)$ ,  $(\{\text{H}\}, \mu)$ ,  $(\{\text{H}\}, \mu)$ , and  $(\{\text{H}\}, \mu)$ .

$$(23) \quad \begin{array}{cccc} \tau_0 & \tau_1 & & \\ \text{L} & \text{H} & & \\ | & | & \diagdown & \diagup \\ \mu & \mu & \mu & \mu \\ \sigma_0 & \sigma_1 & \sigma_2 & \sigma_3 \end{array}$$

In this way, we can encode the surface results of association—for example, morae whose vowels have a tonal contour. In example (22), the surface form of  $(\{\text{LHL}\}, \mu)$  would be a mora with a falling-rising tone, which may be transcribed as  $\check{\mu}$ . Note that this is why there is an asymmetry between the many-to-one associations defined in Section 3.2 and 3.3; the TBUs act as timing segments forming the basis of the yield alphabet, and in this way it is important to keep track of the associations to each TBU.

In order to keep track of symbols like  $\check{\mu}$ , which represent associated segments, we first define the *general yield alphabet*  $\Gamma$  as the set of all possible associations. Below,  $\mathbf{T}^*$  refers to the set of all possible strings of symbols in the alphabet  $\mathbf{T}$ .

**Definition 1** *The general yield alphabet is  $\Gamma = \mathbf{T}^* \times \Sigma$*

Note that  $\Gamma$  is an infinite alphabet. Assuming that languages only employ finite alphabets for their surface representations,  $\Gamma_L$  for some language will be a finite subset of  $\Gamma$ . In the examples above,  $(\{\text{LHL}\}, \mu) \in \Gamma_L$ , for example, but  $(\{\text{HLHLHHHLHLH}\}, \mu)$  might not be in  $\Gamma_L$ . In the notation below,  $|w|$  refers to the length of string  $w$ ;  $\max(S)$  and  $\text{card}(S)$  refer to the length of the longest string in a set  $S$  and the cardinality of  $S$ , respectively.

**Lemma 1** *A language-specific yield alphabet  $\Gamma_L \subseteq \Gamma$  of a language is finite iff there is some maximum bound on the length of strings in its left projection,  $\pi_0(\Gamma_L)$ :<sup>1</sup>*

$$(\exists n \in \mathbb{N}) [|\Gamma_L| = n] \iff (\exists m \in \mathbb{N}) [\max(\pi_0(\Gamma_L)) = m]$$

This leads us to one final universal axiom:

**Axiom 4** *For any set of autosegmental diagrams in a language,  $\Gamma_L$  must be finite.*

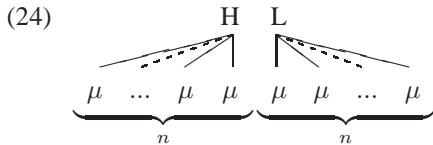
We can then build a linear *yield* of an autosegmental model  $\mathcal{W}$ :

**Definition 2**

$$\text{Yield}(\mathcal{W}_A) = \gamma_0\gamma_1\dots\gamma_n \in \Gamma_L^* \text{ where } \gamma_i = (w \in \mathbf{T}^*, \sigma_i \in \Sigma) \text{ s.t. } w \Delta \sigma_i$$

**3.5 Axioms and the regularity conjecture** With the three axioms and our yield defined, it is now possible to discuss the computational complexity of autosegmental diagrams. We consider *sets* of AP association models; the set of the combined yields of a set of AP models is a set of strings. As such, we can classify the complexity of these sets of strings just as any other set of strings.

As an example, the following diagram represents an infinite set of AP association models in which there are  $2n$   $\mu$  TBUs,  $n$  being any integer, from the center of which an H and an L spread outwards.



This is clearly an odd example not expected to be seen in natural language, and it blatantly violates One-to-One Association, Axiom 3, as both the first and last members of the tonal tier are part of non-unary associations. However, without this axiom, it is a logically possible set of diagrams. The yield of this set of diagrams, with  $\check{\mu}$  representing  $\{H, \mu\}$  associations and  $\grave{\mu}$  representing  $\{L, \mu\}$  associations, looks like this:

<sup>1</sup> The proof of Lemma 1 is as follows:

**Proof:** For  $(\exists n \in \mathbb{N}) [|\Gamma_L| = n] \Rightarrow (\exists m \in \mathbb{N}) [\max(\pi_0(\Gamma_L)) = m]$ :

If  $|\Gamma_L|$  is finite, then there is some member  $w$  of  $\pi_0(\Gamma_L)$  such that  $(\forall v \in \pi_0(\Gamma_L)) [|w| \geq |v|]$ . The length of  $w$  is then  $m \in \mathbb{N}$  s.t.  $\max(\pi_0(\Gamma_L)) = m$ .

For  $(\exists n \in \mathbb{N}) [|\Gamma_L| = n] \Leftarrow (\exists m \in \mathbb{N}) [\max(\pi_0(\Gamma_L)) = m]$ :

If there is some  $w = \max(\pi_0(\Gamma_L))$ , only a finite number of strings can be of equal or lesser length than  $w$ , given that  $\mathbf{T}$  is a finite alphabet. As  $\Sigma$  is also finite, the maximum size of  $\Gamma_L$  is  $\text{Card}(\mathbf{T}^{\geq |w|} \times \Sigma)$ , which is equal to some  $n \in \mathbb{N}$ .

□



$$(25) \quad \underbrace{\acute{\mu} \dots \acute{\mu} \acute{\mu}}_n \underbrace{\grave{\mu} \grave{\mu} \dots \grave{\mu}}_n$$

This is thus a homomorphism of the context-free pattern in (1b),  $a^n b^n$ . However, Axiom 3, sets of AP diagrams that generate this stringset are legal. In general, it appears that the well-formedness conditions originally posited for AP will keep the stringset yields of sets of AP models conforming to Axioms 1, 2, 3, and 4 within the regular region of complexity. This is formalized below in Conjecture 1.

**Conjecture 1** *The set of strings resulting from the yields of any set of autosegmental representations which at least follow Axioms 1, 2, 3, and 4 will be regular.*

## 4 Conclusions and Further Work

This paper has presented the beginnings of a framework to examine the complexity of Autosegmental Phonology. It has also formalized a conjecture that the yields of AP diagrams corresponding to the well-formedness axioms stay within the regular region. There remains much work to be done. The first step is proving the conjecture; given that everything discussed here has been rigorously defined, the formal machinery for this is in place. However, as the MSO constraints discussed in this paper are as general as possible, the next step is empirical exploration of language-specific constraints, to see whether or not they can be encoded into MSO, or, as a next step, whether the entire enterprise can be accomplished with less powerful logics, such as first order logic.

The current proposal deals solely with single autosegmental diagrams and their resulting strings; however, of course, like other phonological theories, changes from underlying form to surface representation are a central part of AP. An interesting potential for the framework proposed here is the ability to look at UR/SR *pairs* of diagrams, and likewise, UR/SR pairs of their yields. In this way, AP processes can be directly compared to recent work (Chandlee & Heinz, 2012; Heinz & Lai, 2013; Chandlee, in prep.) studying the computational properties of linear UR/SR pairs.

## References

- Bird, Steven & T. Mark Ellison (1994). One-level phonology: Autosegmental representations and rules as finite automata. *Computational Linguistics* 20.
- Bird, Steven & E. Klein (1990). Phonological events. *Journal of Linguistics* 26, 33–56.
- Büchi, J. Richard (1960). Weak second-order arithmetic and finite automata. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* 6, 66–92.
- Chandlee, Jane & Jeffrey Heinz (2012). Bounded copying is subsequential: Implications for metathesis and reduplication. *Proceedings of the 12th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, Association for Computational Linguistics, Montreal, Canada, 42–51.
- Chandlee, Jane (in prep.). *On the Nature of Locality in Phonological Processes*. Ph.D. thesis, The University of Delaware.
- Chomsky, Noam (1956). Three models for the description of language. *IRE Transactions on Information Theory* 113–124.
- Chomsky, Noam & Morris Halle (1968). *The Sound Pattern of English*. Harper & Row.
- Goldsmith, John (1976). *Autosegmental Phonology*. Ph.D. thesis, Massachusetts Institute of Technology.
- Goldsmith, John A. (1979). *Autosegmental Phonology*. Garland Press.
- Goldsmith, John A. (1990). *Autosegmental & Metrical Theory*. Basil Blackwell, Inc.
- Graf, Thomas (2010a). Comparing incomparable frameworks: A model theoretic approach to phonology. *University of Pennsylvania Working Papers in Linguistics* 16:2, p. Article 10.
- Graf, Thomas (2010b). *Logics of Phonological Reasoning*. Master's thesis, University of California, Los Angeles.
- Heinz, Jeffrey (2009). On the role of locality in learning stress patterns. *Phonology* 26, 303–351.
- Heinz, Jeffrey (2010). Learning long-distance phonotactics. *Linguistic Inquiry* 41, 623–661.
- Heinz, Jeffrey & Regine Lai (2013). Vowel harmony and subsequentiality. Kornai, Andras & Marco Kuhlmann (eds.), *Proceedings of the 13th Meeting on Mathematics of Language*, Sofia, Bulgaria.
- Johnson, C. Douglas (1972). *Formal aspects of phonological description*. Mouton.
- Kaplan, Ronald & Martin Kay (1994). Regular models of phonological rule systems. *Computational Linguistics* 20, 331–78.
- Kay, Martin (1987). Nonconcatenative finite-state morphology. *Proceedings, Third Meeting of the European Chapter of the Association for Computational Linguistics*, 2–10.
- Kaye, Jonathan, Jean Lowenstamm & Jean-Roger Vergnaud (1990). Constituent structure and voerment in phonology. *Phonology Yearbook* 7, 193–231.

- Kornai, András (1991). *Formal Phonology*. Ph.D. thesis, Stanford University.
- Kornai, András (1995). *Formal Phonology*. Garland Publication.
- de Lacy, Paul (2011). Markedness and faithfulness constraints. Oostendorp, Marc Van, Colin J. Ewen, Elizabeth Hume & Keren Rice (eds.), *The Blackwell Companion to Phonology*, Blackwell.
- Leben, William R. (1973). *Suprasegmental phonology*. Ph.D. thesis, Massachusetts Institute of Technology.
- Leben, W. R. (1978). The representation of tone. Fromkin, Victoria (ed.), *Tone—A Linguistic Survey*, Academic Press, 177–219.
- Partee, Barbara, Alice ter Meulen & Robert Wall (1993). *Mathematical Methods in Linguistics*. Dordrecht, Boston, London: Kluwer Academic Publishers.
- Potts, Christopher & Geoffrey K. Pullum (2002). Model theory and the content of OT constraints. *Phonology* 19, 361–393.
- Rogers, James, Jeffrey Heinz, Margaret Fero, Jeremy Hurst, Dakotah Lambert & Sean Wibel (2013). Cognitive and sub-regular complexity. *Formal Grammar*, Springer, vol. 8036 of *Lecture Notes in Computer Science*, 90–108.
- Sagey, Elizabeth (1986). *The Representation of Features and Relations in Non-Linear Phonology*. Ph.D. thesis, Massachusetts Institute of Technology.
- Sagey, Elizabeth (1988). On the ill-formedness of crossing association lines. *Linguistic Inquiry* 19, 109–118.
- Wiebe, Bruce (1992). *Modelling Autosegmental Phonology with Multi-Tape Finite State Transducers*. Master's thesis, Simon Fraser University.