LOCALITY AND NON-LINEAR REPRESENTATIONS IN TONAL PHONOLOGY

by

Adam Jardine

A dissertation submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Linguistics and Cognitive Science

Spring 2016

LOCALITY AND NON-LINEAR REPRESENTATIONS IN TONAL PHONOLOGY

by

Adam Jardine

Approved: ____
Benjamin Bruening, Ph.D.
Chair of the Department of Linguistics and Cognitive Science

Approved: ____
George H. Watson, Ph.D.
Dean of the College of Arts and Sciences

Approved: ____
Ann L. Ardis, Ph.D.
Senior Vice Provost for Graduate and Professional Education

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: ______________________________

Jeffrey Heinz, Ph.D.
Professor in charge of dissertation

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: ______________________________

Irene Vogel, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: ______________________________

Peter Cole, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: ______________________________

Rémi Eyraud, Ph.D.
Member of dissertation committee

ACKNOWLEDGEMENTS

It takes a village to write a dissertation, and I have been fortunate in my graduate career to be in the company of some truly amazing human beings. First, I can't possibly fully communicate my appreciation for my advisor, Jeff Heinz, but I will try and fail anyway. Jeff always puts his students first, and I have benefited tremendously from that. His relentless positivity has kept me motivated through many a rough period, and his dedication to intellectual and philisophical clarity will serve as a model for years to come. I owe Jeff much, including many drinks.

The rest of my committee have been outstanding guides to the murky waters of beginning a career in research. Once, when I was struggling with a proof, Rémi Eyraud admonished me that "You are thinking in English. You have to think in *math*!" I will probably spend the rest of my career trying to live up to that statement. Peter Cole taught me how to write clearly, demanded me to think clearly, and showed me how to navigate the minefield of publishing a linguistics paper (even after being rejected twice). I've learned much from many great discussions with Irene Vogel, in her seminars, research groups, and elsewhere, and her skepticism of the computational approach has rightfully kept me on my toes.

There are of course many other researchers who have left a huge mark on this work and on myself in general. The ideas of Jim Rogers course through this dissertation, and even though it has been a few years since he visited UD, he has remained a massive influence (and a great guy). Similarly, it has been great to interact with and learn from Thomas Graf. Jane Chandlee deserves special mention for somehow effortlessly being a role-model, collaborator, and friend, all at the same time. Kevin McMullin has done the same, although we still need to write a paper together. I thank Angeliki Athanasopoulou and Kristian for putting up with the long and arduous task of writing the Banyaduq paper, as well as the plentiful input Tim Mckinnon gave me on that and other phonological issues. Many thanks also to Bert Tanner and his students Jie Fu, Kostas Karydis, and Prasanna Kannappan for putting up with a linguist trying to keep up with robotics & control work.

I have also benefited greatly from being able to work with some great phonologists, who despite being famous and important have been extremely friendly and willing to help me out on some project or another. In particular, it has been great to interact with Bill Idsardi, who has dispensed untold amounts of advice over the past couple of years. Many thanks also go to Harry van der Hulst, Larry Hyman, and Haruo Kubozono. I also can't go without mentioning the professors who got me into linguistics in the first place, at UNH all those years ago—Rochelle Lieber, Aya Matsuda, and Naomi Nagy. Thank you for giving me something meaningful to do with my life.

I also have to thank the department that has nurtured me over the past five years. First mention goes to my PhD cohort—Gordon P. Hemsley, Amanda Payne, Albeiro Ramirez, Justin Rill, David Rubio, Xin Tao, and honorary member Rachel Bristol. Even though some have already gone their separate ways, from the beginning they have been both terrifyingly smart intellectual rivals and fantastic friends, and I wouldn't have gotten this far without them. I also thank those who came before me, for showing me the way—Tim O'Neill, Demitrios Skordos, Rebecca Jia, Joo Young Kim, Eman Al Khalaf, Lan Kim, Yugyoung Park, and Cesar Koirala—and those who came after, and let me play wise older student from time to time—in particular, Hyun Jin Hwangbo, Mike Donovan, Taylor Lampton Miller, and Kristina Strother-Garcia. I also thank Jane Creswell, for keeping everything in order all these years, and to Satoshi Tomioka, Gaby Hermone, Robin Andreasen, Ben Parrell, Ben Bruening, Arild Hestvik, Fred Adams, Nina Straitman, and Yaping Tsai for making the UD Department of Linguistics & Cognitive Science a fun and stimulating learning environment. I wouldn't have wanted to do my PhD anywhere else.

I am thankful also for much institutional support through my graduate years:

teaching assistant fellowships from the UD Department of Linguistics & Cognitive Science, research assistantships from NSF grants #1123692 and #1035577, and a Graduate Research Fellowship and Dissertation Fellowship from the UD Office of Graduate & Professional Education.

The friendships of Elliot Liebert, Lacey Hebert, Kyle Dowley, Becca Asaki, Steve Gintz, Laura Fink, Joe and Liz Morency, Jeremy and Haruka Fesi, Akiko and Yoon Choi, Rina and Aaron Gravatt, and Dom, Nami, and Anna Natale, none of whom had anything to do with my studies, nonetheless made them a lot easier to get through. Thank you for reminding me that there is a world out there outside of linguistics. There were also many dozens of musicians whose work saw me through these sometimes trying years, in particular Nick Cave, Kazuki Tomokawa, and Carly Rae Jepsen.

Finally, I thank Dick Jardine for instilling in me a curiosity about the world around me, Cary Jardine for giving me a love of books and writing, and Matt Jardine for simultaneously being a great brother and a great friend. It has been so great to be near you guys again, and your support has meant everything. To my family back east—Katsumi Yamamoto, Narumi Ueda, and Shinichi Ito—I am eternally grateful for your continued encouragement, even though we're so far away now. And last, but certainly not least, to Yuki Jardine, my incredible wife, who has always been positive and encouraging despite the stress and poverty that comes with being married to a grad student. You have been my motivation and inspiration. This dissertation is dedicated to you, circles and everything.

TABLE OF CONTENTS

ABSTRACT

This dissertation provides support for the hypothesis that surface well-formedness in phonological tone patterns is governed by language-specific, *local* constraints over autosegmental representations. The particular notion of locality invoked in this dissertation is that of *banned substructure constraints*, which are drawn from the theory of computation, formal language theory, and formal learning theory (McNaughton and Papert, 1971; García et al., 1990; Rogers et al., 2013). Essentially, any pattern describable with such constraints is local because the well-formedness of a structure with respect to the pattern is based entirely on its composite substructures of a fixed size. The primary novel contribution of the current work is to extend this notion of computational locality from strings to autosegmental structures by way of mathematical graph theory, and to develop a theory of tonal well-formedness based in banned substructure constraints over autosegmental representations. Through analyses of attested edge-based, quality-specific, and positional tone association patterns, as well as long-distance patterns, it is shown that such a theory can describe a range of major tonal generalizations, including ones beyond the power of both string-based local theories and standard explanations of tone in Optimality Theory. Furthermore, a local theory of constraints excludes unattested patterns requiring *global* evaluation that are predicted by other theories. Finally, it is discussed how banned substructure constraints can be connected to a restrictive theory of phonological input/output generalizations, and that there is a method for learning them.

A secondary contribution of this dissertation is show that autosegmental representations are string-like in that they can be derived through the *concatenation of graph primitives*. Essentially, important properties of autosegmental representations can be seen as emerging from the concatenation of a finite alphabet of primitives, just as strings are built out of a finite alphabet of symbols. This novel approach to defining autosegmental representations not only makes the correct empirical prediction that languages cannot have unbounded 'contouring', it also allows for direct comparison of autosegmental grammars to string grammars. It is also shown how this notion of concatenation can be recruited for understanding input/output generalizations, and how it can be used to learn autosegmental grammars from string inputs.

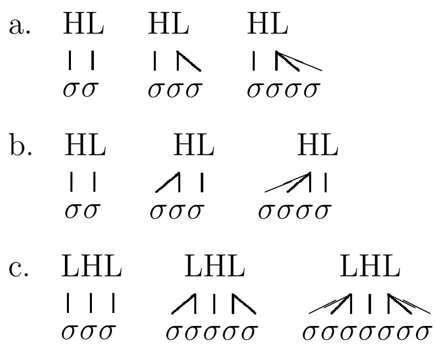
Chapter 1

INTRODUCTION

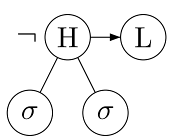
1.1 Overview

The central hypothesis of this dissertation is that surface well-formedness in phonological tone patterns is governed by a restrictive set of *local* constraints over autosegmental representations. Here, 'local' is a well-defined term meaning that the constraints are *inviolable*, *language-specific*, and that they are only able to *ban sub-structures*. This notion of locality is grounded in the theory of computation, formal language theory, and formal learning theory, and is drawn from work showing that such constraints over strings provide a restrictive, learnable theory of the typologies of stress and segmental phonology (Heinz, 2007, 2009, 2010a; Heinz et al., 2011; McMullin and Hansson, to appear; Rogers et al., 2013; Chandlee, 2014). The primary novel contribution of the current work is to extend the notion of computational locality from strings to autosegmental structures by way of mathematical *graph theory*, and to show that this provides for an attractive theory of well-formedness in phonological tone for a number of reasons. First, it can describe a range of major tonal generalizations, including those that are beyond the power of both the string-based local theories mentioned above and standard explanations of tone in Optimality Theory. Second, these local constraints can be connected to a restrictive theory phonological input/output generalizations, and there is a method for learning them. Finally, the computationally simple nature of local constraints excludes unattested, *globally*-computed patterns predicted by other theories.

For example, (1.1) illustrates two attested patterns and one unattested pattern in tone. The example in (1.1a) shows the general state of affairs in Mende (Leben, 1973, 1978; Dwyer, 1978), in which multiple association is only allowed to occur on the right edge of the word. Hausa (Newman, 1986, 2000) has an opposite pattern in which multiple association only appears on the left, as in (1.1b).[1] Finally, an unattested pattern is given in (1.1c) in which a single H tone associates closest syllable to the center of the word (only examples with odd-numbered syllables are given):

(1.1)


A theory of tonal phonology that is fundamentally local explains why (1.1a) and (1.1b) are attested but (1.1c) is not. The attested patterns can be described by constraints which ban autosegmental substructures; for example, that the initial H does not spread in Mende can be captured with the constraint in (1.2). While the exact notation will be discussed in Chapters 4 and 5, (1.2) identifies and bans a structure in which a nonfinal H has spread to more than one syllable.

(1.2) ¬


With such a structure banned in Mende, the initial H in (1.1a) cannot spread and so the L does instead. For Hausa, in contrast, this structure would not be banned, and so the initial H is free to spread. Complete analyses of these patterns will be given in Chapter 5, but they can be fully described by listing forbidden substructures like the one in (1.2). However, banned substructure constraints cannot describe the pattern in (1.1c)—intuitively, this is because there is no finite set of substructures we

[1] Full discussion of the data in these languages can be found in Chapter 2.

can ban in order to ensure that the number of syllables on either side of the H tone are equal. Capturing this pattern instead requires *global* evaluation in that it is necessary to count the number of syllables on either side. Thus, a *local* theory of tone excludes (1.1c) from the typology.

This touches on an important contribution of the current work, which is to further understand the nature of well-formedness constraints in phonology. The exploration of the nature of phonological well-formedness has roots in early autosegmental phonology, in which constraints on well-formed representations partly explained phonological patterns through constraining the application of rules (Goldsmith, 1976; Clements, 1977; Archangeli and Pulleyblank, 1994). Surface well-formedness came to the forefront in Optimality Theory (OT; Prince and Smolensky, 1993, 2004), which argued that phonological processes are *driven* by surface MARKEDNESS constraints which judge the relative well-formedness of potential output structures. This lead to insightful analyses of tonal phonology based on surface well-formedness in autosegmental representations (Meyers, 1997; Yip, 2002; Zoll, 2003).

However, Eisner (1997b) shows how the commonly-used MARKEDNESS constraint family of ALIGN constraints are capable of producing exactly the unattested pattern in (1.1c). This illustrates an extremely important point: any theory of phonology which aims to meaningfully distinguish between possible and impossible phonological patterns must include some notion of what a possible well-formedness constraint is. de Lacy (2011), echoing Eisner's concerns, calls for *constraint definition languages* which make explicit the possible range of constraints and how they are interpreted. Similar work in OT constraining its well-formedness constraints from the perspective of mathematical logic (Potts and Pullum, 2002), automata theory (Riggle, 2004), and perceptual and articulatory factors (Hayes et al., 2004) also exist. However, it is also known that the optimization at the center of OT is extremely powerful, and can generate complex patterns with very simple constraints (Eisner, 1997b; Riggle, 2004; Gerdemann and Hulden, 2012), thus obscuring the value of having a restrictive theory of MARKEDNESS.

In contrast, the theory of local constraints advanced by this dissertation is based in grammars of logical statements whose relative expressivity is well-defined (Büchi, 1960; Rogers and Pullum, 2011; Rogers et al., 2013; Graf, 2010a,b). As discussed in detail in Chapter 3 and Chapter 8, a logical language of statements referring to phonological structures provide a well-defined set of constraints whose interpretation is explicit (and thus constitute a constraint definition language in the sense of de Lacy (2011)). Logical statements which can only *ban* structures (e.g., (1.2)) are among the most restrictive kinds of logical statement, and are fundamentally local in a computational sense, which shall be discussed further momentarily. By showing that these statements can describe a wide range of tonal patterns, the primary contribution of this dissertation is a well-defined, restrictive theory of tonal well-formedness constraints which retains the insights of MARKEDNESS but also non-trivially distinguishes attested patterns such as (1.1a) and (1.1b) from logically possible, yet unattested patterns such as (1.1c).

In service of explicitly defining constraints and their expressiveness, a secondary contribution of this dissertation is to understand the properties of autosegmental representations in terms of *concatenation of graph primitives*. This novel approach to autosegemental representations, put forward in Chapter 4, allows not only for the precise definition of logical constraints over these representations, but also for the direct comparison of the expressivity of autosegmental grammars to string grammars. This is particularly important in establishing that tonal patterns which are 'long-distance' over strings—i.e., not describable by local string grammars—can be described by local grammars over autosegmental structures. It also, as discussed in Chapter 8, means that it is possible to learn local autosegmental grammars directly from string input.

Finally, as the focus of this dissertation is on surface well-formedness, a potential criticism of these results is that they do not apply to phonological *transformations* from underlying form to surface form. This is not true, however. The surface locality of the surface generalizations analyzed in this dissertation is a *fact* independent of how one might derive them from phonological transformations. In other words, it would make any theory of phonological transformations in tone stronger if it aimed to adhere to this surface locality in the output.

Another advantage, then, of defining the aforementioned concepts in mathematical logic and graph theory is that they can be directly applied to phonological transformations. Chapter 7 extends graph concatenation to graphs representing phonological transformations to build a restrictive theory of input/output correspondence. This then allows the graph constraints defined in Chapter 5 to be applied directly to transformation graphs in order to describe the tone patterns reviewed in Chapter 2 in terms of changes from underlying autosegmental representations to surface autosegmental representations. Thus, while the focus of the dissertation is on surface well-formedness, there is a demonstrated way to integrate the typological *fact* that tonal well-formedness is local over surface representations into a theory of phonological transformations.

1.2 Locality and Phonological Explanation

As mentioned above, this notion of locality is based on banned substructures. As this differs from other definitions of 'local' that have been invoked in phonological theory, it is worth discussing this difference and articulating that a substructure-based notion of locality is a meaningful one for phonological theory because it is has an independent basis in the theories of computation and learnability.

Locality in phonology has often been defined in terms of *adjacency* (Odden, 1994; Gafos, 1996; Chiośain and Padgett, 2001). To give an autosegmental example, Odden (1994)'s LOCALITY CONDITION is given below in (1.3).

(1.3) LOCALITY CONDITION: (Odden, 1994, (20)) In a relation involving A,B and the nodes α, β which they immediately dominate, nothing may separate α and β unless it is on a distinct plane from that of α or β.
[where a *plane* is a tier and the one dominating it – AJ]

Schematically, in (1.4a) and (b) below, α and β are local under this definition, but not in (c) and (d) (because γ intervenes).

(1.4)	a.	A	B	b.	A	C	B	c.	A	C	B	d.	A		B
		|	|		|		|		|	|	|		|		|
		α	β		α		β		α	γ	β		α	γ	β

(Odden, 1994, (21–2))

Restricting phonological processes or relations to adjacent units then can restrict the set of possible grammars predicted by the theory. This notion of locality thus often relies on *underspecification*, i.e. the omission of certain kinds of information tiers, to capture 'long-distance' generalizations (Liberman and Prince, 1977; Halle and Vergnaud, 1982; Steriade, 1987; Odden, 1994). This can be seen schematically in (1.4b), where α and β are adjacent on the tier even though their associated units A and B are separated by an intervening C (which is not associated to anything on the α/β tier).

The notion of locality based on banned substructures also aims to restrict the range of possible grammars, but it is based on what Chandlee (2014) calls 'contiguity', not adjacency, and it is grounded in the theory of computation. When we consider the kinds of computations that are logically possible, those that are necessary to describe, evaluate, and learn the patterns describable by banned substructure constraints are among the least complex (McNaughton and Papert, 1971; García et al., 1990; Rogers et al., 2013; Heinz, 2010b). To get a sense why, let us look at the evaluation of strings with respect to the pattern of Kagoshima Japanese, which, as shall be discussed further in Chapter 3, is describable by banned substructure constraints over strings. In words of Kagoshima Japanese, words have a single H tone on either the penultimate or ultimate mora; for example, LLHL is well-formed with respect to this pattern but *LHLL is not. This can partially be described with the banned substructure constraint in (1.5a) prohibiting a H mora followed by two L morae (and thus necessarily in pre-penultimate position). Whether or not a string is well-formed according to this constraint can be calculated by taking a window three morae long and scanning through the word, looking to see if it contains the banned HLL sequence, as shown in (1.5b) and (c) (the symbols $\rtimes$ and $\ltimes$ indicate left and right word boundaries, respectively).

(1.5) a. ¬HLL

b. $\boxed{\rtimes\ \mathrm{L}\ \mathrm{L}}\ \mathrm{H}\ \mathrm{L}\ \ltimes$ ✓ $\rtimes$LL
   $\rtimes\ \boxed{\mathrm{L}\ \mathrm{L}\ \mathrm{H}}\ \mathrm{L}\ \ltimes$ ✓ LLH
   $\rtimes\ \mathrm{L}\ \boxed{\mathrm{L}\ \mathrm{H}\ \mathrm{L}}\ \ltimes$ ✓ LHL
   $\rtimes\ \mathrm{L}\ \mathrm{L}\ \boxed{\mathrm{H}\ \mathrm{L}\ \ltimes}$ ✓ HL$\ltimes$

c. $\boxed{\rtimes\ \mathrm{L}\ \mathrm{H}}\ \mathrm{L}\ \mathrm{L}\ \ltimes$ ✓ $\rtimes$LH
   $\rtimes\ \boxed{\mathrm{L}\ \mathrm{H}\ \mathrm{L}}\ \mathrm{L}\ \ltimes$ ✓ LHL
   $\rtimes\ \mathrm{L}\ \boxed{\mathrm{H}\ \mathrm{L}\ \mathrm{L}}\ \ltimes$ ✖ HLL

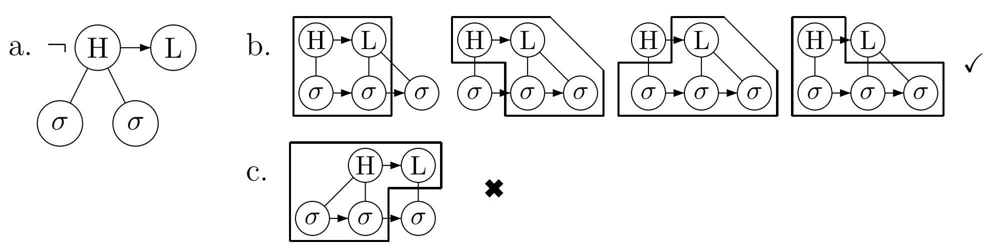
LLHL evaluates as well-formed because the banned HLL substructure is not found; *LHLL is ill-formed because scanning finds the HLL structure. Note that this is an extremely *simple* computation—the 'scanner' only considers what it currently sees in the box. It does not remember what it has seen, and so it cannot count or perform other calculations based on other parts of the string. It simply raises a flag when it sees a banned substructure, and judges the string well-formed when it does not.

This, then, is the *local* nature of banned substructure constraints: well-formedness is based on contiguous structures of a specific size. As pointed out by Rogers et al. (2013), this gives banned substructure constraints a straightforward cognitive interpretation, as well-formedness is determined entirely by the scanning operation illustrated in (1.5). It also allows for learning model for these constraints, as a learning algorithm only needs to scan through input data in this way in order to discover the pattern (García et al., 1990; Heinz, 2007, 2010a, 2011).

That these local constraints can be recruited for a restrictive, learnable theory of phonological well-formedness was first posited by Heinz (2007) and extended to phonological transformations by Chandlee (2014). However, it was immediately found that the notion of strict contiguity was too strong, as of phonology does indeed exhibit long-distance generalizations that refer to non-contiguous segments (some reviews can be found in Odden, 1994; Hansson, 2001, 2010). However, it is more restrictive to change—in a well-defined way—what we mean by ‘contiguous’ than to abandon this notion of locality, as shall be discussed in more detail in Chapters 3 and 8. Subsequent work showed how banned substructure constraints can also capture many long-distance generalizations by considering different substructures in strings (Heinz, 2007, 2010a; Heinz et al., 2011).

However, this is not quite sufficient for tone, as Chapter 3 will show examples of well-formedness patterns in tone which are beyond description by established banned substructure constraints in strings. This aligns with arguments from traditional phonology saying that tone requires some form of representation beyond strings (Goldsmith, 1976; Yip, 2002; Hyman, 2011b, 2014). The goal of this dissertation, then, is to show how this computational notion of locality can be extended to autosegmental representations to create a sufficient theory for well-formedness in tone. In this case, then ‘contiguity’ is defined over connected *subgraphs*, or substructures of autosegmental representations, but the idea of scanning for well-formedness is the same. How the well-formedness of autosegmental structures can be evaluated with respect to the constraint in (1.2) by scanning through contiguous *autosegmental* substructures is illustrated in (1.6).

(1.6)



As explained in Chapter 5 and in Chapter 6, these constraints are still restrictive thanks to the local evaluation of well-formedness. However, locality in autosegmental structures allows for describing long-distance patterns, because tone bearing units which appear non-contiguous in a string may be contiguous when considering their associations to tonal autosegments. Furthermore, because locality is not limited to strict adjacency, this obviates the need for underspecification in order to keep things local (although it reliant on adherence to the Obligatory Contour Principle). This is discussed in detail in Chapter 6. Finally, as described in Chapter 8, methods of learning banned substructure constraints in strings can be directly ported to banned substructure constraints in autosegmental representations, meaning that the language-specific constraints used in the analyses of various languages in this dissertation are all learnable.

1.3 Some Representational Assumptions

As alternative representational theories have been recently proposed (Cassimjee and Kisseberth, 2001; Hyman, 2014; Leben, 2006; Shih and Inkelas, 2014), before proceeding any further it is necessary to briefly defend this thesis' use of autosegmental representations. First, as a theory of representation, autosegmental representations are the best-studied in terms of their formal properties (Bird and Klein, 1990; Coleman and Local, 1991; Eisner, 1997a; Kornai, 1991, 1995; Wiebe, 1992; Jardine, 2014). More importantly, autosegmental representations continue to provide insight into phonological phenomena (Marlo, 2007; McCarthy, 2010a; Walker, 2014). As to be seen throughout this dissertation, autosegmental representations straightforwardly capture relations between tone bearing units (henceforth TBUs) that result in contours and 'plateaus' of TBUs that agree in tone through multiple association between phonological units. There are other arguments for autosegmental representations that are not clearly addressed by these alternatives, particularly with regards to tone, such as the 'stability', or independence of tonal units with respect to phonological or morphological processes that modify their host TBUs. The reader is referred to the substantial literature on this topic (ex., Goldsmith, 1976; Yip, 2002; Hyman, 2011b, 2014).

Finally, it is not clear that the alternatives are different from autosegmental representations in a substantial way. In Shih and Inkelas (2014)'s ABC+Q theory, for example, association is ostensibly dealt away with but is replaced with sub-TBUs (to capture contours) and correspondence relations across TBUs and sub-TBUs for agreement. It is thus extremely likely that the local autosegmental constraints on multiple association 'translate' into local constraints over the correspondence relations between

TBUs and sub-TBUs. This would preserve the generality of this dissertation's central result, although formally defining ABC+Q is beyond the purview of this dissertation.

One final representational assumption is that, as the focus on this dissertation is on 'surface' patterns, i.e. the output of some level (i.e., word or phrase) of the phonology, the constraints outlined here will largely deal with autosegmental representations for which tones are present on all TBUs. Autosegmental representations with gaps in association, such as in analyses using floating tones and underspecification, shall be largely left to future work, although the concepts in this dissertation apply just as easily to such representations, and they will be discussed where appropriate. In particular, Chapter 7 goes into some detail regarding how underlying representations with underspecification can be defined and then related to surface forms which are fully specified.

1.4 Outline of the Dissertation

The logical structure of the dissertation is as follows. It should be noted that, as Chapters 2 and 3 provide the empirical and formal background, respectively, and Chapter 4 defines autosegmental representations in terms of concatenation, the core results of the dissertation do not begin until Chapter 5 starts the analysis of the typology of tone patterns in terms of banned subgraph grammars. However, as described above, the level of formal detail in Chapters 3 and 4 leads to theoretical payoffs in the remainder of the dissertation regarding the explicitness of the grammars (Chapter 5), comparing their expressivity (Chapter 6), applying them to phonological transformations (Chapter 7) and learning (Chapter 8).

Chapter 2 provides a picture of cross-linguistic variation in surface well-formedness generalizations in tone patterns and then shows how language-specific variation in tone patterns has previously been accounted for using autosegmental representations. The tone patterns in question are divided into two categories: 'tone-mapping' patterns in which well-formedness generalization refers primarily to directional, quality-specific, or positional restrictions on association; and 'long-distance' patterns in which the correct generalization refers to unbounded stretches of TBUs. After reviewing the basic properties of autosegmental representations, the chapter then discusses the merits and drawbacks of two conceptions of grammars over autosegmental representations: rule-based grammars and optimization-based grammars. While rule-based grammars arguably require some arbitrary rules in order to account for independent behavior of tones, optimization-based grammars typically rely on Align constraints, which have been shown to be computationally undesirable, and are unable to capture all of the patterns in the typology without arguably ad-hoc constraints. As these previous analyses refer to global directionalityy, they both miss that the surface generalizations are fundamentally *local* in the sense advocated here.

Chapter 3 offers an alternative to the above conceptions of grammar based in formal language theory and mathematical logic. Chapter 3 motivates grammars as inviolable, language-specific banned substructure constraints, and argues that, at least for tone, these constraints must operate over autosegmental representations. This is based on Rogers et al. (2013)'s hierarchy of logical languages for expressing constraints over strings. With a logical language defined for a set of objects (e.g., the set of all possible strings over an inventory of symbols), we may write statements that describe a subset of this set (e.g., the set of well-formed strings of a particular language). As argued in their work and Heinz (2007, 2010a), the least expressive of these logical languages, which is restricted to banning substructures of strings, offers a very strong hypothesis for phonological well-formedness. This is because these constraints are evaluated with reference only to these particular substructures, not the entire representation. The chapter then shows, through the long-distance tone patterns introduced in the preceding chapter, that this hypothesis is not sufficient for tone, at least when applied to strings. Moving from strings to autosegmental representations is proposed as a solution to this issue, as it is demonstrated that it is more restrictive to enrich the structure rather than use more powerful logical grammars.

In response to the problem of expressivity raised in Chapter 3, **Chapter 4** presents a formally sound way of defining autosegmental representations using autosegmental *graphs*. Mathematical graphs are used in order to make explicit all of the information contained in autosegmental representation, and it is shown how the graph theoretic concept of *graph concatenation* can define autosegmental representations from a set of *graph primitives*. This concatenation operation is shown to guarantee that the No-Crossing Constraint (NCC) and the Obligatory Contour Principle (OCP) are preserved for the set of graphs if they are preserved in the set of graph primitives. Looking at APRs through this concatenation operation offers several advantages. One, it accurately captures the fact that unbounded spreading is attested in natural language, but unbounded 'contouring' is not. Two, it highlights the string-like nature of autosegmental representations, a connection between linear and autosegmental representations which is used later in the dissertation to directly compare graph and string grammars, build 'correspondence graphs' which represent phonological transformations, and provide a method for learning autosegmental grammars from string inputs. Additionally, some special empirical cases which may require refinement of the concatenation operation are also discussed.

Chapter 5 synthesizes the concepts from Chapters 3 and 4 to define a logical language of banned substructure constraints over autosegmental representations. As the previous chapter showed how to define these representations as graphs, the relevant substructures are now connected sub*graphs*. The chapter then illustrates how these banned substructure constraints can describe the range of 'tone-mapping' patterns first discussed in Chapter 2. It argues that these analyses compare favorably to the previous analyses discussed in that chapter, as they give a unifying analysis of directional, quality-specific, and positional association patterns (unlike derivational theories) and do not under- or overgenerate in the same way as optimization-based theories. This is because, unlike these previous theories, evaluation only occurs over connected subgraphs, and not with reference to the entire representation.

Chapter 6 then extends the subgraph-based analysis of the previous chapter to the long-distance tone patterns discussed in Chapters 2 and 3. This establishes that these patterns are within the range of banned subgraph grammars, which means that these grammars are more powerful than the local string grammars introduced in Chapter 3. In order to make this comparison, the chapter first establishes how the graph concatenation operation allows for direct comparison of string and autosegmental grammars. Additionally, the chapter shows that, while more expressive than local string grammars, banned subgraph constraints are not as expressive as the more powerful string logics discussed in Chapter 3. This further shows that banned subgraph grammars give a restrictive explanation of the typology of surface well-formedness in tone. It is also discussed how the describability of one pattern, Wan Japanese, depends on certain representational assumptions, but that these assumptions are reasonable.

Chapter 7 begins a theory of transformations over autosegmental structures based on graph constraints over derivations. It is shown how the idea of graph primitive concatenation introduced in Chapter 4 can be used to build a restrictive set of 'correspondence graphs' representing input/output structures out of primitives representing individual phoneme-phoneme correspondences. Banned subgraph constraints can then be used to specify phonological processes. This is first illustrated with strings, and then extended to autosegmental representations. Example patterns from Chapter 2, viewed as input/output mappings, are analyzed in this framework, showing that we can directly integrate local surface constraints into analyses of phonological transformations. This thus preserves the insight of OT that surface well-formedness drives phonological processes while also maintaining the fundamental local nature of the patterns identified by this dissertation.

Chapter 8 looks at two problems not yet addressed by the dissertation. The first is a kind of tone pattern that is not describable with banned subgraph constraints using the representations so far employed in the dissertation. Dubbed the 'superstructure problem', it is a general problem in which a well-formed structure with respect to a pattern is a superstructure of an ill-formed structure with regards to that pattern. Illustrated with contours in Aghem, this is an issue for banned substructure constraints, as it is impossible to ban the ill-formed structure without also banning the well-formed superstructure. However, it is shown how this problem can be solved by minimal additions to the representation which create a contrast such that the ill-formed structure is no longer a substructure of the well-formed structure. While this entails further representational primitives, it is discussed how these primitives have already been used (although implicitly) in other analyses in phonology.

The second problem is that of learning, although this problem is too easily approached. The second half of Chapter 8 defines learning as it relates to the classes of string patterns introduced in Chapter 3, and shows how a learning models for banned substructure constraints in strings can be extended to banned substructure constraints in graphs. It also shows how concatenation allows this learning model, with certain representational assumptions, to learn directly from strings. The result of this is a learning model for phonological patterns that did not previously have a learning model.

Chapter 9 concludes by reviewing the results of the dissertation and looking to future work, which includes a more comprehensive theory of correspondence transformations and their relation to other views of phonological transformations, potential representational differences between segmental and tonal phonology, and explicit definitions of other representations.

Chapter 2

SURFACE PATTERNS IN TONE

The focus of this chapter is to provide a picture of the nature of variation in surface tone patterns in natural language by giving empirical examples exhibiting two important aspects of this variation. The first aspect is that which can be characterized by differences in how tonal 'melodies' are realized over strings of tone-bearing units (TBUs); that is, syllables or morae. The second aspect, a well-known property of tone, is that language-specific tonal generalizations can refer to unbounded stretches of TBUs (Yip, 2002; Hyman, 2011b). This can result in 'long-distance' patterns, a notion which will be given a precise definition in the following chapter. It is thus important for any theory of tone to capture these two particular aspects of language-specific variation in tone, and that will be the primary empirical goal of this dissertation.

As a representational framework for describing these patterns, autosegmental phonological representations (APRs; Goldsmith, 1976; Coleman and Local, 1991), are introduced and briefly argued for. One of the primary insights of for APRs, which is that they can describe tonal patterns in terms of independent melodies realized over strings of TBUs, is demonstrated throughout this chapter. Other arguments for APRS exist; however, these draw from their insightful analysis of tonal *alternations* and thus less relevant to the types of patterns considered here.

The other primary function of this chapter is to give an overview of previous theoretical explanations of the range of tone patterns, in order to later compare them with this dissertation's analysis based on local substructure constraints. Both derivational frameworks, which comprise the association conventions and rules of traditional Autosegmental Phonology (Leben, 1973; Goldsmith, 1976; Leben, 1978; Hyman, 1987;

Hewitt and Prince, 1989; Archangeli and Pulleyblank, 1994), as well as the constraint-based framework of Optimality Theory (Prince and Smolensky, 1993, 2004), in particular the autosegmental treatment of tone in Optimal Tone Mapping Theory (Zoll, 2003), are discussed. The relative merits and drawbacks of each are discussed, but the primary drawback of both is that they fail to miss the *local* nature of the variation in phonological tone patterns, an aspect which is the focus of the analysis put forward by this dissertation.

The structure of this chapter is as follows. §2.1 introduces APRs through the case of Mende, providing details about the definition of APRs which will become important for their formal treatment in Chapter 4. §2.2 then reviews two important types of variation in tone patterns that have been traditionally described with autosegmental representations: tone-mapping patterns and long-distance patterns. §2.3 then discusses the merits and drawbacks of previous attempts at describing the empirical range of tone association patterns both in derivational frameworks and in Optimality Theory. §2.4 then concludes.

2.1 Autosegmental Representations

This section establishes autosegmental phonological representations (APRs) as the representational theory in which the description of variation in surface tone patterns are couched. This will allow, first in later sections of this chapter and then in later chapters in this dissertation, for the *explanation* of this variation through grammars which operate over these representation.

First, APRs are briefly motivated using an example from Mende in §2.1.1, focusing on the insight of the independence of tonal melodies from TBUs. Some universal well-formedness conditions on APRs are explicitly defined in §2.1.2.

2.1.1 Mende

One of the classic arguments for APRs comes from tone mapping in the Mande language Mende, spoken in Sierra Leone (Leben, 1973, 1978; Dwyer, 1978). Mende nouns separate into tone categories, examples of which are given in (2.1). Items (2.1a) through (c) show words whose syllables are all high-toned, items (2.1d) through (e) show words whose syllables are all low-toned, and items (2.1g) through (i) show words whose syllables start high and end low, (2.1j) through (l) show words whose syllables start low and end high, and (2.1m) through (o) show words whose syllables start low, go high, and then go back to low. In the following, [a̋̂] represents a rising-falling tone.

(2.1) Mende word tone (Leben, 1973, 1978)

a. kɔ́	'war'	b. pɛ́lɛ́	'house'	c. háwámá	'waist'
d. kpà	'debt'	e. bɛ̀lɛ̀	'pants'	f. kpàkàlì	'three-legged chair'
g. mbû	'owl'	h. ngílà	'dog'	i. félàmà	'junction'
j. mbǎ	'rice'	k. nìká	'cow'	l. ndàvúlá	'sling'
m. mbǎ̂	'companion'	n. nyàhâ	'woman'	o. nìkílì	'groundnut'

With rising contours as F sequences, falling contours as R, and falling-rising as R-F, the possible tone melodies in Mende are as follows:

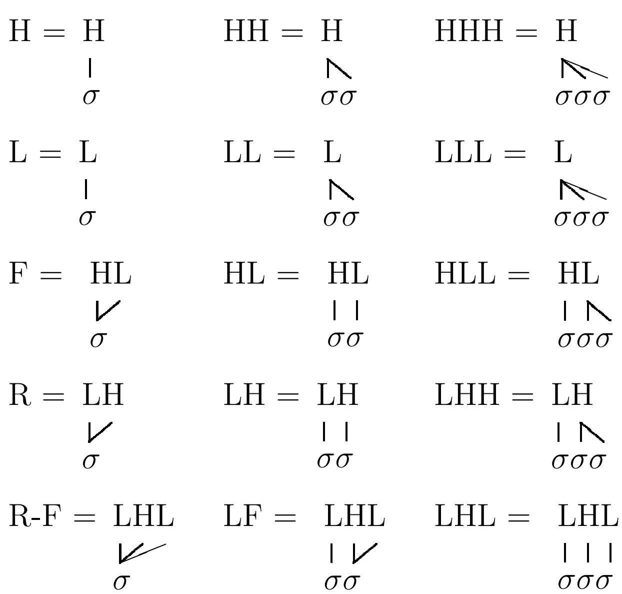
(2.2) Mende surface tone patterns

H	HH	HHH
L	LL	LLL
F	HL	HLL
R	LH	LHH
R-F	LF	LHL

Leben (1973) highlighted the fact that there far fewer attested surface tone patterns in Mende than the possible combinations of H, L, F, R, and R-F-toned syllables. For example, he claimed the absence of *HLH or *F-R (falling-rising contours). He also noted that (with the exception of LLH, which will be discussed later), tonal 'plateaus' of syllables with the same tone are limited to the right word edge (ex., HLL but *HHL), as are contours (ex. LF and R but *FL or *RL).[1] To explain these restrictions, he proposed that Mende words were underlyingly specified for one of five melodies: H, L, HL, LH, or LHL, and that the surface melodies were generated by rules for mapping these melodies to syllables left-to-right. The core insight is that the tone strings exist *independently* of the segmental strings. While his original analysis predated AP, his analysis found a straightforward interpretation in AP and as such was adapted into Goldsmith (1976)'s original presentation of the framework.

The following AP diagrams of the string patterns in (2.2) illustrate this. The H, L, HL, LH, LHL melodies are represented as strings of tonal units distinct from strings of syllables. Association lines then indicate which syllable each tone is realized.[2]

(2.3)



[1] It was later noted by Dwyer (1978) that Mende does allow, among other things, HLH sequences on words of three or more syllables. Dwyer (1978) also discusses the existence of LLH patterns. I will ignore these complications for now, in order to focus on the original justification for tone mapping rules. I consider additional Mende data in §2.2.1.3.

[2] As mentioned above, whether syllables or mora are the relevant tone-bearing unit can vary from language to language (or even within a language—see Kubozono (2012) for examples). As this dissertation is focused on the associations between tones and TBUs, the question of what is a valid TBU in a particular language is largely orthogonal to the main issues in the dissertation and shall not be discussed in any depth.

Note that plateaus are now represented as *multiple association* of a single tone to multiple syllables, as in (2.4a) below, and contours are represented as multiple association of multiple tones to a single syllable, as in (2.4b) below.

(2.4) a. LHH = $\begin{array}{l} \text{LH} \\ | \ \diagdown \\ \sigma\sigma\sigma \end{array}$

b. LF = $\begin{array}{l} \text{LHL} \\ | \ \diagup \\ \sigma\sigma \end{array}$

The generalization that plateaus and contours both only appear on the right edge of the word can now be recast as the generalization that only multiple association occurs on the right edge of the word. As mentioned previously, (Leben, 1973) and subsequent analyses in the derivational framework analyze this using left-to-right directional association, which will be illustrated in detail momentarily. However, right-edge-only multiple association is by no means universal, and so subsequent sections of this chapter will document the variation in tonal association patterns and then present previous theoretical attempts to explain this variation.

However, it is first necessary to explicitly state some universal constraints on APRs that all of these analyses share.

2.1.2 Autosegmental well-formedness

The preceding section informally discussed various assumptions about the structure of APRs. This section makes these assumptions, which I will call *well-formedness* conditions, explicit. These assumptions are important for the analyses of the language-specific variation in tone patterns given in the following sections. Also, making them explicit is necessary for providing a mathematically explicit definition of APRs and grammars over APRs, which will be undertaken in Chapters 4 and 5.

2.1.2.1 Tiers

A foundational well-formedness condition on APRs is that phonological units are arranged onto distinct tiers. For example, in the representation in (2.4b), repeated below in (2.5), there are two tiers, one containing tonal units and one containing timing tier units.

(2.5) LHL

$\sigma\sigma$

APRs are not limited to two tiers. Use of multiple tiers became quite commmon, especially in Feature Geometry-style theories of segmental phonology (Sagey, 1986; Clements and Hume, 1995). However, the basic rules of association, such as constraints against line-crossing (more on this below), are usually posted to only hold between pairs of tiers (called *charts* or *planes*; Coleman and Local, 1991; Goldsmith, 1976; Odden, 1994). As such, this dissertation will focus on two-tiered APRs.

It is often left implicit that like autosegments tend to be on the same tier, and tiers tend to house autosegments of the same type. For example, neither of the following is a well-formed APR in Mende:

(2.6) a. * L σ H

σ L

b. * HL

$\sigma\sigma$

L

The tiers in (2.6a) contain a mix of tone and timing units. In (2.6b), the tones have been split up into two tiers. No phonologist would ever posit an APR like (2.6a); in fact, it is so far out of the realm of possibility that such APRs are rarely ever explicitly banned. However, to define APRs mathematically, as is done in Chapter 4, we will need to do exactly that. It is true that mixing different types of units on a tier occurs in segmental *templatic* morphology and phonology (McCarthy, 1979, 1986, 1989). However, as Coleman and Local (1991) note, this dramatically increases the power of the APR formalism, as it allows circumvention of other well-formedness conditions, such as constraints on line-crossing (to be discussed momentarily). For the tone patterns discussed here, this kind of power is not necessary, and we can assume timing tier units appear on the timing tier and tone autosegments appear on the tonal tier.

APRs like (2.6b) are sometimes used in tone, but only in intermediate representations. One situation in which they are used is to separate L tones generated by ‘depressor’ consonants from tones originating from underlyingly specified ‘normal’ tones (Hyman, 2014). However, in order to be interpretable at the surface, the disparate tonal tiers must undergo a kind of *tier conflation* (McCarthy, 1986) into a single tonal tier. Let us then assume that on the *surface*, APRs like (2.6b) are banned, and tones must all appear on one tier.

2.1.2.2 The Obligatory Contour Principle

Another assumption about tiers at work in the Mende analysis in §2.1.1 concerns the content of the melody tier. For example, we saw that, assuming the underlying melodies L, H, HL, LH, and LHL, according to the generalization that multiple association only occurs on the right edge, the following APR (which corresponds to an unattested *HHL string of syllables) is ill-formed in Mende.

```
(2.7)  *  H  L
          |\  |
          σσσ
```

However, this assumption about the underlying melodies is a bit stipulative. Why do we not have a melody *HHL, which obtains a surface *HHL string of syllables while still adhering to the generalization that multiple association only occurs to the right edge?

```
(2.8)  *  HHL
          | | |
          σσσ
```

Leben (1973) posited that it was no accident that underlying melodies like *HHL, or *LLH, or *LHH, or *HLL, etc., were not necessary to analyze Mende. He proposed what Goldsmith (1976) named the Obligatory Contour Principle (OCP), which became a fundamental motivating constraint for generative phonology. The following formulation is due to McCarthy (1986).

(2.9) The Obligatory Contour Principle (McCarthy, 1986, p.208)

At the melodic level, adjacent identical elements are prohibited.

This automatically bans melodies such as HHL. The OCP has been the subject of much discussion in generative phonology. It was argued for as a constraint motivating a variety of processes in McCarthy (1986) and Yip (1988a). In an influential paper, Odden (1986) argued against it as a universal. Finally, some researchers in tone reinterpreted as a constraint referring to both the melodic level and the timing tier (Hewitt and Prince, 1989; Meyers, 1997). This version of the OCP is based on what Hewitt and Prince call *structural adjacency*. Let us briefly review how this version of the OCP is different from that of (2.9).

Essentially, melody autosegments A and B are structurally adjacent if they are adjacent on the tier and they are *not* associated to nonadjacent timing tier units. The relevant cases are schematized below in (2.10), taken from Hewitt and Prince (1989). Timing tier units are marked as ×.

(2.10) Structural adjacency (Hewitt and Prince, 1989, p. 178, (5))

```
a. Adjacent        A B      A B      A B      A B
                            |          |      |  |
                            × ×      × ×      × ×

b. Non-adjacent    A   B
                   |   |
                   × × ×
```

The variant of the OCP proposed in Hewitt and Prince (1989) is thus as follows:

(2.11) OCP (structural adjacency version). (Hewitt and Prince, 1989, p. 177, (3))

No melodic element may be structurally adjacent to an identical element.

It should be noted that the versions of the (2.9) and (2.11) only differ in one particular case. This is illustrated below in (2.12) with H tones associated to syllables.

(2.12)

	OCP (2.9)	Struc. Adj. OCP (2.11)
a. H H | | σ σ	*	*
b. H L H | | | σ σ σ		
c. H H | | σ σ σ	*	

According to definition (2.10a), adjacent H tones are structurally adjacent if only one of them is associated to a syllable, or if both are associated to adjacent syllables. In any of these cases, the structural adjacency version of the OCP agrees with the original tier-adjacency version of the OCP, as exemplified in (2.12a) above. Also, if a L intervenes between Hs, as in (2.12b), they are not tier adjacent and thus will violate neither version of the OCP. The difference only occurs when underspecification is allowed, as in (2.12c), which violates the original OCP but not the structural adjacency version.

Thus, the difference between the two versions of the OCP only applies in cases where underspecified TBUs are a possibility. The discussion in this chapter, as well as throughout most of this dissertation, will focus on surface APRs in which L tones are present, and thus the two versions are essentially the same.

It is also worth briefly discussing the universality of the OCP. In a famous paper, Odden (1986) argued against the OCP as a hard universal. However, his arguments focus on the OCP as a constraint on underlying, lexical forms. In fact, many of his examples of violations of the OCP only do so in the underlying forms. For example, he cites the following contrast in Zezuru between forms with two adjacent H tones and forms with a single, doubly associated H:

(2.13) Zezuru underlying contrasts Odden (1986, p. 367)

a. H H /denga/ 'sky' b. H /shato/ 'python'

This contrast can be argued for based on differing behavior with regards to tonal rules. However, the OCP violating form in (2.13a) does not *surface* as such, due to a lowering rule that applies to successive Hs. Thus, the surface form of (2.13a) in isolation is as in (2.14).

(2.14) H L
denga 'sky'

Thus, while the OCP may not be obeyed underlyingly in Zezuru, it is obeyed on the surface. Odden (1986) also gives as evidence of tone melodies violating the OCP a pattern in Karanga Shona; however, Hewitt and Prince (1989) give a competing analysis, explicated in §2.3.1.1 below, in which the OCP is not violated on the surface.

In fact, the only true examples of OCP violation listed in Odden (1986) are signaled by phonetic downstep. For example, Odden lists the contrasting APRs in (2.15) for two nouns in Kishambaa (Odden, 1986, Fig. 13):

(2.15) a. H nyoka 'snake' b. HH ngoto 'sheep'

This is partially motivated by the different surface pronunciation of the two forms: the first, (2.15a) 'snake' is pronounced with two level H tones, [nyóká], and (2.15b) 'sheep' is pronounced with a H followed by a downstepped H; [ngóꜜtó]. Odden (1986) reports similar facts for Temne.

Thus, when viewed as a constraint on the surface, the OCP appears extremely robust. In fact, Hyman (2014) states "[c]ases of tautomorphemic OCP violations are extremely rare" (p. 371). This is important when when viewing autosegmental representations as derived from concatenation, as will be seen in Chapter 4. That (at least in tone) these violations are marked by downstep (as in Kishambaa) is also particularly relevant. For now, the OCP will be treated as universally applying *on the surface*, and unless otherwise noted, ‘OCP’ shall refer to the constraint in (2.9) which only refers to the melody tier.

2.1.2.3 Association lines

The next notions of well-formedness deal with the associations between units on these tiers. Most versions of AP posit strict constraints on these associations. Note that, for Mende, none of the following are desirable as valid surface forms:

(2.16) a.* LHL b.* HL c.* HL

$\sigma\sigma$ $\sigma\sigma\sigma$ $\sigma\sigma\sigma$

In (2.16a) and (2.16b) have autosegments left not associated to anything. In (2.16c), the association pairing H with the third syllable crosses that pairing the L with the second syllable. To ban such structures, Goldsmith (1976) posited a Well-formedness Condition, given in (2.17) (generalized to TBUs instead of just vowels, to which it originally referred).

(2.17) The Well-formedness Condition (Goldsmith, 1976, (24))

a. All TBUs are associated with at least one tone;
All tones are associated with at least one TBU.

b. Association lines do not cross.

Goldsmith (1976) interpreted his Well-formedness Condition as a kind of ‘active’ constraint which, at each step of the derivation, filled in the minimum number of association lines until it was maximally satisfied.

It can also be considered a constraint on surface APRs. For example, both (2.16a) and (b) are invalid because they each violate (2.17a)—(2.16a) because it has an unassociated tone, and (2.16b) because it has an unassociated TBU. Subsequent work on APRs, such as Pulleyblank (1986)’s analysis connecting floating tones to downstep, has shown that (2.17a) is too strong. This will not be relevant for the tone patterns discussed in this dissertation, but Chapter 4 will return to this point.

The constraint in (2.17b), which later came to be known as the No Crossing Constraint (NCC; Hammond, 1988), is generally considered to be more universal. Informally, the APR in (2.16c) can be said to violate the NCC because the association lines between the H and the third syllable and the L and the second syllable intersect. Formally, defining the NCC has actually proven quite complicated; a full discussion will be taken up in Chapter 4. For now, it is enough to understand the informal definition.

2.1.2.4 Interim conclusion: Autosegmental representations

This section has established the basic aspects of APRs and autosegmental well-formedness that will be sufficient to discuss the tone patterns at the focus of this dissertation. The following discussion of these patterns will thus assume as representational theory APRs with a homogenous tier structure as discussed in §2.1.2.1, the OCP as defined using locality on the melody tier in (2.9), and both parts of the Well-formedness Condition in (2.17).

Of course, as was mentioned throughout this section, there have been arguments against the universality of some of these notions of well-formedness, in particular the OCP and part (a) of the Well-formedness Condition. These differences are largely orthogonal to the central goal of this dissertation, which is to determine the *computational* nature of the variation in tonal patterns. However, they will be addressed where appropriate (in particular, Chapter 4, §§4.5.2 and 4.5.3).

Finally, this section has not spent a great deal of time *arguing* for APRs, aside from the point in §2.1.1 that they insightfully capture the idea of melodies associating to strings of syllables. This is of course not the only evidence for APRs. One example is Pulleyblank (1986)'s use of floating tones, mentioned above, to explain tonal alternations in a number of languages. As tonal units with no anchor, floating tones embody the autosegmental idea of tonal units which exist independently of the phonological units over which they are realized. Such arguments are based on tonal *alternations* they are not particularly relevant to the focus of this dissertation, which is on static, surface tonal patterns. The reader is referred to Odden (2013) for the traditional arguments for APRs. Additionally, this dissertation presents additional, *computational* evidence for APRs, as the following chapters show how local constraints over APRs provide a better fit to the typology of tonal variation than local constraints over strings. First, the remainder of this chapter establishes what that variation is and how it has previously been treated in the literature.

2.2 Language-specific Variation in Association and Melodies

Having established some working representational assumptions, we can now turn to a typology of tone patterns to examine *language-specific* variation in autosegmental well-formedness. As outlined in the introduction, this variation will be examined in two rough categories. First is the variation in "tone-mapping" phenomena, or how melodies associate to strings of syllables (as in the Mende example above). The second are "long-distance" phenomena for which the correct generalization refers to TBUs separated by unbounded lengths. In autosegmental terms, these are often due to a restriction on what constitutes a valid melody, irrespective of the number of TBUs to which each melody unit is associated. While many types of distance phenomena have been noted in tone, the particular cases studied in the latter half of this section will be seen in the following chapter to be 'long-distance' in a precise, computational sense.

First, however, let us examine variation in tone-mapping phenomena.

2.2.1 Variation in melody association in tone-mapping

Recall from §2.1.1 above that the basic generalization in Mende was that melodies "mapped" onto TBUs such that, when viewed as autosegmental representations, multiple association only occurred on the right edge. The following will show that this consraint on autosegmental well-formedness is by no means universal. Citing data from Hausa (Newman, 1986, 2000), Kukuya (Archangeli and Pulleyblank, 1994; Hyman, 1987; Zoll, 2003), additional patterns in Mende (Dwyer, 1978; Leben, 1978), and Northern Karanga Shona (Hewitt and Prince, 1989; Meyers, 1987; Odden, 1986), the following shows that language-specific autosegmental well-formedness generalizations can be *directional* (i.e., refer to whether multiple association can occur on the right or left), *quality-specific* (i.e., specific to H or L tones), or *positional* (i.e., refer to TBUs at specific positions). Thus, any theory of tone must be able to capture these three kinds of variation.

2.2.1.1 Hausa

Is the Chadic language Hausa (Newman, 1986, 2000), spreading and contours occur on the left edge of the word, not the right.

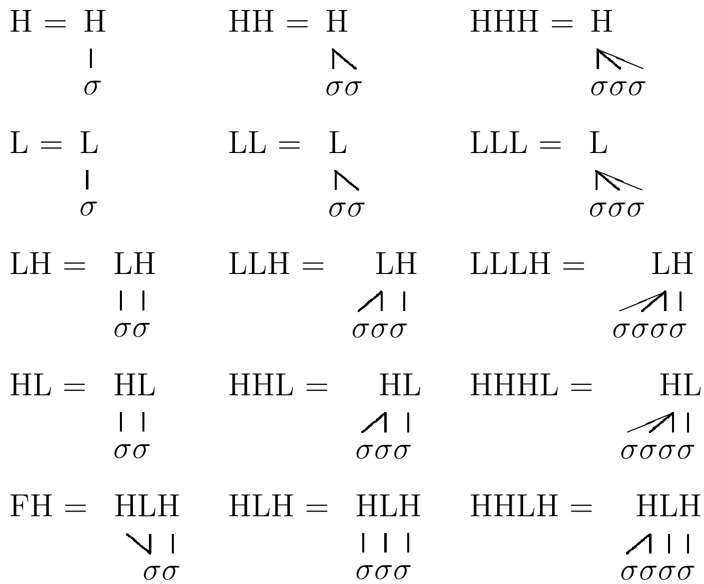
(2.18) Hausa word tone

a. jáa	'pull'	b. jíráa	'wait for'	c. béebíyáa	'deaf mute'
c. wàa	'who?'	d. màcè	'woman'	e. zàmfàrà	'Zamfara'
f. jàakíi	'donkey'	g. jìmìnúu	'ostriches'	h. bàbbàbbàkú	'well roasted'
i. fáadì	'fall'	j. hántúnàa	'noses'	k. búhúnhúnàa	'sacks'
l. mântá	'forget'	m. kákkáràntá	'read'	n. kákkáràntá	'reread'

(2.19) Hausa surface tone patterns

H	HH	HHH
L	LL	LLL
LH	LLH	LLLH
HL	HHL	HHHL
FH	HLH	HHLH

In this way, Hausa is the mirror image of Mende, with the exception that HLH is allowed, only falling contours are allowed, and monosyllables with contours are (for the most part) absent (Newman, 2000). For example, while three-syllable HL words were pronounced HLL in Mende, they are HHL in Hausa. In autosegmental terms, this means that multiple association in Hausa is allowed at the left, and not the right edge:

(2.20) 

Thus, languages can differ in directionality of association; i.e., whether multiple association can occur on the right or left edge (the use of the term 'directionality' will become more clear when the analysis of these mapping patterns using directional association is covered in §2.3.1.1).

As with Mende, there are additional complications to the Hausa data beyond the basic directionality. In particular, as pointed out by Zoll (2003), there are some morphologically simplex forms with surface LHH melodies, which (continuing to the observe the OCP) require left-to-right directionality. However, Newman (1986, 2000) provides plentiful evidence that basic association in Hausa occurs such that multiple association is only allowed on the left edge. This dissertation will thus abstract away from the monomorphemic exceptions raised by Zoll.

2.2.1.2 Kukuya

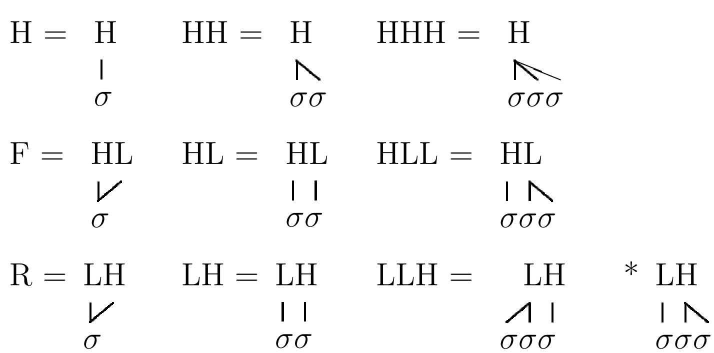
Kukuya (Bantu, Kongo; Archangeli and Pulleyblank, 1994; Hyman, 1987; Zoll, 2003). Kukuya shows a very similar distribution of tone patterns as Mende, with the exception that *LHH forms are not possible, while instead LLH forms are. The following table (from Zoll, 2003, p.229) summarizes the Kukuya patterns.

(2.21) Kukuya word tone patterns ([Zoll], [2003])

H		HH		HHH	
a. bá	‘palms’	b. bágá	‘show knives’	c. bálágá	‘fence’
F		HL		HLL	
d. kâ	‘to pick’	e. sámà	‘conversation’	f. káràgà	‘to be entangled’
R		LH		LLH	(*LHH)
g. sǎ	‘ knot’	h. kàrá	‘paralytic’	i. m^{w}àrə̀gí	‘younger brother’
R-F		LF		LHL	
j. bvǐ̀	‘falls’	k. pàlî	‘goes out’	l. kàlə́gì	‘turns around’

The generalization here, as pointed out by Zoll, is that there is a *quality*-specific restriction on association: L tones may freely multiply associate, but H multiply associates just in the case that the only tone in the word. Thus, (2.21c) bálágá ‘fence’ is licit when *HHL is not. This generalization is clearer when looking at the autosegmental diagrams for H-, LH- and HL-melody autosegmental representations of the above data:

(2.22)



Note that whereas multiple association in contours only occurs on the right edge (e.g., (2.21k) LF [pàlî] ‘goes out’), in terms of multiple association of tones, in cases where both L and H tones appear in the melody only L, and not H, can multiply associate. Thus, *LHH is illicit, but LLH is allowed. Addtionally, HHH is allowed because no other tones appear on the melody tier. Thus, Kukuya is an example of a quality-specific well-formedness condition that refers to specific tonal phonemes (this terminology is due to Zoll (2003)).

2.2.1.3 Mende (continued)

Another type of autosegmental well-formedness condition comes from Mende itself, in the form of additional data highlighted by Dwyer (1978) and then addressed by Leben (1978). These additional data show patterns which are noted by both authors as much lower lexical frequency than the 'canonical' patterns given in (2.1) and (2.2) in §2.1.1.[3] These data concern the realization of HL and LH melodies.[4] As the examples below show, Mende admits, for example, both HL (2.23a) and HF (2.24a) in bisyllabic forms, and both LHH (2.23d) and LLH in (2.24d) in trisyllabic forms. The 'canonical' HL and LH associations from (2.1) are repeated in (2.23) below, while the additional attested HL and LH associations are given in (2.24).

(2.23) HL and LH melody forms from §2.1

Melody		2σ			3σ	
HL		HL			HLL	
	a.	ngílà	'dog'	b.	félàmà	'junction'
LH		LH			LHH	
	c.	nìká	'cow'	d.	ndàvúlá	'sling'

(2.24) HL and LH melody forms from Dwyer (1978, p. 174, (7))

Melody		2σ			3σ	
HL		HF			HHL	
	a.	kɔ́nyô	'friend'	b.	séwúlò	'rodent'
LH		LR			LLH	
	c.	(not attested)		d.	lèlèmá	'mantis'

[3] Dwyer (1978) states that Leben's original analysis (i.e. the one covered in §2.1.1) "account[s] for at least 90% of the modern Mende morphemes and probably 98% of Proto Southwestern Mande" (p. 185).

[4] There are also additional complications with regards to downstep in Mende, which I can abstract away from without compromising my analysis of tone mapping. For some discussion on how downstep factors into the theories advanced in this dissertation, see Chapter 4, §4.5. For further details on the specifics of Mende downstep the reader is referred to Dwyer (1978) and Leben (1978).

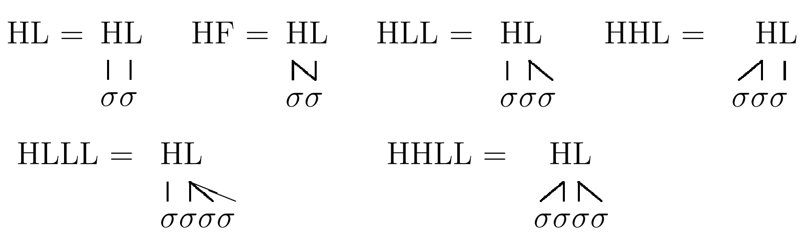
The full set of patterns is in (2.25) below, with the additional possible melody realizations highlighted in bold. This table is a summary of one given by Dwyer (1978, p. 169, (2)).

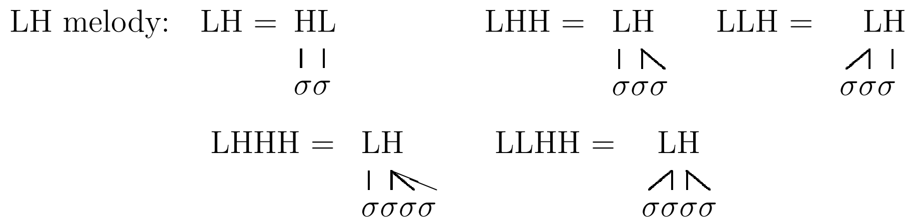
(2.25) Mende string patterns (Dwyer, 1978, p. 169, (2))

(H melody)	H,	HH,	HHH,	HHHH
(L melody)	L,	LL,	LLL,	LLLL
(HL melody)	F,	HL,	HLL,	HLLL,
		HF,	**HHL**,	**HHLL**
(LH melody)	R,	LH,	LHH,	LHHH,
			LLH,	**LLHH**
(LHL melody)	B,	LF	LHL,	LHLL

It should be noted that the new 4σ forms for the HL and LH melodies are HHLL and LLHH, respectively, and not HHHL and LLLH. Dwyer (1978) lists HHLL and LLHH as valid expressions of HL and LH melodies, respectively (p. 169, (2)), however he gives no example or comment on such forms. (Leben (1973, 1978) makes no mention of these forms.) Assuming that these are correct, the correct generalization for Mende is then that, while in general multiple association is usually restricted the right edge, an initial tone may either associate to the first syllable or both the first and second syllables, as long as this does not create a rising tone. The autosegmental representations for the full set of possible associations of HL and LH melodies in multisyllabic forms are thus are as follows:

(2.26) HL melody:





Note that LR, a two-syllable form in which an initial L tone associates to both syllables to create a contour (c.f. HF), is not attested on the surface. Dwyer (1978), among others, analyzes a class of surface LH words as underlyingly LR, based on morphophonological alternations. Leben (1973, 1978) argues against such an analysis, and shows that this distinction is much more clearly analyzed as an accentual one. Regardless, LR patterns are not seen on the *surface*, which is the focus of the current discussion.

The main lesson of this Mende data is that, because an initial, nonfinal tone can associate to one of the first two syllables but *not* to a third syllable—note that HF and HHL, but not *HHHL, are attested—language-specific autosegmental well-formedness can also refer to 'positional' information, or associations to particular syllables in a string. The next set of data will show this more dramatically.

2.2.1.4 N. Karanga Shona

The final set of language-specific constraints on association come from the Northern Karanga dialect of Shona (henceforth simply 'N. Karanga'; Hewitt and Prince, 1989; Meyers, 1987; Odden, 1986). N. Karanga has morphologically based tone alternations on the verb for which well-formedness refers to directionality, quality, and position. Hewitt and Prince focus on four tenses in N. Karanga, but for the current purposes it will be sufficient to examine just two, the Assertive and Non-assertive. Furthermore, N. Karanga verbs come either with or without an underlying H tone; I focus on toned verbs here.

The tone patterns for underlyingly toned verb roots in the Assertive and Non-assertive tense are summarized in (2.27) and (2.28) below. As do Hewitt and Prince, I focus on the tone pattern of the domain comprising the root and suffixes, ignoring the prefixes /ku/ and /handáka/, which comprise a different phonological domain. In the Assertive tense, a series of three H toned syllables is followed by an unbounded stretch of L toned syllables.

(2.27) Assertive

a. kù-$\underline{\text{p}}$-á	'to give'	H
b. kù-téng-á	'to buy'	HH
c. kù-téng-és-á	'to sell'	HHH
d. kù-téng-és-ér-à	'to sell to'	HHHL
e. kù-téng-és-ér-àn-à	'to sell to e.o.'	HHHLL
f. kù-téng-és-és-ér-àn-à	'to make take a lot for e.o.'	HHHLLL

The Non-assertive tense is very similar, but there is additionally a final H tone. In shorter words, the initial span of H tones is less than three syllables so that an L and, in words of three syllables or more, a final H are realized.

(2.28)

Non-assertive

a. hàndákà-p-á	'I didn't give'	H
b. hàndákà-tór-à	'I didn't take'	HL
c. hàndákà-tór-ès-á	'I didn't make take'	HLH
d. hàndákà-tór-és-èr-á	'I didn't make take for'	HHLH
e. hàndákà-tór-és-ér-àn-á	'I didn't make take for e.o.'	HHHLH
f. hàndákà-tór-és-ér-ès-àn-á	'I didn't make take a lot for e.o.'	HHHLLH
g. hàndákà-tór-és-ér-ès-ès-àn-á	(same as f.)	HHHLLLH

There are multiple generalizations here. First, in both tenses, the leftmost H associates maximally to the first three syllables, as exemplified by the APRs for 6σ forms in the Assertive and Non-assertive given below.

(2.29) Assertive / Non-assertive

HHHLLL= H L $\sigma\sigma\sigma\sigma\sigma\sigma$

HHHLLH= H L H $\sigma\sigma\sigma\sigma\sigma\sigma$

The second H of the NON-ASSERTIVE tense associates to the final syllable, and only to this syllable. The initial H will not spread onto the penultimate syllable (as if to preserve the structural version of the OCP), which remains L.

(2.30) NON-ASSERTIVE

```
HLH=  HLH      HHLH=  H L H      HHHLH=  H   L H
      | | |           |\ | |              |\\  | |
      σσσ             σσσσ               σσσσσ
```

Modulo this restriction, the initial H spreads as close as it can to the third syllable. In this way, it can also be called a 'positional' generalization, as it refers to a particular syllable in the word. Note also that there is also a directional and quality-specific component, as a second and word-final H cannot be multiply associated (while a final L can, in the ASSERTIVE tense).

2.2.1.5 Interim conclusion: well-formedness of association

In this section, we have seen that languages can vary with respect to association in three ways. In Mende and Hausa, we saw directional well-formedness generalizations governing *where* multiple association can occur, whereas in Kukuya we saw a quality-sensitive generalization govering *for which tone* multiple association can occur. Finally, in Mende and in N. Karanga Shona, we saw well-formedness generalizations referring to syllables at specific positions. Thus, a theory of tonal phonology needs to capture all of these types of well-formedness; previous attempts at capturing this variation shall be discussed below in §2.3. However, first, it is important to cover another important type of autosegmental well-formedness, namely how generalizations which refer to the well-formedness of melodies can lead to long-distance phenomena.

2.2.2 Well-formedness of melodies and long-distance phenomena

This section briefly reviews another important aspect of tone, which is that it often operates over long distances. This is not a new idea in phonology; for example, Yip (2002) and Hyman (2011b) point out a number of examples of how tonal units can "see" each other over large numbers of TBUs, which provides additional evidence for the independence of melody units from TBUs. This section surveys just a few cases of such long-distance patterns, and the following chapter will show exactly why these particular patterns are of interest: they are 'non-local' in terms of strings.

This section examines the surface pattern resulting from unbounded high-tone plateauing (Hyman, 2011b; Jardine, to appear; Kisseberth and Odden, 2003) as well as the accent patterns of Hirosaki Japanese (Haraguchi, 1977; Kobayashi, 1970) and Wan Japanese (Breteler, 2013; Kubozono, 2011a; Uwano, 2012).

2.2.2.1 Unbounded tone plateauing

An excellent example of tone operating at a distance is *unbounded tone plateauing* (UTP; Hyman, 2011b; Jardine, to appear; Kisseberth and Odden, 2003), a well-attested tonal process that Hyman (2011b) cites as a striking example of the distance over which tones can operate. In UTP, all underlying H tones in a domain merge to form a plateau, no matter the distance between the TBUs to which they are specified. While UTP is a process, and the following discussion will motivate it as such, the real interest is in the resulting surface generalization, which is that any domain can contain only one, connected plateau of H-toned TBUs.

While UTP is attested in many languages (Hyman, 2011b; Jardine, to appear), Luganda (Hyman, 2011b; Hyman and Katamba, 2010) exhibits a particularly striking example. Nouns in Luganda can be either underlyingly unspecified for a tone or specified for one H tone, which may appear in different positions in the word. The former are pronounced as all L, the latter pronounced with a H tone on the specified TBU in isolation and in phrases with morphemes unspecified for tones. The following abstracts away from L tones inserted at intermediate or late stages in the as they do not bear on the realization of plateauing.

(2.31) Luganda toneless forms (Hyman and Katamba, 2010)

No H tone

a.	/ki-tabo/	kitabo	LLL	'book'
b.	/mu-tund-a/	mutunda	LLL	'seller'

(2.32) Luganda forms with one tone (Hyman and Katamba, 2010)

a.	/ki-kópo/	kikópo	LHL	'cup'
b.	/ki-sikí/	kisikí	LLH	'log'
c.	/mu-tém-a/	mutéma	LHL	'chopper'
d.	/byaa-walúsiimbi/	byaa-walúsiimbi	LLLHLLL	'of Walusimbi'
e.	/mu-tund-a+bi-kópo/	mutunda+bikópo	LLLLHL	'cup seller'

However, in forms with two underlying H-specified TBUs, the TBUs in between are all pronounced H.

(2.33) Luganda forms with two tones

a.	/mu-tém-a+bi-sikí/	mutémá+bísíkí LHHHHH	'log chopper'
b.	/bikópo byaa-walúsiimbi/	bikópó byáá-wálúsiimbi LHHHHHHHLLL	'the cups of Walusimbi'

That this plateauing process is truly *unbounded* can be seen by its application in larger phrases. Under certain morpho-syntactic conditions, noun-verb sequences can also form a domain for UTP. In (2.34c) below, /walúsimbi/ 'Walusimbi (proper name)' is an adjunct, and thus does not form a phonological phrase with the verb (adjacent Hs in the verb are deleted by a version of Meussen's Rule). In (2.34d) and (e), however, the two words do form a phonological phrase, and plateauing occurs accross all TBUs in between the underlying Hs.

(2.34) Luganda verb+noun combinations (Hyman and Katamba, 2010, (14))

a.	/tw-áa-mú-láb-a walúsimbi/	tw-áá-mu-lab-a walúsimbi
	'we saw him, Walusimbi'	HHLLL LHLL
b.	/tw-áa-láb-w-a walúsimbi/	tw-áá-láb-wá wálúsimbi
	'we were seen by Walusimbi'	HHHHHHLL
c.	/tw-áa-láb-a byaa=walúsimbi/	tw-áá-láb-á byáá-wálúsimbi
	'we saw those of Walusimbi'	HHHHHHHHLL

The result of UTP is that the relevant domain can only have one plateau of H toned TBUs. In other words, domains with discontinuous H-toned TBUs are banned, regardless of the distance between them.

(2.35) *HLH, *LHLLLH, *HLLLLLHLLHL, . . .

This surface generalization derived from UTP is thus long-distance because it refers to TBUs which can be separated by any distance.

In autosegmental terms, the generalization is that the melody tier can at most contain one H tone. The following illustrates this by contrasting the APRs of nouns in (2.32) in isolation with those of their compound. (originally given in (2.33)). In Luganda the TBU is not the syllable, as in the preceding languages, but the mora (Hyman and Katamba, 2010), and so the following APRs will be so labeled (although this is not a significant difference for the purposes of the present discussion).

(2.36)

a.	'chopper'	mutéma	LHL	LHL $\mu\mu\mu$
b.	'log'	kisikí	LLH	LH $\mu\mu\mu$
c.	'log chopper'	mutémá+bísíkí	LHHHHH	L H $\mu\mu\mu\mu\mu\mu$
d.	" "	*mutéma+bisikí	*LHLLLH	* LH L H $\mu\mu\mu\mu\mu\mu$

To form the compound, the simple concatenation of the two nouns in (2.36d) is ungrammatical because it contains two distinct H tones on the melody tier. In contrast, the attested compound, (2.36c), contains only one. Thus, in the case of the surface pattern of UTP, when viewed in terms of autosegmental representations, the long-distance generalization reduces to a statement that melody tier can contain at most one H.

2.2.2.2 Hirosaki Japanese

A very similar example can be found in Hirosaki Japanese (Haraguchi, 1977).[5] In Hirosaki Japanese, there must be *exactly* one H tone on the melody tier—in contrast to Luganda, all-L words are not attested (c.f. the toneless Luganda words in (2.31)). The following data from Haraguchi (1977) exemplify this pattern. The relevant TBU for Hirosaki Japanese is also the mora, including the moraic coda nasals, which can carry tone (e.g. in (2.37i) [tòráǹkù] 'trunk'). The relevant domain comprises both nouns and certain suffixes.[6]

[5] Kobayashi (1970) cites a variety called Hirosaki Japanese for which the intonational facts are quite different, involving spreading of the H tone. While the data cited there warrant further study, this dissertation focuses on the data from Haraguchi (1977), who cites as source Konoshima (1961).

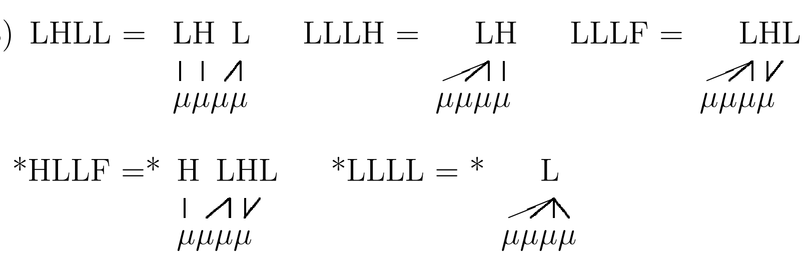
[6] For both Hirosaki Japanese and Wan Japanese, to be discussed below, this domain will be referred to as the 'word' although it includes many suffixes and compounds. The term for this phrase in Japanese linguistics is *bunsetsu* (see, e.g., Uwano, 1999), which does not map readily to either the English 'word' or 'phrase'.

(2.37) Hirosaki Japanese (Haraguchi, 1977, pp. 76–7)

Noun	Isolation	+NOM	Noun	Isolation
a. 'handle'	é	è-gá	f. 'chicken'	nìwàtòrí
	H	LH		LLLH
b. 'picture'	ê	é-gà	g. 'lightening'	kàmìnàrî
	F	HL		LLLF
c. 'candy'	àmé	àmè-gá	h. 'fruit'	kùdàmónò
	LH	LLH		LLHL
d. 'rain'	àmê	àmé-gà	i. 'trunk'	tòráǹkù
	LF	LHL		LHLL
e. 'autumn'	ákì	ákì-gà	j. 'bat'	kóòmòrì
	HL	HLL		HLLL

There are two major restrictions here. One is that there must be exactly one H or F-toned mora in the word—if there is a F-toned mora, then there cannot be an H, and vice-versa. Additionally, F-toned mora can only appear word-finally; e.g., LHLL is attested in (2.37i) [tòráǹkù] 'trunk' and LLLF is attested in (2.37g) [kàmìnàrî] 'lightening' but *LFLL is not attested.

These restrictions become much more clear when looked at as APRs. If we view F-toned mora as mora multiply associated to an H and L tone, then the restriction on H- and F-toned mora reduces to the statement that there must be exactly one H tone on the melody tier.

(2.38)


There is also a quality-sensitive restriction on multiple association: H cannot associate to multiple morae. That the falling contour is restricted to the right word edge can also be seen as a directional restriction of that only word-final morae can associate to multiple tones.

(2.39) *LHHL = * $\begin{array}{c}\text{LH L}\\ \mu\mu\mu\mu\end{array}$

*LFLL = * $\begin{array}{c}\text{LH L}\\ \mu\mu\mu\mu\end{array}$

Again, modulo spreading of the H, this is very similar to the UTP generalization in that there can only be one H tone in the melody. Hirosaki Japanese is further interesting for its additional requirement that there *must* be a H tone somewhere in the word. Like UTP, it is these two requirements that make this tone pattern 'long-distance', as the requirement holds over the entire length of the word. The following chapter (in §§3.5.1 and 3.6.1) will demonstrate in a precise way how this makes the pattern long-distance. However, it is important to remember that, in terms of the melody tier in the APR, these 'long-distance' generalizations can be reduced to a statement about the well-formedness of the melody tier.

2.2.2.3 Wan Japanese

The final long-distance pattern considered in this dissertation is from Wan Japanese, a Ryukyuan dialect spoken on Kikaijima island (Uwano, 2012; Kubozono, 2011b; Breteler, 2013). Like many other Japanese dialects, Wan Japanese is a "two-pattern" system in which words choose (in no phonologically predictable way) from one of two melody and association patterns. Following Kubozono (2011b) and others, I will refer to these as Type α and Type β. While type α has a rather simple directional association pattern unlike the one discussed above for Hausa, association to Type β is complex and sensitive to morphological information in a way that it can be classified as "long-distance".

Thus, while the following discussion will focus on Type β words in Wan, it will be useful to first review Type α. Regardless of the morphological composition of the word, Type α words are in general high with a low tone on the penultimate mora. The following data are from Breteler (2013, p. 46, Table 4.18). The relevant TBU is again the mora.

(2.40) Wan Type α

Noun	Isolation	+Nom	'from ∼'	'from ∼ also'
a. 'mosquito'	ká	kà-ŋá	ká-kàrá	ká-káràmú
	H	LH	HLH	HHLH
b. 'water'	mìdú	mídù-ŋá	mídú-kàrá	mídú-káràmú
	LH	HLH	HHLH	HHHLH
c. 'tatami mat'	tátàmí	tátámì-ŋá	tátámí-kàrá	tátámí-káràmú
	HLH	HHLH	HHHLH	HHHHLH

In autosegmental terms, this can be described simply as maximally a HLH melody with multiple association of tones to TBUs only allowed on the right edge (and no multiple association of TBUs to tones allowed anywhere).

(2.41) HLH = HLH $\mu\mu\mu$ HHLH = HLH $\mu\mu\mu\mu$ HHHLH= H L H $\mu\mu\mu\mu\mu$

The Type β is more complex, as the association of tones to is sensitive to the morphological information in the word. First, nouns in isolation follow a similar pattern to Type α, except there are low tones on the final and antepenultimate morae in phrase final position. In phrase-medial position, the final mora is high instead of low.

(2.42) Wan Type β nouns in isolation (Breteler, 2013, p6. 46, Table 4.18)

		Phrase-final	Phrase-medial
a. 'pot'	nábì	HL	HH
b. 'knife'	hàtánà	LHL	LHH
c. 'glutinous rice'	múcjìgúmì	HLHL	HLHH
d. 'sweet potato field'	háńsúúbàtéè	HHHHLHL	HHHHLHH

In autosegmental terms, the phrase-final pattern is simply the realization of a HLHL melody in which, as in Type α, only the initial H can be multiply associated. The phrase-medial pattern is almost the same, except it has a HLH melody in which the second H must associate to the final two morae. Thus, maximally, nouns in isolation follow one of the following two patterns:

(2.43) Phrase medial: $\mathrm{H}^n\mathrm{LHL} =$

$$\begin{array}{c} \mathrm{HLHL} \\ \mu \underbrace{\ldots}_{n} \mu\mu\mu\mu \end{array}$$

Phrase final: $\mathrm{H}^n\mathrm{LHH} =$

$$\begin{array}{c} \mathrm{HLH} \\ \mu \underbrace{\ldots}_{n} \mu\mu\mu\mu \end{array}$$

However, there is an additional complication when taking suffixes into consideration, and it is this complication that is the focus of the discussion and analysis in this dissertation (as it has been in previous analyses). The following discussion focuses on the phrase-final version of the pattern, which maintains the HLHL pattern.[7] When suffixes are attached, the second L remains word-final, whereas the first L remains in the antepenultimate position in the *stem*. As the sub-domain beyond the stem is relevant here, the boundary is marked with a '-' below.

(2.44) Wan Type β

Noun	Isolation	+Nom	'from ∼'	'from ∼ also'
a. 'pot'	nábì	nábí-gà	nábí-kárà	nábí-kárámù
	HL	HH-L	HH-HL	HH-HHL
b. 'knife'	hàtáná	hàtáná-gà	hàtáná-kárà	hàtáná-kárá-mù
	LHH	LHH-L	LHH-HL	LHH-HHL
c. 'gluti-	múcjìgúmì	múcjìgúmí-gà	múcjìgúmí-kárà	múcjìgúmí-kárá-mù
nous rice'	HLHL	HLHH-L	HLHH-HHL	HLHH-HHHL

---

[7] The phrase-medial version, as with the nouns in isolation, is HLH.

In terms of strings, the noun stem follows a maximally H^nLHH pattern, where H^n is some span of Hs, whereas the suffixes follow a H^mL pattern, where H^m represents all the mora in between the '-' suffix domain boundary and the last mora in the word.

(2.45) $\mathrm{H}^n\mathrm{LHH}\text{-}\mathrm{H}^m\mathrm{L}$

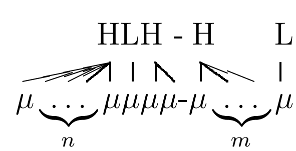
This may not seem like a long-distance generalization at first blush, but note that a final HHL sequence of morae is only allowed when a '-' boundary is present—contrast this with the patterns in (2.44) with the phrase-final forms in (2.42), which necessarily had a LHL pattern on the final three morae. In other words, the realization of the final three mora in the Type β pattern depend on the presence or absence of a '-' boundary. However, the '-' may be in principle removed from the final three morae by any number of number of H-toned mora, if we assume that the H^mL generalization for the suffix domain to be correct. Thus, we can term it long-distance. The following chapter will discuss this in more precise terms.

There are thus two possible autosegmental interpretations of the suffixed forms. One is that the generalization is that a HLHL melody associates such that the second H associates to the final mora in the stem and all but the final mora in the suffixes.

(2.46) $\mathrm{H}^n\mathrm{LHH}\text{-}\mathrm{H}^m\mathrm{L} =$

The other interpretation is that the stem takes its phrase-medial HLH melody and association paradigm, whereas the suffix domain takes a separate, HL melody. The generalization with regards to multiple association is then that in either domain, only a domain-initial H can associate to more than two morae. The two melodies can be separated by an additional '-' boundary on the melody tier (which can be interpreted as avoiding a violation of the OCP—see Chapter 4, §4.5.2).

(2.47) $\mathrm{H}^n\mathrm{LHH}\text{-}\mathrm{H}^m\mathrm{L} =$



Previous analyses have chosen to interpret the facts in terms of the first generalization, and so assume a represenation such as in (2.46). One argument in favor of such a representation is that the melody is HLHL for both the isolation forms and the suffixed forms. However, what this representation misses that (2.47) does not is that suffixed nouns take the same tone HLH tone meldoy as isolation form nouns in phrase-medial position (c.f. (2.42) and (2.43)). Thus, the first melody of (2.47) can simply be interpreted as a normal phrase-medial noun intonation. A second argument against (2.47) may be that, in terms of phonetics, the two adjacent Hs are pronounced as a single, uninterrupted H tone.[8] This argument does not hold water, however, as unpronouncable morpheme boundaries are posited regularly in analyses of segmental phonology, without phonetic 'breaks' which mark their presence. Thus, neither of the potential arguments against the second interpretation are particularly convincing. In fact, an additional argument *for* a representation such as in (2.47), rather than (2.46), comes from Chapter 6's analysis of this pattern, which shows that only the former allows for a description of the pattern with local constraints over autosegmental representations (thus making it like the rest of the patterns introduced in this chapter).

Regardless, whether (2.46) or (2.47) is correct, in Wan Japanese we see a pattern which can be considered long-distance at the string level which is not necessarily reducible to generalizations about the melody. Thus, any theory of tone must be able to capture this kind of long-distance behavior as well.

2.2.3 Interim conclusion: tone patterns and autosegmental well-formedness

This section has reviewed two important aspects of language-specific variation in tone patterns. The first are tone-mapping generalizations which refer to restrictions on how melodies are associated to strings of syllables. Languages may have directional association generalizations which restrict multiple association to the right edge, as in

[8] This argument would be somewhat speculative, however, as to the best of my knowledge no detailed phonetic work has yet been done on Wan Japanese.

Mende (with some exceptions), or to the left edge, as in Hausa. There are also quality-sensitive patterns which treat association differently for specific tones, as in Kukuya, for which H is only allowed to multiply associate when no L is present. Finally, there are positional generalizations, such as in Mende and in N. Karanga Shona, which refer to association to a syllable at a specific position.

The second aspect of language-specific variation in tone is that it includes long-distance generalizations which necessarily refer to units on the timing tier that are unboundedly far away. UTP, in which no two TBUs can be associated to distinct H tones, was one such generalization. A similar generalization was seen in Hirosaki Japanese accent, in which no two TBUs can be associated to a H tone. It was shown that, when viewed autosegmentally, both of these generalizations can be viewed with a restriction that the melody can only contain one H tone. Finally, Wan Japanese Type β words showed a generalization which depended on the presence or absence of a morpheme boundary, which resulted in a long-distance restriction on association that was not reducable to a generalization about well-formed melodies.

The purpose of this dissertation is to show how this variation can be characterized by a computational notion of locality over APRs. For the sake of theory comparison, the remainder of the chapter reviews previous attempts to characterize this variation, in order to ultimately show how the theory advanced in this disseration is superior to these characterizations.

2.3 Previous Explanations of Variation in Tone Patterns

The tone patterns described in the previous section have been analyzed previously in what, broadly speaking, can be characterized as either *derivational* or *optimization-based*. Derivational autosegmental analyses, starting with (Goldsmith, 1976), modify autosegmental structures step-by-step through a system of rules and constraints. Optimization-based grammars, applied to tone by a number of authors (Meyers, 1997; Yip, 2002; Zoll, 2003, inter alia), evaluate autosegmental structures with respect to a number of violable, universal constraints in a single step (or, as with

Stratal OT, two steps). The first part of this chapter reviews the relative merits and drawbacks of each with respect to the explanation of tone-mapping phenomena. The remainder of this chapter then does the same for long-distance patterns.

2.3.1 Analyses of association in tone-mapping

The following reviews both derivational and optimality-based explanations of the typology of association generalizations discussed in §2.2.1. Briefly, derivational theories deal with the edge-based directional association patterns by positing association paradigms which assign tones to TBUs one-by-one, either right-to-left (Mende and Kukuya), left-to-right (Hausa), or from both edges inward (N. Karanga). Quality-specific generalizations, like that in Kukuya, must then be dealt with language-specific rules applying after this association paradigm. Finally, positional generalizations are either accounted for with rules (N. Karanga) or accentual analyses utilizing underlying associations to specific syllables (Mende). While the derivational theories are descriptively adequate in that they can describe the full range of patterns, there is a preponderence of language specific rules and association paradigms, and thus the typological predictions of a derivational theory of tone mapping are unclear.

Within the framework of Optimality Theory (Prince and Smolensky, 1993, 2004), the predicted typology is derived from the possible rankings of a fixed set of universal constraints. Zoll (2003)'s theory of Optimal Tone Mapping constraints provides an analysis of the above typology, focusing on the explanation of quality-specific tone association patterns through Clash and Lapse constraints militating against spreading of a H and L tone, respectively, to adjacent TBUs. A strength of this analysis is that, through the emphasis on these Markedness surface constraints, it more insighfully captures quality-specific behavior, whereas this must be accounted for with somewhat arbitrary rules in a derivational framework. As also shown below, directional generalizations can captured with Align constraints (McCarthy and Prince, 1993), although there are two major issues with these constraints. First, they are calculated using *global* evaluation, which has been shown to be powerful enough to make bad typological predictions. More importantly, they cannot capture positional generalizations such as that in N. Karanga. One attempt to fix the N. Karanga problem using metrical structure is reviewed (Topintzi, 2003), but it is shown how it requires an analysis-specific constraints which provides no theoretical improvement over derivational analyses which posit language-specific rules.

Finally, while both approaches have their respective merits and drawbacks, both miss the generalization that the language-specific restrictions on association in tone-mapping patterns are *local* in the sense that they can be described by constraints specifying the well-formedness of substructures of a particular size, a notion that will be described explicitly in the following chapter.

2.3.1.1 Tone mapping with rules and parameters

Let us begin with the derivational framework. From early on, one of the keys to autosegmental explanations of tone patterns was directional association, which gives an insighful analysis of both "directional" patterns. For example, a surface HLL form in Mende, such as (2.1i) [félàmà] 'junction', is derived from the underlying form is thus as in (2.49a), with the HL melody distinct from the string of syllables over which it will be manifested. Williams (1971) and Leben (1973), while predating the autosegmental framework, specifically posit left-to-right tone mapping rules, which were adopted by Goldsmith (1976) for "default" (i.e., accentless) association, and other work. An explicitly autosegmental definition of left-to-right association can be found in Pulleyblank (1986)'s Association Convention:

(2.48) Association Conventions: (Pulleyblank, 1986, p.11, (14))
Map a sequence of tones onto a sequence of tone-bearing units,

a. from left to right
b. in a one-to-one relation

Through (2.48), associations between the melody string and the syllables then proceed one-to-one, left-to-right, which for an HL melody realized over three syllables results in (2.49b), which would be pronounced HLL.

(2.49) a.. HL → HL → HL → b.. HL

$\sigma\sigma\sigma$ $\sigma\sigma\sigma$ $\sigma\sigma\sigma$ $\sigma\sigma\sigma$

In left-to-right association, unassociatied melody autosegments associate in a one-to-one manner to unassociated timing tier autosegments, starting with the leftmost melody autosegment and leftmost timing tier autosegment. This initial pair associates first, and then the process repeats for each successive melody autosegment and timing tier autosegment, if there are any.

Due to this last step, left-to-right tone mapping correctly predicts that a plateau of a tone, represented here as autosegmental *spreading* of the last tone autosegment to any remaining timing tier syllables, will only occur at the right edge of the word. As such, this tone mapping cannot generated the unattested *HHL pattern, which corresponds to the following AP diagram:

(2.50) * H L

$\sigma\sigma\sigma$

This analysis thus makes the prediction that contours only occur at the right edge of the word, as tone/timing tier autosegment pairs at the left edge associate one-to-one first. As noted above, this prediction is correct for Mende, as contours only occur on the final syllable. For example, a low-falling pattern is analyzed naturally in AP as the manifestation of a LHL melody over two syllables, as in (2.51a) below.

(2.51) a. LHL → LHL → LHL → LHL b. * LHL

$\sigma\sigma$ $\sigma\sigma$ $\sigma\sigma$ $\sigma\sigma$ $\sigma\sigma$

The unattested form in (2.51b), with a nonfinal contour, is thus correctly predicted to not exist. Thus, left-to-right autosegmental tone mapping rules explain the basic surface tonal patterns in Mende.

However, a left-to-right autosegmental tone mapping process predicts the wrong forms for Hausa, for which multiple association appears on the left edge. Newman (1986) shows how instead a right-to-left association paradigm in Hausa generates the correct surface patterns. In right-to-left association, pairs of autosegments associate starting with the *last* autosegments on each tier, as in (2.52) below, which has the same HL/3-σ UR as (2.49).

(2.52)
$$\begin{array}{ccccccc} \text{HL} & \to & \text{HL} & \to & \text{HL} & \to & \text{HL} \\ \sigma\sigma\sigma & & \sigma\sigma\sigma & & \sigma\sigma\sigma & & \sigma\sigma\sigma \end{array}$$

Because right-to-left tone mapping operating in the reverse of the left-to-right process described above, in cases like (2.52) when there are more timing tier autosegments than tone autosegments, it is the initial timing tier autosegments that are 'left out' of the one-to-one association, and are thus associated to the first tone. This predicts word-initial tone plateaus, which is correct in the case of Hausa. For example, (2.52) obtains the tone pattern of (2.18j) [hántúnàa] 'roses'.

The other prediction of right-to-left tone mapping is that contours should appear word-initially as well. This is also correct for Hausa. If we allow for an underlying HLH melody in Hausa, right-to-left tone mapping predicts the existence of both 4-σ words with an HHLH pattern and 2-σ words with an initial falling FH pattern, as per the derivations below in (2.53).

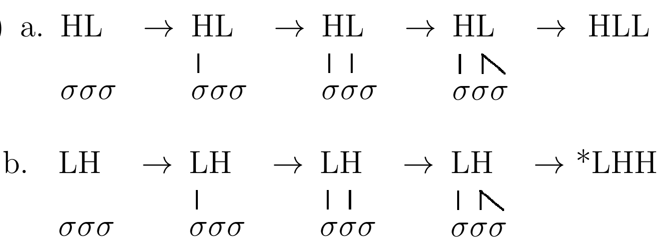
(2.53) a.
$$\begin{array}{ccccccccc} \text{HLH} & \to & \text{HLH} & \to & \text{HLH} & \to & \text{HLH} & \to & \text{HLH} \\ \sigma\sigma\sigma\sigma & & \sigma\sigma\sigma\sigma & & \sigma\sigma\sigma\sigma & & \sigma\sigma\sigma\sigma & & \sigma\sigma\sigma\sigma \end{array}$$

b.
$$\begin{array}{ccccccc} \text{HLH} & \to & \text{HLH} & \to & \text{HLH} & \to & \text{HLH} \\ \sigma\sigma & & \sigma\sigma & & \sigma\sigma & & \sigma\sigma \end{array}$$

This prediction is correct, as both patterns are attested, namely in (2.18k) [kákkáràntá] 'reread' and (2.18l) [mântá] 'forget'.

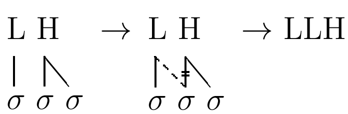
The Hausa data thus show that left-to-right tone mappings are not universal, and that to capture "directionality"-based patterns the direction of association in a derivational framework must be somehow parameterized. Archangeli and Pulleyblank (1994) offer one formalization of this, in that directionality is included as an explicit parameter on each rule.

However, parameterized directionality not capture "quality-specific" association patterns like that of Kukuya. Recall that with the exception of trisyllabic LLH, the tone patterns in Kukuya can be given the same left-to-right tone mapping analysis as Mende. That this analysis gives the correct surface pattern for HLL forms, but not LLH patterns, is shown in (2.54) below.

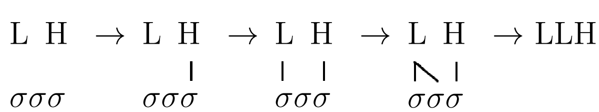
(2.54)


Hyman (1987) and Archangeli and Pulleyblank (1994) achieve this effect through the imposition of additional association rules. Hyman (1987) posits a L-Spreading rule that undoes the leftmost association of a doubly associated H:

(2.55) Kukuya L-Spreading (Hyman, 1987, p. 316, (7))



As Zoll (2003) correctly points out, this is an *ad hoc* solution, as the environment of L-spreading is exactly the problematic tone pattern. Archangeli and Pulleyblank (1994) instead stipulate there is a right-to-left Final H Association rule that takes priority over general left-to-right association. The effect is that a melody-final H first associates to the last syllable in the word, leaving the preceding L to associate (left-to-right) to the other syllables.

(2.56)


Zoll (2003) argues that both of these analyses miss an important generalization: in Kukuya, H is not allowed to spread unless it is the only tone in the melody. Additionally, when we consider data beyond that originally given in (2.1) in §2.1.1, we find a similar issue to Kukuya. Thus, a simple directional left-to-right analysis of tone mapping in Mende correctly obtains the surface form for LHH forms like (2.24a) [ndàvúlá] 'sling', it incorrectly predicts that LLH is impossible. Unlike Kukuya, however, LHH is also allowed, so rules like Hyman (1987)'s L-Spreading rule, which explicitly avoid LHH patterns, cannot be applied to Mende.

Leben (1978) offers an accentual analysis to deal with the additional patterns in (2.24); however, this leads Leben to posit a more complex tone mapping convention than that in (2.48):

(2.57) Mende Tone Mapping Rule (Leben, 1978, p.200, (52))

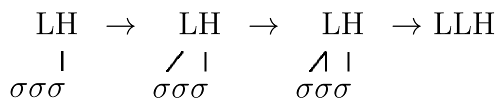
a. Associate a final H with the rightmost syllable
b. For any tones that are not associated with any syllables, associate the first tone with the first syllable, the second with the second, and so on.
c. Any syllable that has no tone is associated with the tone of the preceding syllable, if there is one. Otherwise, tone assignment takes place according to the well-formedness condition. [i.e., (2.17) –AJ]

For example, bisyllabic HL melody forms with a final falling tone (like (2.24a) [kɔ́nyô] 'friend') have an underlying association between the H and the final syllable. Left-to-right association (2.57b) thus proceeds as follows:

(2.58) HL → HL → HL → HF
$\sigma\sigma$ $\sigma\sigma$ $\sigma\sigma$

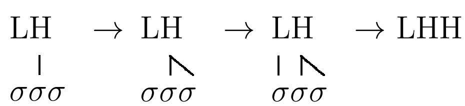
Note that as the Mende Tone Mapping Rule in (2.57) refers specifically to H tones, a LR form contour cannot be generated with this paradigm, which correctly predicts the absence of this melody realization pattern (recall from §2.2.1.3 that LR is unattested).

The new directionality rules (2.57b) and (2.57c) break up left-to-right spreading in two steps. This is necessary to correctly derive LLH (as in (2.24)c [lèlèmá] 'mantis') as follows. First the H associates to the final syllable (2.57a). Then, the L associates to the initial syllable *only* (2.57b). This can be seen in the second step below in (2.59).

(2.59) 

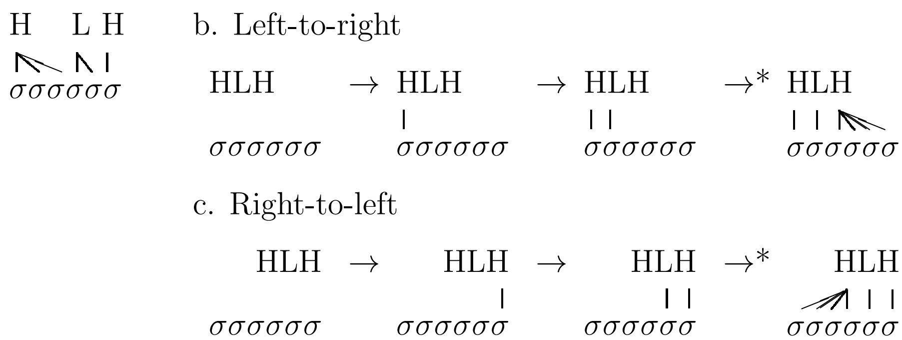
After the L associates to the first syllable in the second step above in (2.59), by (2.57c), the second syllable associates to the L, and not the H, to correctly derive LLH. This nuance in the association is necessary, as simple left-to-right association would, having associated the L to the first syllable, incorrectly associated the H to the second.

Thus, in order to account the LHH pattern on forms such as (2.24a) [ndàvúlá] 'sling', it is necessary to posit an underlying association between the H and the second syllable, in order to avoid the derivation in (2.59). Given this initial association, the H then associates to the rightmost syllable by (2.57a), and then (2.57b) fills in the association between the L and the initial syllable.

(2.60) 

Thus, Leben (1978)'s analysis for the full set of Mende data requires the additional stipulations in (2.57). However, Leben does not make clear what the cross-linguistic status of the mapping rules in (2.57) is. While he does go on in the same paper to analayze some of the complications in the Hausa pattern with (2.57b) and (c), he does not make it explicit whether or not they are meant to constrain or expand the rule-based theory of tone mappings in general. As such, it is difficult to interpret the typological predictions of (2.57).

Finally, in order to capture the combination of positional, directional, and quality-specific association seen in N. Karanga Shona, Hewitt and Prince (1989) posit bidirectional 'edge-in' tone mapping (Yip, 1988b). Clearly, neither pure left-to-right or right-to-left spreading will achieve the correct patterns. We could posit an HL melody for the ASSERTIVE and a HLH melody for the NON-ASSERTIVE, but these alone cannot generate the tertiary spreading of the initial H tone or, in the case of the NON-ASSERTIVE, the medial stretch of L. To illustrate, the correct APR for the NON-ASSERTIVE (2.28f) HHHLLH pattern is given below in (2.61a), and (2.61b) and (2.61c) show how it cannot be derived with simple directional mappings.

(2.61) a.


There are two essential pieces to Hewitt and Prince (1989)'s analysis of these two patterns. First, they invoke Yip (1988b)'s concept of *edge-in association*, in which the leftmost and rightmost elements of the melody tier are, in that order, associated to the leftmost and rightmost elements of the timing tier, respectively.

(2.62) Edge-in association (EIA Hewitt and Prince, 1989, p.178, (7))
For a melody /a...z/,

a. Link a to the initial timing tier unit
b. Link z to the final timing tier unit
c. Link any remaining melody units left-to-right

Hewitt and Prince posit that, the verb roots in both (2.27) and (2.28) carry an underlying H, which associates to the left edge of the root. For the NON-ASSERTIVE tense, they posit an additional -H tone suffix, which by EIA then associates to the right edge. L tones are analyzed as the result of a default association rule that occurs following all of their proposed rules for H association.

(2.63)

	Assertive	Non-assertive
URs	H σσσσ	HH σσσσ
EIA (a)	H | σσσσ	HH | σσσσ
EIA (b)	—	H H | | σσσσ

Note that the 'alternating' nature of this definition is crucial to their analysis; the leftmost melody element must first associate to the leftmost timing tier unit. Thus a single H associates to the left edge, and not the right.

For the ternary spreading of the initial H, Hewitt and Prince posit two 'local' (for them, meaning strictly adjacent) spreading rules, Root Tone Spread and General H-spread. The first applies to root-initial H tones, and the second applies to any H tone.

(2.64) a. Root Tone Spread (RTS; Hewitt and Prince, 1989, p. 181, (12)))

H

$[_{\text{root}}$ σ σ

b. General Tone Spread (GTS; Hewitt and Prince, 1989, p. 181, (13)))

H

σ σ

Crucially, Hewitt and Prince posit that both of these rules are blocked by the structural adjacency version of the OCP (2.11) from creating any structurally adjacent H tones. (This OCP also motivates the deletion of the suffix H in monosyllabic and bisyllabic Non-assertive forms, as it cannot associate without creating an OCP violation.) Example derivations for these rules are given below in (2.65).

(2.65)

	Assertive	Non-a (4 σ)	Non-a (6 σ)
URs	H $\sigma\sigma\sigma\sigma$	HH $\sigma\sigma\sigma\sigma$	HH $\sigma\sigma\sigma\sigma\sigma\sigma$
EIA	H $\sigma\sigma\sigma\sigma$	H H $\sigma\sigma\sigma\sigma$	H H $\sigma\sigma\sigma\sigma\sigma\sigma$
RTS	H $\sigma\sigma\sigma\sigma$	H H $\sigma\sigma\sigma\sigma$	H H $\sigma\sigma\sigma\sigma\sigma\sigma$
GTS	H $\sigma\sigma\sigma\sigma$	(blocked)	H H $\sigma\sigma\sigma\sigma\sigma\sigma$
SR	H L $\sigma\sigma\sigma\sigma$	H LH $\sigma\sigma\sigma\sigma$	H L H $\sigma\sigma\sigma\sigma\sigma\sigma$
String	HHHL (2.27d)	HHLH (2.28d)	HHHLLH (2.28f)

The surface strings in (2.65) illustrate how Hewitt and Prince's analysis derives the correct forms for both tenses in N. Karanga. To summarize, Hewitt and Prince's tone mapping relies on initial edge-in association, which then feeds additional spreading rules. This edge-in association operates in a very different manner from the left-to-right and right-to-left tone mappings we saw earlier for Mende and Hausa. It should also be noted that Hewitt and Prince's definition of edge-in association differs from Yip (1988b)'s original formulation, which Yip convincingly argues is necessary to describe some phenomena in Arabic morphology and tone phonology in a rule-based, autosegmental framework. It is unclear, then, what exactly the theory of edge-in association is, and what exactly the range of association patterns is that it is meant to allow.

Thus, the directional, quality-sensitive, and positional tone-mapping generalizations discussed earlier in this chapter can all be analyzed using a derivational framework. However, this has led to a proliferation of association paradigms, from basic right-to-left and left-to-right association, to Leben (1978)'s revised association paradigm, to Yip (1988b) and Hewitt and Prince (1989)'s edge-in association paradigms.

Furthermore, several of these analyses employed language-specific rules. There thus remains a question of what, exactly, the predicted typology of tone-mapping patterns under a derivational autosegmental framework is. It also, as Zoll (2003) points out for the restrictions on multiple association of H tones in Kukuya, can miss surface generalizations in the patterns. The following shows how analyses in Optimality Theory, which offers a clear way of looking at the typology through permutations of rankings of a set of universal, violable constraints, attempt to address these issues.

2.3.1.2 Tone mapping with optimization

Perhaps the most comprehensive example of a theory of language-specific association well-formedness that is grounded in violable constraints, Zoll (2003)'s Optimal Tone Mapping (OTM) has two main goals. One, it aims to highlight quality-sensitive association; i.e., generalizatons about association that are true for specific tonemes (e.g., H or L). Two, it claims that directional association is fundamentally about *morphological* alignment, and is not phonological in nature. However, while Zoll's theory makes the important contribution of bringing quality-sensitive association to the fore, it is too strong in that it cannot capture the positional association of N. Karanga Shona (a flaw Zoll readily admits).

At the core of OTM are two simple markedness constraints, Clash and Lapse. The former penalizes adjacent H toned TBUs, and the latter penalizes L toned TBUs.

(2.66) OTM Constraints (Zoll, 2003, p.239, (26))

a. Clash
There is no H sequence on adjacent TBUs (*HH, *HF, etc.)
Assign one violation for each clashing pair of TBUs.

b. Lapse
There is no non-H sequence on adjacent TBUs (*LL, *LR, *∅L, etc)
Assign one violation for each lapsing pair of TBUs.

It should be noted that Clash and Lapse are distinct from the OCP, because Clash and Lapse penalize adjacent TBUs of identical tones regardless of whether or not they are associated to distinct autosegments, whereas the OCP penalizes only those associated to distinct autosegments:

(2.67)

<table>
<tr><th></th><th>CLASH</th><th>OCP</th></tr>
<tr><td>H H
| |
σ σ</td><td>*</td><td>*</td></tr>
<tr><td>H
/\
σ σ</td><td>*</td><td></td></tr>
</table>

The relative ranking of CLASH and LAPSE determines whether a H tone or a L tone in a melody will spread. Zoll shows how ranking CLASH over LAPSE thus precisely captures the behavior of tones in Kukuya. Recall from (2.21) that Kukuya mostly seemed to follow a left-to-right association pattern, with the exception of LLH forms:

(2.68) Kukuya word tone patterns (=2.21)

H		HH		HHH	
a. bá	‘palms’	b. bágá	‘show knives’	c. bálágá	‘fence’
F		HL		HLL	
d. kâ	‘to pick’	e. sámà	‘conversation’	f. káràgà	‘to be entangled’
R		LH		LLH	(*LHH)
g. sǎ	‘ knot’	h. kàrá	‘paralytic’	i. m^wàrə̀gí	‘younger brother’
R-F		LF		LHL	
j. bvı̌̂	‘falls’	k. pàlî	‘goes out’	l. kàlə́gì	‘turns around’

Zoll (2003) points out a clear generalization about the surface associations: H is not allowed to spread in any of these cases. This fact is captured simply in OTM by ranking CLASH over LAPSE:

(2.69)

L H σσσ	Clash	Lapse
L H σσσ	*!	
☞ L H σσσ		*

H L σσσ	Clash	Lapse
H L σσσ	*!	
☞ H L σσσ		*

However, H does need to spread in the case that it is the only tone in the melody, as in (2.68b) and (2.68c). Because Clash is still a violable constraint, this can be achieved by ranking it below faithfulness constraints such as Dep, which militates against epenthesis of tones in order to satisfy Clash.

(2.70)

H σσσ	Dep	Clash	Lapse
L H σσσ	*!		*
☞ H σσσ		**	

Thus, ranking Clash above Lapse, but below any faithfulness constraints, accurately describes the association pattern in Kukuya. Importantly, it captures the generalization that H cannot spread in the presence of a L tone.

However, Zoll cannot get away from directionality completely. For the Hausa patterns in §2.2.1.1, Zoll invokes *morphologically*-based directionality, noting that many of the melodies which associate right-to-left come from suffixes. She thus invokes an Align-R constraint (McCarthy and Prince, 1993) which works to keep associations from the suffix tones as far to the right as possible. For example, (2.18g) [jìmìnúu] 'ostriches' takes its LH melody from the /-uu/ plural suffix. To achieve right-to-left directionality, Zoll posits that the elements of hte suffix are subject to the following constraint:

(2.71) ALIGN(plural, R, stem, R)

Plural tones align with the right edge of the verb stem.

Zoll does not give an explicit rule for calculating violations of this ALIGN constraint. From her discussion, it must be interpreted this way: "For each tone in the plural suffix, assign one violation for each TBU in between the right edge of the stem and the rightmost TBU to which the tone is associated." If this is ranked above both CLASH and LAPSE, then we correctly obtain right-to-left association:

(2.72)

L H_{PL} $\sigma\sigma\sigma$	ALIGN	CLASH	LAPSE
a. L H_{PL} $\sigma\sigma\sigma$	*!*	*	
☞ b. L H_{PL} $\sigma\sigma\sigma$	*		*

In both forms in (2.72), the rightmost association of the L tone does not align with the right edge of the word stem. However, it is misaligned more in (2.72a) than in (2.72b), because the spreading of the H in (2.72a) means that the rightmost association of the L is two TBUs from the right edge of the stem, rather than one in (2.72b). Thus, (2.72b) is the optimal candidate. Again, this hinges on a particular interpretation of the ALIGN constraint: note that in (2.72b), the association between the L and the first σ, which is two TBUs from the right edge, is not counted as a violation of the ALIGN constraint.

Zoll (2003) makes the strong claim that directional ALIGN constraints can only be morphologically triggered. However, there are two issues with Zoll's use of morphological ALIGN constraints. First, cannot capture patterns like Mende, in which directionality is the correct generalization (at least for 90% of the forms, as noted above in Footnote 3), but is not morphologically motivated.

Second, ALIGN constraints are theoretically undesirable because they predict bizarre patterns (Eisner, 1997b; McCarthy, 2003). For example, Eisner (1997b) shows an ALIGN constraint which produces 'centering' of association of H tone to the middle of a word. This is due exactly to 'counting' of distance necessary for the evaluation of ALIGN that distinguishes candidate (2.72a) from (2.72b) above. Eisner (1997a,b) instead argues that markedness constraints should be fundamentally *local*, and shows how the evaluation of ALIGN constraints is fundamentally *global*—that is, they count distance over an entire domain. The following chapters show how to precisely define local versus global evaluation, and comes to the same conclusion as Eisner.

Finally, beyond any theoretical undesirabilities of ALIGN, empirically, it cannot cover the facts of N. Karanga. As discussed above, toned verbs in N. Karanga ASSERTIVE and NON-ASSERTIVE tenses begin with a plateau of up to three H tones, respecting the OCP.

(2.73) N. Karanga verbs

ASSERTIVE	NON-ASSERTIVE
H	H
HH	HL
HHH	HLH
HHHL	HHLH
HHHLL	HHHLH
HHHLLL	HHHLLH
HHHLLLL	HHHLLLH

From an OTM standpoint, both CLASH and LAPSE must both be ranked low, as spans of Hs and Ls are both allowed. Thus, explanation for the pattern must come from somewhere else. However, this behavior cannot be captured by ALIGN constraints. Ignoring whether or not ALIGN can be phonological or morphological, let us posit an

Align constraint similar to the one in (2.71) which forces tones to be aligned with the right edge of the word. Given such a constraint, for words of sufficient length, the attested candidate is harmonically bound by candidates in which the H spreads all the way to the right.

As an example, let us consider five- and seven-syllable forms in the Non-assertive. That the first H never spreads all the way to the syllable immediately preceding the second H can be expressed by ranking OCP above Align. In the five-syllable form (2.74), this obtains the attested form. However, in the seven-syllable form (2.75), it incorrectly chooses *HHHHHLH over HHHLLLH. Assume L tones are inserted due to the usual interaction of a high-ranked constraint requiring tone specification and a low-ranked Dep(L) constraint against the insertion of L tones.

(2.74)

H H σσσσσ	OCP	Align	Clash	Lapse
a. HLHL σσσσσ		*!**		*
b. HH σσσσσ	*!	*	****	
☞ c. HLH σσσσσ		**	**	

(2.75)

H H σσσσσσσ	OCP	Align	Clash	Lapse
☞ a. HLH σσσσσσσ		***	***	
✖ b. H L H σσσσσσσ		**!***	**	

The reason for this is that Align is absolute—it wants tones as far to the right edge as possible. However, spreading of the initial H in N. Karanga is only *ternary*. Thus, for forms with more than 5 syllables, such as in (2.75), the attested form (marked with a ✖) loses to an unattested form whose initial H has associated as far to the right as possible.

This highlights an issue OT has with bounded spreading: it is hard to view spreading over exactly three TBUs as optimal over spreading all the way (as motivated by Align) or not spreading at all (as motivated by Clash). Topintzi (2003) offers one possible solution to this problem by invoking metrical feet, and proposing a constraint FtHdLocality which bars spreading past the head of a foot.

(2.76) FtHdLocality: Spread a tone only up to the adjacent foot head.

Assuming binary and trochaic foot construction, the tableau is as follows for an Associative form (i.e., with one underlying H tone). Topintzi's analysis also requires that the initial H is underlyingly associated to the initial syllable and that L tone syllables are underlyingly specified. While she uses different constraints to motivate spreading, I will stick with Align-R for the sake of continuity with the above discussion.

(2.77)

H | $\sigma\sigma\sigma\sigma$	FtHdLocality	Align	Clash	Lapse
a. H | $(\sigma\sigma)(\sigma\sigma)$		*!**		***
b. H $(\sigma\sigma)(\sigma\sigma)$	*!		***	
☞ c. H $(\sigma\sigma)(\sigma\sigma)$		**	**	

However, as Topintzi (2003) herself notes (p. 332), there is no evidence at all for metrical feet in N. Karanga Shona. This renders this particular analysis *ad hoc*. Furthermore, the constraint FtHdLocality seems tailor-made for this particular analysis. Thus, it does not represent any theoretical improvement over the language-specific rules posited by Hewitt and Prince (1989).

2.3.1.3 Interim summary: well-formedness of association

We have now seen both derivational and optimality-based explanations of the range of melody association in tone-mapping patterns. Derivational frameworks were able to describe the full range of directional, quality-specific, and positional association generalizations through the use of a variety of association paradigms and language-specific rules. However, given the language-specificity of these analyses, in typological terms it is unclear what *range* of association patterns it predicts to exist. In Zoll (2003)'s OTM, the constraints were universal and thus it makes clear typological predictions, but these predictions both overgenerate (through the Align constraints) and undergenerate (with respect to N. Karanga Shona).

In sum, this discussion has shown how there is no theory of association which can both capture all of the surface generalizations in a straightforward way while also making clear, yet restrictive, typological predictions. The following chapter will introduce a computational theory of local constraints which has clear typological predictions. However, in order to be successful as a theory of tone, this notion of locality must *also* capture long-distance generalizations. The following shows how derivational and optimality-based grammars achieve this using autosegmental representations, which will motivate the definition in Chapter 5 of a computational theory of local constraints over autosegmental representations.

2.3.2 Analyses of long-distance generalizations

Having seen how derivational and OT grammars analyze basic association constraints, we can now turn to how these theories are able to capture the long-distance tonal patterns originally discussed in (2.2.2).

2.3.2.1 Unbounded Tone Plateauing

Recall from §2.2.2.1 that unbounded tone plateauing (UTP) corresponds to a long-distance surface generalization such that for *any* two H-toned TBUs, *all* intervening TBUs must also be H-toned. In autosegmental terms, this means that there can only be one distinct H tone on the tonal tier. The examples below from Luganda are repeated from (2.36).

(2.78)	a.	'chopper'	mutéma	LHL	LHL $\mu\mu\mu$
	b.	'log'	kisikí	LLH	LH $\mu\mu\mu$
	c.	'log chopper'	mutémá+bísíkí	LHHHHH	L H $\mu\mu\mu\mu\mu\mu$
	d.	" "	*mutéma+bisikí	*LHLLLH	* LH L H $\mu\mu\mu\mu\mu\mu$

Again, as strings of TBUs, this appears to be the interaction of H-toned TBUs at long distances. However, if the intervening TBUs are analyzed as underlyingly unspecified for a tone, then in an APR the Hs are adjacent on the melody tier. This is exactly how Hyman and Katamba (2010) analyze it, formalizing the process (which they refer to as 'H tone plateauing') thusly:

(2.79)
$$\begin{array}{ccccccc} H & & H & & & H & \\ \mu & \mu^n & \mu & \rightarrow & \mu & \mu^n & \mu \end{array}$$

The plateau in (2.32a) [mutémá-bísíkí] 'log-chopper' can thus be analyzed as follows:

(2.80)
$$\begin{array}{ccccc} \text{H} & & \text{H} & & \text{H} \\ \text{/mu-tém-a/} & + & \text{/bi-sikí/} & \rightarrow & \text{mutémá-bísíkí} \end{array}$$ 'log-chopper'

While associated to TBUs which may be separated by any number of toneless TBUs, the Hs are adjacent on the melody tier, and so their interaction can be seen as local in this way. The advantages of this kind of locality are thoroughly documented by Odden (1994). While as Odden stresses that this notion of locality is predicated on underspecification of ignored timing tier units for the particular feature, as in (2.79) and (2.80), later chapters will show that, using the definition of locality defined in Chapter 3, strict adherence to the OCP will also allow seemingly long-distance patterns to remain local over APRs, and that this notion of locality does not require underspecification.

One aspect of UTP that the rule-based analysis misses is that UTP is part of a cross-linguistic tendency to avoid surface HLH sequences—note that this can also be seen in the absence of HLH melodies in Mende. In OT, which motivates alternations based on surface markedness constraints, the generalization that two distinct Hs cannot appear in the domain becomes the driving force of the alternation. Yip (2002), for example, motivates UTP in Digo this with a constraint *Trough which penalizes "dips or troughs" in between H-toned syllables (p. 137). While she does not formalize it, the behavior of the constraint can be interpreted as penalizing surface HLH melodies:

(2.81) *Trough: Troughs of L tones are not permitted between H tones. Mark one violation per every HLH sequence in the melody.

If we rank *Trough over a faithfulness constraint NoFusion militating against fusing two autosegments from the input (this is Yip's terminology; see also the Unif-IO constraints of Pater (2004) and Meyers (1997)), but below other Faithfulness constraints (e.g., a Max constraint against deleting the underlying H tones), then plateauing emerges as the optimal solution to inputs with two underlying H tones. The following example tableau assumes, as the rest of the chapter has, that surface candidates adhere to full specification and the NCC.

(2.82)

H H μμμμμμ	Max	*Trough	NoFusion
a. LHLHL μμμμμμ		*!	
b. LHL μμμμμμ	*!		
☞ c. L H L μμμμμμ			*

In the above tableau, candidate (a), in which both underlying Hs surface intact, violates *Trough, due to the resulting HLH sequence on the melody tier. Candidate (c), which fuses the two Hs together, creating a plateau between them, thus avoids this sequence and thus violating *Trough, although it violates the lower-ranked NoFusion.

Thus, in OT we can also capture this long-distance generalization via a constraint on the melody. Note that, unlike the rule-based analysis, through the *Trough constraint it motivates the alternation with the surface generalization, first noted in §2.2.2.1, that only one H can appear in the melody tier.

2.3.2.2 Hirosaki Japanese

Recall that a similar generalization exists in Hirosaki Japanese: there must be exactly one H on the melody tier. There are two additional restrictions: this H cannot be associated to multiple TBUs, and both a H and a following L tone can only associate to the same TBU on a word-final TBU. The relevant APRs are repeated below from (2.38) and (2.83) in §2.2.2.2.

(2.83) LHLL = LH L / μμμμ   LLLH = LH / μμμμ   LLLF = LHL / μμμμ

HLLF = H LHL *LLLL = * L

$\mu\mu\mu\mu$ $\mu\mu\mu\mu$

*LHHL = * LH L *LFLL = * LH L

$\mu\mu\mu\mu$ $\mu\mu\mu\mu$

Haraguchi (1977) analyzes this pattern in an early derivational autosegmental framework using starred accents. Under his analysis, non-final H-toned morae and final F-toned morae are underlyingly accented. He stipulates that Hirosaki Japanese has a basic LHL melody, and that the H initially associates to either an accented mora—of which there may only be one—or the final mora in the word. The following is adapted from Haraguchi (1977) (Haraguchi has tones associate directly to vowels, whereas the following, without loss of generality, has tones associate to morae, in keeping with the representational assumptions in the rest of the chapter.)

(2.84) Tone association rule (TAR; Haraguchi, 1977, p. 71, (4))

a. If a string as at least one $\overset{*}{\mu}$, associate the H tone of the basic tone melody with the leftmost $\overset{*}{\mu}$.
b. If it has no $\overset{*}{\mu}$, associate the H tone of the basic tone melody with the rightmost V.

The first row of APRs in (2.83) for LHLL, LLLH, and LLLF forms would thus be derived from underlying $\mu\overset{*}{\mu}\mu\mu$, $\mu\mu\mu\mu$, and $\mu\mu\mu\overset{*}{\mu}$, respectively. The following diagrams show the initial association for these forms. Note that initial association for (2.85a) and (2.85c), which are accented, applies according to TAR (a), whereas initial associaton for (2.85(b)), which is unaccented, applies according to TAR (b).

(2.85) Initial associations by (2.84)

a.
$$\mu\overset{*}{\mu}\mu\mu \xrightarrow{\text{TAR (a)}} \begin{array}{l}\;\;\text{LH L}\\ \;\;\;\;| \\ \mu\overset{*}{\mu}\mu\mu\end{array}$$

b.
$$\mu\mu\mu\mu \xrightarrow{\text{TAR (b)}} \begin{array}{r}\text{LHL}\;\\ |\;\;\; \\ \mu\mu\mu\mu\end{array}$$

c.
$$\mu\mu\mu\overset{*}{\mu} \xrightarrow{\text{TAR (a)}} \begin{array}{r}\text{LHL}\;\\ |\;\; \\ \mu\mu\mu\overset{*}{\mu}\end{array}$$

Association to the remaining TBUs then proceeds according to Haraguchi's *Universal Tone Association Conventions*, a modification of Goldsmith (1976)'s Well-formedness Condition (2.17) which restricts multiple association to initial and final tones in the melody:

(2.86) The Universal Tone Association Conventions (UTAC) of Haraguchi (1977, p.11)

a. If a domain contains only one free tone, or if it contains only one free tone to the right (or left) of a bound [='associated' –AJ] tone, the free tone should be associated with every free tone- bearing unit or every free tone-bearing unit on the same side of the bound tone. I.e.,

$$\left(\begin{array}{c}V\\ |\\ T_1\end{array}\right)\begin{array}{c}P\\ \vdots\\ T_2\end{array}\;//$$

(where P is the maximal sequence of free tone-bearing units, and T_2 is a free tone. // indicates that this is a mirror image process.)

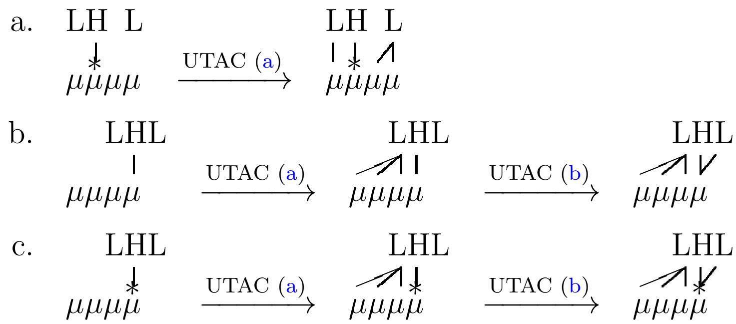
b. If a domain contains no [tone-bearing unit] to the right (or left) of a bound [tone-bearing unit], and if it contains at least one free tone, the free tone should be associated with the bound tone-bearing unit. I.e.,

$$\begin{array}{ll}V & \\ |\;\diagdown & \\ T & Q\end{array}\;//$$

(where Q is the maximal sequence of free tones.)

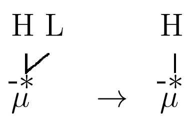
For Haraguchi's analysis of Hirosaki Japanese, UTAC (a) has the effect of associating the initial and final L tones of the LHL to any unassociated TBUs. For an L tone with no unassociated TBU, UTAC (b) associates it instead to the nearest 'bound' (=associated) TBU, which in this case is the TBU associated to the H. Thus:

(2.87) Remaining associations by (2.86)



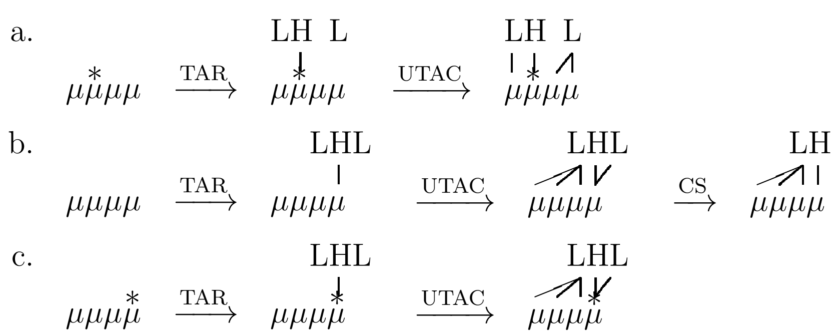
As the Universal Tone Association Conventions create an unwanted contour in the case of the unaccented (b) (which, remember, should surface as LLLH). To deal with this, Haraguchi (1977) posits the contour simplification rule for unaccented TBUs in (2.88) (following Haraguchi's notation, $\bar{\overset{*}{\mu}}$ denotes an unaccented mora). A similar, yet distinct, rule (omitted for brevity) is necessary for when (2.86) creates an initial contour with the initial L and a H associated to an initial accented mora.

(2.88) Contour simplification (CS; Haraguchi, 1977, p. 72, (6))



The full derivations for the APRs in (2.83) for LHLL, LLLH, and LLLF forms are thus as follows:

(2.89) Derivations for LHLL, LLLH, and LLLF



Thus, Haraguchi (1977)'s analysis correctly accounts for the tone patterns. Again, to capture the generalization that one and only one H tone appears on the melody tier, this analysis requires the stipulation that the basic melody in Hirosaki Japanese is LHL. The restriction of contours to the word-final mora comes from two places: one, the restriction of the UTAC to restrict multiple association to the word edges, and language-specific rules that desimplify contours. Note also that evaluation of Haraguchi (1977)'s tone association rule in (2.84) is again *global*: it must search the entire TBU tier for an accented $\overset{*}{\mu}$, and, if failing, associate it to a final μ.

While, to the best of my knowledge, no OT analyses of these data exist, a very similar analysis could be posited using two align ALIGN constraints, one forcing an H to align to an accented mora, and a lower-ranked ALIGN constraint forcing the H to align to the left edge of the word.[9] A high ranking of Zoll (2003)'s *CLASH constraint would then prevent the H from associating to multiple TBUs. The restriction of contours to the final TBU can be captured by a COINCIDE(contour, final TBU) constraint, as also utilized by Zoll (2003). However, there is still the question of the melody. One solution, to follow Haraguchi and, as to be discussed below, Breteler (2013), is to simply stipulate that the LHL melody is part of the input. However, we can also treat the restriction to exactly one H as part of the grammar. We have already seen that the *TROUGH constraint from the UTP analysis can force there to be at *most* one H, as we have already seen, but to *require* a H tone, we would need a highly-ranked HAVEH constraint which is violated by APRs whose melody does not contain a H tone.

Regardless, the data in Hirosaki Japanese are analyzable in both derivational frameworks, and the long-distance quality of the pattern can be attributed to a restriction on the melody to a L(H)L. However, we also see that these analyses suffer from the same pitfalls as for the tone-mapping patterns above: the derivational analysis invokes globally-evaluated association rules and language-specific rules that 'edit' the resulting associations, and the OT analysis requires ALIGN constraints. Additionally, at least in Haraguchi (1977)'s derivational analysis, the melody is simply stipulated in an statement external to rest of the grammar.

[9] A detailed OT analysis of the data in Kobayashi (1970) is given in Li (2014).

2.3.2.3 Wan Japanese Type β

A similar characteristic can be seen in previous analyses of the Wan Japanese Type β pattern. Recall that, phrase-medially, Wan Japanese Type β nouns follow one of the following two patterns, depending on whether or not they are in isolation or are suffixed:

(2.90) Isolation: $\mathrm{H}^n\mathrm{LHL}$

Suffixed: $\mathrm{H}^n\mathrm{LHH}\text{-}\mathrm{H}^m\mathrm{L}$

Previous accounts of these patterns explain the difference between them through association at different levels. Kubozono (2011b), for example, posits an underlying HLH tone for Type β, which associates right-to-left at (what I will call) the stem level. To account for the fact that the second H associates to the penultimate mora of the stem, he posits that the stem-final mora is 'extrametrical' and thus invisible to association. An example is given below for a 5-μ form.

(2.91)
$$\begin{array}{ccc} \mathrm{HLH} & & \mathrm{HLH} \\ & & \;\;| \; | \; | \\ \mu\mu\mu\mu\mu & \rightarrow & \mu\mu\mu\mu\mu \end{array}$$

Kubozono (2011b) posits the final L a boundary tone, which is added to the final mora of the *word*. This creates the difference between nouns in isolation and suffixed nouns. In nouns in isolation, the boundary L fills in the stem-final (=word-final) mora following the penultimate H-toned mora. In suffixed nouns, this leaves a gap of unassociated morae between the stem-penultimate mora and the word-final mora. The following example contrasts a 5-μ noun in isolation with a 5-μ noun followed by a 2-μ suffix domain.

(2.92)
$$\begin{array}{cc} \text{Isolation} & \text{Suffixed} \\ \mathrm{HLHL} & \mathrm{HLH} \quad\;\; \mathrm{L} \\ \;\;|\;|\;|\;| & \;|\;|\;| \qquad\; | \\ \mu\mu\mu\mu\mu & \mu\mu\mu\mu\mu\text{-}\mu\mu \end{array}$$

Both Hs then 'fill' in this gap and any unassociated morae at the beginning of the word. The entire derivation is summarized in (2.93).

(2.93)

	Isolation	Suffixed
Underlying form	HLH $\mu\mu\mu\mu\mu$	HLH $\mu\mu\mu\mu\mu$-$\mu\mu$
Stem level	HLH $\mu\mu\mu\mu\mu$	HLH $\mu\mu\mu\mu\mu$
Word level		
Boundary tone	HLHL $\mu\mu\mu\mu\mu$	HLH L $\mu\mu\mu\mu\mu$-$\mu\mu$
H spreading	HLHL $\mu\mu\mu\mu\mu$	HLH L $\mu\mu\mu\mu\mu$-$\mu\mu$
Output	HHLHL	HHLHH-HL

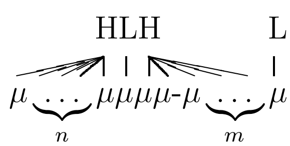
Breteler (2013) gives an account in Stratal OT (Booij, 1997; Kiparsky, 2000), in which optimization takes place at two levels with distinct grammars, which is in many ways similar to the above analysis. One major difference is that Breteler argues against the second L as a word-level boundary tone, as Type α words do not exhibit it (recall that the surface melody for Type α is invariably HLH). He thus posits HLHL as the underlying melody for Type β. At the stem level, the melody associates right-to-left, and then at the word-level a quality-specific *SPREAD(L) constraint against L spreading (virtually identical to Zoll (2003)'s *LAPSE) forces the second H to spread over any additional morae in the suffix domain.

Both the derivational and OT accounts thus both have several important similarities. First, the surface difference in tone patterns between nouns in isolation and suffixed nouns is derived through different association processes occurring at different levels. Thus the 'long-distance' generalization at the surface discussed in (2.2.2.3)—that whether or not a final HHL sequence of mora is allowed depends on whether or not a '-' boundary occurs earlier in the word—is derived not solely from a statement about the melody, as in UTP and Hirosaki Japanese, but from the interaction of the autosegmental structure and the morphophonology.

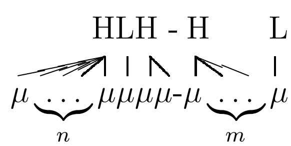
Second, as in Haraguchi (1977)'s analysis for Japanese, they posit that the underlying melody (whether HLH or HLHL) is part of the input to the grammar, rather than part of the grammar itself. This is notably distinct from, example, a *TROUGH constraint in OT, which explicitly makes melody well-formedness part of the grammar. This can be turned into an empirical question: do speakers of Wan Japanese consider words with other melodies, such as LHL melodies of Tokyo Japanese, grammatical? If we posit that the melodies of Wan words are simply an accident of the lexicon, then we expect that speakers should find other melodies acceptable. This is likely not the case, as, for example, loanwords—which likely are imported through other dialects—take on Wan melodies (Kubozono, 2011b; Breteler, 2013). While a full exploration of this issue is beyond the purview of this dissertation, this is a major difference between these analyses and the one presented in this dissertation, which makes melody well-formedness an explicit part of the grammar.

Finally, as first mentioned in §2.2.2.3, both analyses assume that strings of the form H^{n}LHH-H^{m}L correspond to APRs of the form in (2.94a), rather than that in (2.94b).

(2.94) a.



b.



The difference again is that for (2.94a), the stem and suffix domains share a single melody, but in (2.94b), they each get their own melody, with the stem melody conforming to the phrase-medial HLH melody. The analysis given later in this dissertation will argue, based on the idea of locality, for the latter.

2.3.3 Interim summary: analyses of long-distance generalizations

This section has shown how the long-distance generalizations in UTP, Hirosaki Japanese, and Wan Japanese can be analyzed in both derivational and optimality-based frameworks. In UTP and Hirosaki Japanese, the generalizations were captured by constraining the melodies, either by rules/constraints that affected the melody (UTP) or by stipulating the melody in advance (Hirosaki Japanese). In Wan Japanese, the long-distance generalization was captured through differing association paradigms operating at different morphophonological levels. However, these analyses suffer from the same pitfalls as the tone-mapping analyses discussed above. Derivational frameworks employed language-specific rules and association paradigms whose operation require "global" evaluation over entire autosegmental structures. Similarly, OT-based analysis of Hirosaki Japanese requires ALIGN constraints, and Wan Japanese required a derivation-like stratal approach. Additionally, while the melody well-formedness generalization in UTP was handled as part of the grammar, for Hirosaki Japanese and Wan Japanese, melody well-formedness was instead relegated to the lexicon.

2.4 Conclusion

This chapter has introduced, along with autosegmental representations, a number of examples of phenomena in tone that illustrate the kinds of well-formedness generalizations over these representations that any theory of tone must be able to capture. The chapter first reviewed basic generalizations referring to how tones associate to TBUs, illustrating using variation in tone-mapping phenomena. Namely, these included *directional* generalizations, in which multiple association of tones to TBUs or vice-versa were only allowed on one word edge or another, *quality-specific* generalizations which apply to the associations of either H or L tones, or *positional* generalizations referring to association to TBUs at specific positions in the word. It was then shown how *long-distance* phenomena in tone can be cast autosegmentally as generalizations referring to melodies and morphophonological information.

We then saw how both previous analyses in both derivational and OT frameworks manipulated autosegmental representations to capture these generalizations, although there were a number of issues. Generally speaking, derivational frameworks have been powerful enough to capture all of these phenomena, but they required a variety of language-specific association paradigms and rules. This proliferation of analytical tools makes it unclear, from a typological perspective, what the theory predicts as a possible or impossible association paradigm. Under OT, through the notion of universal, violable constraints we get a well-defined notion of a possible association paradigm, but OT both undergenerates (in the case of N. Karanga Shona) and overgenerates (in the case of Align constraints).

Finally, what is missed by *both* the derivational and OT theories of association is that, while they both use theoretical machinery which evaluates APRs *globally*, the generalizations reviewed in this chapter are fundamentally *local* over APRs in a well-defined sense. The purpose of the following chapter is to define this notion of locality over strings, and then to argue, through the use of the long-distance generalizations mentioned in this chapter, that this notion of locality needs to be extended to APRs in order to capture the full range of attested tone patterns. The remainder of this dissertation is thus concerned with presenting an alternative view of language-specific well-formedness which defines and focuses on this property of locality over APRs and shows that it is favorable to the analyses presented in this chapter because it is both *sufficient* to capture the range of attested tone patterns, but it also provides for a *restrictive* theory of the typology of tone. It will also be shown how this property can inform both a theory of learning over autosegmental representations and a theory of input-output transformations over autosegmental representations.

Chapter 3

LOGICAL GRAMMARS

The purposes of this chapter are twofold. The first goal is to introduce a computational notion of *locality* in well-formedness that gives clear, yet restrictive, typological predictions, and is thus a strong alternative to the rule- and optimization-based grammars discussed in the previous chapter. This notion is drawn from a literature defining logical constraints over strings and applying these constraints to theories of well-formedness and learning in segmental phonology (Graf, 2010a,b; Heinz, 2007, 2009, 2010a; Heinz et al., 2011; Rogers et al., 2013). The second purpose of this chapter is to show that these local grammars over strings are not sufficiently powerful for tonal phonology, thus motivating the local autosegmental grammars argued for in the rest of the dissertation. This move to keep local grammars but consider more complex structures is justified by the fact, illustrated in this chapter, that it is more restrictive to enrich the representation than increase the power of the logical grammar.

This chapter introduces the formal background necessary for understanding a view of phonological constraints based in formal logic and formal language theory. In this view, grammars are *logical* constraints over surface structures that are *inviolable* and *language-specific*. These logical constraints specify *formal languages* (Büchi, 1960; Rogers, 1998; Rogers et al., 2013; Thomas, 1982) which can be used to model the sets of well-formed strings of phonemes (or other units) in natural language patterns (Graf, 2010a,b; Heinz, 2007, 2009, 2010a; Heinz et al., 2011; Rogers et al., 2013). It will be shown here that the most restrictive kind of logical constraint are the *banned substructure constraints*, which shall be argued for throughout this dissertation as a restrictive theory of language-specific well-formedness constraints on phonological structures.

To give an example from tone in the Japanese dialects, Kagoshima Japanese words have exactly one high pitched syllable, either on the final or the penultimate syllable (Hirayama, 1951; Haraguchi, 1977; Kubozono, 2012). Thus, LLHL and LLLH are well-formed strings of syllables in Kagoshima (respective examples are [kagaríbi] 'watch fire' and [irogamí] 'colored paper' (Haraguchi, 1977)), but *LHLL is not. That this well-formedness constraint is psychologically real for Kagoshima speakers can be seen in the fact that it is applied to novel forms in the dialect, for example the recent English borrowing [makudonarúdo] 'McDonald's' (Kubozono, 2012).

We can model part of the competence native speakers have regarding the well-formedness of tone patterns in Kagoshima Japanese with a logical constraint ¬HLL, which can be interpreted as saying "A H-toned syllable cannot be followed by two low-tone syllables." More formally, this constraint picks out a set of strings which *satisfy* this constraint by not containing the structure HLL. Thus, the tone pattern LLLLHL (i.e., that for 'McDonald's') is a member of this set, because it does not contain the sequence HLL. In contrast, the string *LLHLLL (a possible approximation of the English tone pattern for 'McDonald's) is not a member of this set, because it contains a HLL sequence. The logical constraint ¬HLL is a banned substructure constraint because it specifies an invalid part of a string.

The advantages to logical constraints are numerous. First, they are *explicit* in that their interpretations are well defined. Second, their *expressive power* is similarly well-defined, and different types of logic with different levels of expressiveness can be used as hypotheses for natural language phonology. Third, through Grammatical Inference (GI; de la Higuera, 2010), we are given a theory of learning the formal languages described by (the most restrictive of these) logics. It will be shown how this can be extended to the logical constraints proposed in this dissertation in Chapter 8. Most importantly for the present work, however, is that we can vary *structure* while keeping logical languages constant. This will be shown in this chapter for different string structures, which will be shown to have different levels of expressiveness given a fixed logic, and will become more relevant in the next chapters when we move from strings to APRs.

It should be noted that the logical constraints defined here can only refer to sets of structures, and so most readily model *surface* constraints on well-formedness. Thus, for most of the dissertation, we shall not consider phonological *transformations* from underlying representations to surface representations. However, Chapter 7 will show how, by defining a set of structures which model transformations, such constraints can be used in a theory of phonological transformations which incorporates the surface-based well-formedness constraints outlined in this and the following chapters.

This chapter is structured as follows. §3.1 introduces the mathematical basics for the discussion in this and following chapters, including formally defining strings and formal languages. §3.2 discusses how formal languages can be related to well-formedness in natural language phonology. §3.3 introduces the idea of logical grammars to describe formal languages, and then §3.4 relates different kinds of logics, their expressiveness, and how varying logic and structure can offer different hypotheses for well-formedness in phonology. §3.6 then shows how the logic over strings with the current best fit to segmental phonology is inadequate for tone, prompting the move to logic over graphs in the rest of the dissertation. §3.7 concludes.

3.1 Mathematical Preliminaries I: Sets and Languages

We begin with some of the mathematical background necessary for explicitly defining the concepts in this and later chapters.

3.1.1 Sets, pairs, and tuples

First is basic set notation. For an excellent introduction to set theory written for linguists, see Partee et al. (1993). A *set* is an unordered collection of distinct elements, and will be denoted by elements surrounded with curly brackets $\{\}$. For example $\{1, 2, 3\}$ contains the elements 1, 2, and 3, and as it is unordered it may also be written $\{3, 2, 1\}$, $\{1, 3, 2\}$, etc. Because its elements are distinct, $\{1, 1, 1, 2, 3\} = \{1, 2, 3\}$—that is, it means nothing to refer an element in a set more than once. *Membership* in a set is denoted with $\in$; e.g. $1 \in \{1, 2, 3\}$. The terms 'member' and 'element' will be used interchangeably. The *cardinality* of a set is the number of elements in the set; the cardinality of $\{1, 2, 3\}$ is 3. A set R is a *subset* of S iff all of its elements are also members of S, written $R \subseteq S$. For example, $\{1, 2\} \subseteq \{1, 2, 3\}$. Note also that $\{1, 2, 3\} \subseteq \{1, 2, 3\}$. We may write $R \subset S$ if $R \subseteq S$ and $R \neq S$; R is then a *proper subset* of S. Let the *set-theoretic difference* of two sets S and R, denoted $S - R$, be $\{x \in S | x \notin R\}$ (all elements x in S which are not in R). $\{1, 2, 3\} - \{2, 3\}$ is thus $\{1\}$. The *empty set*, or the set with no members can be denoted either as $\{\}$ or $\emptyset$. The cardinality of the empty set is 0.

Lists of *ordered* elements will be denoted as follows. An *ordered pair* will be written with parentheses (), for example $(1, 2)$. Importantly, because it is ordered, $(1, 2) \neq (2, 1)$. Given two sets S and T, $S \times T$ is the *cross product* of S and T, or the set of all possible pairs (x, y) where $x \in S$ and $y \in T$. (S and T need not be distinct sets.) For example, for $\{1, 2, 3\}$ and $\{1, 2\}$, $\{1, 2, 3\} \times \{1, 2\} = \{(1, 1), (1, 2), (2, 1), (2, 2), (3, 1), (3, 2)\}$. We call any subset $R \subseteq S \times T$ a *relation* between elements in S and elements in T.

For more than two ordered elements, we use a *tuple*, written with angled brackets $\langle\rangle$. For example, $\langle 1, 2, 3\rangle$ is a *three*-tuple, and again $\langle 1, 2, 3\rangle \neq \langle 3, 2, 1\rangle$. Also, in tuples and ordered pairs, distinct elements may be listed more than once; $\langle 1, 1, 2, 3\rangle \neq \langle 1, 2, 3\rangle$.

3.1.2 Alphabets, strings and formal languages

An *alphabet* is a set of symbols. Often for abstract examples I'll use early lowercase Roman letters for these symbols; a, b, c, etc. Alphabets are usually denoted with uppercase Greek letters, most commonly Σ; e.g. $\Sigma = \{a, b, c\}$. Lowercase Greek letters will be arbitrarily reference a symbol from an alphabet Σ; i.e., 'some $\sigma \in \Sigma$'.

Given an alphabet Σ, a *string* over Σ is a finite sequence of symbols in Σ. Formally defining the concept of a *string* over Σ requires two concepts. One is the *empty string*. The empty string, denoted λ, is the string containing no symbols. Like the number 0, it's an abstract notion, but it forms the starting point for defining strings. The other concept is *concatenation*. For a string w and a symbol σ let $w \bullet \sigma$

or more concisely $w\sigma$ denote the concatenation of σ to the end of w. Importantly, for some $\sigma \in \Sigma$, $\lambda \bullet \sigma = \sigma$. This means that σ may refer to a symbol in Σ, or it may refer to a string of length 1. We can safely ignore this technicality.

We can thus define a string recursively as the concatenation of some succession of symbols from Σ. The set of all strings over Σ is denoted Σ^*.

Definition 1 *Given an alphabet Σ, $w \in \Sigma^*$ iff:*

- *$w = \lambda$, or*
- *$w = u \bullet \sigma$; $\sigma \in \Sigma$, $u \in \Sigma^*$*

Nothing else is in Σ^*.

Intuitively, this definition says that all strings over Σ are either λ or the concatenation of some $\sigma \in \Sigma$ to a string over Σ. For example, given $\Sigma = \{a, b\}$, $abb \in \Sigma^*$. We can show this by a series of steps which follow Definition 1. First, $abb = ab \bullet b$, and $ab \in \Sigma^*$ because $ab = a \bullet b$, and $a \in \Sigma^*$ because $a = \lambda \bullet a$, and $\lambda \in \Sigma^*$. The reader can confirm that any string of as and bs (or any string of any symbols from a particular alphabet) can be generated this way.

Concatenation may be extended to strings. Let $w \bullet v$, or more concisely wv, denote the concatenation of two strings. We can then define a *substring*, or contiguous piece of a string. For a string w, v is a substring of w if $w = u_1 v u_2$, for two strings u_1 and u_2. For example, aba is a substring of $abbb\boldsymbol{aba}a$ (substring in bold). Because the empty string is a string, every string is a substring of itself: aba is also a substring of $\lambda aba \lambda = aba$.

For a symbol σ, we write σ^n for a string composed of σ repeated n times. For example, $a^5 = aaaaa$.

A *formal language* (or just *language*, or sometimes *stringset*) $L \subseteq \Sigma^*$ is a set of strings. This set may be infinite, or it may not. Let $\Sigma = \{a, b\}$ again. If $L = \{aa, bb\}$, then L is a language in Σ^*. Note that in this case, L is finite; it has only two members. If $L = \{ab, abab, ababab, abababab, ...\}$, i.e. the set of arbitrary repetitions of ab, then L is also a language in Σ^*. This L is infinite (as indicated by the ellipsis); it has an infinite number of members. Concatenation may also be extended to languages; $L_1 \bullet L_2$ denotes the pairwise concatenation of each string in L_1 with each string in L_2.

With the concatenation of languages, we can define an important special version of Σ^*. Modelling generalizations in natural language phonology, it is often necessary to refer to the beginnings and the ends of words. Let $\rtimes$ and $\ltimes$ be two symbols not in Σ. Let $\rtimes\Sigma^*\ltimes = \{\rtimes\}\bullet\Sigma^*\bullet\{\ltimes\}$ be the set of all words in Σ^* delineated with these beginning and end boundaries. Thus, for any string $w \in \Sigma^*$, $\rtimes w \ltimes \in \rtimes\Sigma^*\ltimes$. For example, if $\Sigma = \{a, b\}$, $aabb \in \Sigma^*$, so $\rtimes aaba \ltimes \in \rtimes\Sigma^*\ltimes$.

This dissertation is primarily interested in infinite languages like this latter example, as shall be discussed further in a moment. However, infinite languages pose a problem—how to represent them in a finite space. Let us define *grammar* any finite representation of a (possibly infinite) formal language. Given a grammar G, let $L(G)$ denote the language represented by (or 'the language of' or 'the language described by') G.

This is a very general definition of the notion of grammar, and there are many different types of grammars, of which the logical constraints used throughout this dissertation are only one example. The following discusses formal languages, grammars, and their relation to phonology.

3.2 Well-formedness Constraints as Formal Languages

A formal language can model phonotactic knowledge of native speakers by representing the set of well-formed words in their language. The infinitude of such a formal language can represent speakers' ability to productively apply well-formedness constraints to new forms. The introduction gave an example of this in Kagoshima Japanese. Let us model this more explicitly with a formal language.

Let $\Sigma = \{\mathrm{H}, \mathrm{L}\}$, representing high- and low-toned syllables respectively. As mentioned in the introduction, in order for a word to be well-formed in Kagoshima Japanese, it must have a single high tone either on the ultimate or penultimate syllable. We can represent this constraint by bifurcating Σ^* into L_{KJ}, the set of strings of syllables which constitute well-formed words in Kagoshima Japanese, and everything else. Examples of these two sets are given below in (3.1).

(3.1)

In L_{KJ}	Not in L_{KJ}
H, HL, LH,	L, LL, HH, LLL,
LHL, LLH,	HLL, HHL, HHH,
LLHL, LLLH,	LHH, LLLL, HLLL,
..., LLLLHL,	..., HLLHLHL,
LLLLLH, ...	LHLHLH, ...

For example, LLLLHL (the tone pattern for [makudonarúdo] 'McDonald's') is in L_{KJ}, but HLLHLHL is not, because it has more than one H tone. Note that the ellipses indicate that L_{KJ} is an infinite language. Thus, for example, LLLLLLLHL is in L_{KJ}, even if a word of such length may not exist in the vocabulary of any particular speaker of the dialect. However, this models the fact that they *would* find such a word well-formed, if it were created through morphological processes or borrowing (such as the 6-syllable 'McDonald's'). However, the infinitude of L_{KJ} also means that strings like $\mathrm{L}^{20}\mathrm{HL}$ are in the language, even though they are unlikely to ever be attested in actual Kagoshima Japanese use. The critical assumption here is that this particular well-formedness constraint holds *independently* on any constraints on the lengths of words, be them well-formedness constraints or some sort of functional constraint (such as the impracticality of extremely long words).

If L_{KJ} is infinite, how could it possibly model the competence of actual speakers of Kagoshima Japanese, whose brains (like all human brains) are finite? The answer was already hinted at in §3.1.2—an infinite formal language can be modeled with a *finite* formal grammar. This is the topic of the next section.

3.3 Logical Grammars

We can represent an infinite formal language $L \subseteq \Sigma^*$ finitely by defining a principled way of checking any string $w \in \Sigma^*$ to see if w is in L. Recall the concept of a *substring*, or a contiguous piece of a string. We can define a *logical language* (or just *logic*) of statements which are evaluated depending on the substrings in a string. As any such statement is finite, it can be interpreted as a *grammar* specifying the set of all strings in Σ^* for which the statement is true. The following is a brief overview of how to define such a logic; for a thorough introduction to such definitions, the reader is referred to Enderton (1972).

As an introduction to the relationship between formal languages and logic, this section is primarily concerned with *propositional logic*. While the following section will show propositional logic to be an undesirable hypothesis for phonological well-formedness, it does provides a good starting point for understanding how logical statements can provide an explicit description of a pattern and will also play a key role in the subsequent discussion of the expressivity of logical languages. It will also serve as a basis for, in the next section, defining the more restrictive logics that give us the notion of locality that will be used throughout this dissertation.

3.3.1 The syntax and semantics of a propositional logic

Defining a logical language comes in two steps. One is defining its *syntax*, or defining the set of sentences in the logical language. The second is defining its *semantics*, which define how each sentence is interpreted. The syntax of our logic based on substrings is as follows in Defintion 2. This logic is a *propositional* logic, meaning it is made up of statements which can be true or false about strings, and so let us denote it as $\mathfrak{L}^P$(Enderton, 1972).

Definition 2 (Syntax of $\mathfrak{L}^P$) *Given a sentence ϕ,*

a. if $\phi = u$ for a substring u of some word $w \in \rtimes\Sigma^*\ltimes$, then $\phi \in \mathfrak{L}^P$

b. if $\phi = (\neg\psi)$ for some $\psi \in \mathfrak{L}^P$, then $\phi \in \mathfrak{L}^P$

c. if $\phi = (\psi_1 \wedge \psi_2)$ for some $\psi_1, \psi_2 \in \mathfrak{L}^P$, then $\phi \in \mathfrak{L}^P$

d. if $\phi = (\psi_1 \vee \psi_2)$ for some $\psi_1, \psi_2 \in \mathfrak{L}^P$, then $\phi \in \mathfrak{L}^P$

e. if $\phi = (\psi_1 \rightarrow \psi_2)$ for some $\psi_1, \psi_2 \in \mathfrak{L}^P$, then $\phi \in \mathfrak{L}^P$

Nothing else is in $\mathfrak{L}^P$.

This simply defines what statements in $\mathfrak{L}^P$ are going to look like. The basic propositions are substrings of $\rtimes\Sigma^*\ltimes$. Why we use $\rtimes\Sigma^*\ltimes$ delineated with the boundary symbols, instead of just Σ^*, will become clear in a moment. The other propositions are defined as linking other propositions via the logical connectives $\neg$, $\wedge$, $\vee$, and $\rightarrow$. Before defining what these mean (though the reader may already have some idea) let us look at a few examples of statements in $\mathfrak{L}^P$.

Let $\Sigma = \{a, b\}$. By Definition 2a any substring of a string in $\rtimes\Sigma^*\ltimes$ is in $\mathfrak{L}^P$. Because $\rtimes aabba\ltimes \in \rtimes\Sigma^*\ltimes$, $a\ltimes$ is a statement in $\mathfrak{L}^P$, aa is a statement in $\mathfrak{L}^P$, ba is a statement in $\mathfrak{L}^P$, $abba$ is a statement in $\mathfrak{L}^P$, etc. We can then build up statements recursively using the definitions of connective statements in Def. 2b through e. For example, because $a\ltimes$ and ba are statements in $\mathfrak{L}^P$, then by Def. 2b $(\neg ba)$ and $(\neg a\ltimes)$ are also in $\mathfrak{L}^P$. We can keep building further. Because $(\neg ba)$ and $(\neg a\ltimes)$ are in $\mathfrak{L}^P$, then by Def. 2c so is $((\neg ba) \wedge (\neg a\ltimes))$. The parentheses here make it explicit how a statement is 'built' out of recursive applications of Def 2, but they are not always necessary and can be visually confusing. Thus, I will omit them when the context is clear; for instance, $((\neg ba) \wedge (\neg a\ltimes))$ may also be written $\neg ba \wedge \neg a\ltimes$.

Again, while the meanings of these logical connectives are perhaps well-known, we have not explicitly defined them. The following defines the semantics of $\mathfrak{L}^P$, by stating under which conditions a string in Σ^* *satisfies* a statement ϕ in $\mathfrak{L}^P$.[1]

Definition 3 (Semantics of $\mathfrak{L}^P$) *Given a sentence $\phi \in \mathfrak{L}^P$ and a word w, $w \models \phi$ (w* satisfies *ϕ):*

a. if $\phi = u$ and u is a substring of $\rtimes w\ltimes$

b. if $\phi = (\neg\psi)$ for some $\psi \in \mathfrak{L}^P$ and $w \not\models \psi$ (w does not *satisfy ψ)*

[1] Technically, the satisfaction of a logical statement is usually defined through a *model* of a string, which represents particular kinds of information in the string, rather than through the string itself. However, given the restricted logics considered in this definition, it is enough to talk about the strings themselves satisfying logical statements. For more on the relationship between models and logic, the reader is referred to (Enderton, 1972).

c. if $\phi = (\psi_1 \wedge \psi_2)$ *for some* $\psi_1, \psi_2 \in \mathfrak{L}^P$ *and* $w \models \psi_1$ *and* $w \models \psi_2$

d. if $\phi = (\psi_1 \vee \psi_2)$ *for some* $\psi_1, \psi_2 \in \mathfrak{L}^P$ *and* $w \models \psi_1$ *or* $w \models \psi_2$

e. if $\phi = (\psi_1 \rightarrow \psi_2)$ *for some* $\psi_1, \psi_2 \in \mathfrak{L}^P$ *and* $w \models (\neg\psi_1 \vee \psi_2)$

The reader can confirm that, as the structure of Def. 3 follows that of Def. 2, we have a way of interpreting how *any* string in Σ^* can satisfy *any* statement in $\mathfrak{L}^P$(I will omit the proof). Let us look at some examples.

Recall from above that $a\ltimes$ and ba, by virtue of being substrings of strings in $\rtimes\Sigma^*\ltimes$, are statements in $\mathfrak{L}^P$. To avoid confusion with their string counterparts, let us use $\phi_{a\ltimes}$ to refer to the logical statement $a\ltimes$, and let us use ϕ_{ba} to refer to the logical statement ba. By Def. 3a, any $w \in \Sigma^*$ which, when delineated with $\rtimes$ and $\ltimes$, contains $a\ltimes$ as a substring, satisfies $\phi_{a\ltimes}$, and likewise with ba and ϕ_{ba}. For example, $aaa \models \phi_{a\ltimes}$, because $a\ltimes$ is a substring of $\rtimes aaa\ltimes$. Likewise, $bbbaaaa \models \phi_{ba}$, because ba is a substring of $\rtimes bbbaaaa\ltimes$. On the other hand, $aab \not\models \phi_{a\ltimes}$ and $aab \not\models \phi_{ba}$, because neither $a\ltimes$ nor ba is a substring of $\rtimes aab\ltimes$. Now consider $\neg\phi_{a\ltimes}$, the negation of $\phi_{a\ltimes}$. Because $aab \not\models \phi_{a\ltimes}$, by Def. 3b, $aab \models \neg\phi_{a\ltimes}$. Likewise, because $aab \not\models \phi_{ba}$, $aab \models \neg\phi_{ba}$. Then, because $aab \models \neg\phi_{a\ltimes}$ and $aab \models \neg\phi_{ba}$, by Def. 3c, $aab \models \neg\phi_{a\ltimes} \wedge \neg\phi_{ba}$. It is in this way that Def. 3 allows us to recursively build up an interpretation of a statement in $\mathfrak{L}^P$ using the interpretation of its composite parts.

There may be many strings which satisfy a statement in $\mathfrak{L}^P$. Each statement ϕ in $\mathfrak{L}^P$ thus bifurcates the set Σ^* into two sets: the set of strings which satisfy ϕ, and the set of strings that do not. The sets of strings in Σ^* which satisfy $\phi_{a\ltimes}$, $\neg\phi_{a\ltimes}$, $\neg\phi_{ba}$, and $\neg\phi_{a\ltimes} \wedge \neg\phi_{ba}$, as well as the sets of strings in Σ^* which do *not* satisfy each statement, are given below in (3.2).

(3.2)

ϕ	Strings satisfying ϕ	Strings not satisfying ϕ
$\phi_{a\ltimes}$	$a, aa, ba, aaa, aba,$	$b, ab, bb, aab, abb, bab,$
	$baa, bba, aaaa, \ldots$	$bbb, aaab, aabb, \ldots$
$\neg\phi_{a\ltimes}$	$b, ab, bb, aab, abb, bab,$	$a, aa, ba, aaa, aba,$
	$bbb, aaab, aabb, \ldots$	$baa, bba, aaaa, \ldots$
$\neg\phi_{ba}$	$a, b, aa, ab, bb, aaa, aab,$	$ba, aba, baa, bba, bab,$
	$abb, bbb, aaaa, aaab, \ldots$	$aaba, abaa, abba, baaa, \ldots$
$\neg\phi_{a\ltimes} \wedge \neg\phi_{ba}$	$b, ab, bb, aab, abb, bbb,$	$a, aa, ba, aaa, aba, baa, bab,$
	$aaab, aabb, abbb, \ldots$	$bba, aaaa, aaba, abaa, \ldots$

The strings that satisfy $\phi_{a\ltimes}$ are exactly those strings which end in a; the strings that satisfy $\neg\phi_{a\ltimes}$ are those which do not end in a. Likewise, the strings that satisfy $\neg\phi_{ba}$ are those in which all as precede all bs. The set of strings which satisfy $\neg\phi_{a\ltimes} \wedge \neg\phi_{ba}$ is the intersection of these two sets; i.e., the strings which satisfy both $\neg\phi_{a\ltimes}$ and $\neg\phi_{ba}$.

In this way, any $\phi \in \mathfrak{L}^P$ can be seen as specifying a formal language; i.e., the set of strings which satisfy ϕ. Thus, any such ϕ is a grammar as defined above in §3.1.2. Parallel to the notation given there, we denote the language specified by a statement ϕ as $L(\phi)$. Formally,

Definition 4 (The language of ϕ) *For any $\phi \in \mathfrak{L}^P$, $L(\phi) = \{w \in \Sigma^* | w \models \phi\}$ (all strings $w \in \Sigma^*$ that satisfy ϕ).*

Thus, for example, $L(\neg\phi_{a\ltimes})$ is the set of strings satisfying $\neg\phi_{a\ltimes}$ in (3.2). We now have a concrete way of finitely representing an infinite formal language. Let us return to how this may be applied to well-formedness constraints in natural language.

3.3.2 A logical grammar for Kagoshima Japanese

In §3.2 we saw the tone pattern of Kagoshima Japanese, in which all words have either a penultimate or final H tone, modelled as a formal language L_{KJ}. This formal language is repeated below in (3.3) from (3.1).

(3.3)

In L_{KJ}	Not in L_{KJ}
H, HL, LH,	L, LL, HH, LLL,
LHL, LLH,	HLL, HHL, HHH,
LLHL, LLLH,	LHH, LLLL, HLLL,
..., LLLLHL,	..., HLLHLHL,
LLLLLH, ...	LHLHLH, ...

We can now represent this infinite language with a finite, logical grammar. The following shows that this can be done by stating a series of substrings which are not allowed to appear in any string in L_{KJ}. This is not the only way to describe L_{KJ} with a logical statement, but why this particular method is used will become clear in §3.4.

First, we note that no strings in L_{KJ} contain the substring HLL, as this represents a H tone that is farther from the right edge than the penult. For example, HLL is a substring of the ill-formed strings ***HLL**L and ***HLL**LL. Similarly, no strings in L_{KJ} contain HLH either, as this would also create a pre-penult H (e.g. *HLL**HLH**L in (3.3)). Statements picking out these two substrings are given the names ϕ_{HLL} and ϕ_{HLH} below in (3.4).

(3.4) a. $\phi_{HLL} = \text{HLL}$ b. $\phi_{HLH} = \text{HLH}$

Additionally, there are no adjacent Hs in any string in L_{KJ}, as no two Hs are allowed to appear in a word in Kagoshima Japanese. The following statement picks out exactly such a structure with a HH substring.

(3.5) $\phi_{HH} = \text{HH}$

Finally, as all strings in L_{KJ} have a H either on the penult or the final syllable, we never see a final LL substring. The logical statement referring to a final LL substring is given in (3.6).

(3.6) $\phi_{LL\ltimes} = \text{LL}\ltimes$

To ban each of these strings, we can simply negate each of the statements in (3.4) through (3.6) and then take the conjunction of these negations. Call this statement ϕ_{KJ}.

(3.7) $\phi_{KJ} = \neg\phi_{HLL} \wedge \neg\phi_{HLH} \wedge \neg\phi_{HH} \wedge \neg\phi_{LL\ltimes}$

The following remark claims that ϕ_{KJ} describes L_{KJ}.

Remark 1 $L(\phi_{KJ}) = L_{KJ}$

Proof: We show that $L_{KJ} \subseteq L(\phi_{KJ})$ and then that $L(\phi_{KJ}) \subseteq L_{KJ}$. First, for $L_{KJ} \subseteq L(\phi_{KJ})$, any strings $w \in L_{KJ}$ will contain exactly one H in either ultimate or penultimate position. Thus, when delineated with $\rtimes$ and $\ltimes$, cannot contain either HLL, HLH, HH, nor LL$\ltimes$ as substrings, and will thus satisfy ϕ_{KJ} and so $w \in L(\phi_{KJ})$.

For $L_{KJ} \subseteq L(\phi_{KJ})$, any $w \in L(\phi_{KJ})$ will have at least one H in either ultimate or penultimate position, because $w \models \neg\phi_{LL\ltimes}$. It also contains exactly one H, as implied by the fact that $w \models \neg\phi_{HH}$, $w \models \neg\phi_{HLH}$, and $w \models \neg\phi_{HLL}$ (the latter banning Hs separated by two or more Ls). Thus, w has exactly one H in penultimate or ultimate position, and so $w \in L_{KJ}$. $\square$

Thus, we have successfuly described the infinite L_{KJ} generalization with a finite logical grammar.

3.3.3 Interim conclusion: logical grammars

To summarize what has been covered so far, we have seen how to model an infinite generalization with a finite grammar. Additionally, in $\mathfrak{L}^P$ we have seen how a class of logical grammars is mathematically explicit. That is, we know all of the possible grammars (=statements) in $\mathfrak{L}^P$, and we have a precise method for interpreting each one.

Another important strength of logical grammars is that we get an understanding of the *expressivity* of different types of logical grammars. This important, as $\mathfrak{L}^P$ is expressive enough to describe patterns that are unattested in phonology, and is thus a bad hypothesis for phonological well-formedness.

3.4 Local Logics

The previous section showed how it was possible to *describe* a phonological pattern in $\mathfrak{L}^P$. How, then, does it relate to a predictive theory of phonological well-formedness? Recall that in the previous chapter, a criticism of previous explanations of tonal phonology in Optimality Theory was that it both undergenerated (in terms of not being able to capture N. Karanga Shona) and overgenerated (through the predictions of ALIGN). In order to provide a better theory of tonal well-formedness, we must thus find a logical language that is both *sufficient* in describing the attested typology of tonal patterns, and *restrictive* in the sense that it does not predict bizarre patterns.

The following illustrates how $\mathfrak{L}^P$ also can describe patterns that are unattested in natural language phonology, due to the *global* evaluation of its statements. This motivates introducing restrictions on $\mathfrak{L}^P$ which are fundamentally *local* in nature. These are $\mathfrak{L}^{NL}$, which is a subset of $\mathfrak{L}^P$, and $\mathfrak{L}^{NL}_T$ introduced in §3.5, which has the same syntax of $\mathfrak{L}^{NL}$ but is interpreted over different structures. It will be then shown how these logics form restrictive *hypotheses* for surface well-formedness constraints in phonology.

3.4.1 An unattested pattern

Let $\Sigma = \{\mathrm{H}, \mathrm{L}\}$ and consider the following statement in $\mathfrak{L}^P$:

(3.8) $\phi_{IH,FH} = \rtimes H \rightarrow H \ltimes$

I name $\phi_{IH,FH}$ as such because it requires of strings in $L(\phi_{IH,FH})$ that if there is an *I̲nitial* *H̲* ($\rtimes$H), then there is also a *F̲inal* *H̲* (H$\ltimes$). In other words, if the string begins with an H, the end of the string must 'agree' and also be H.[2] The reader can confirm that $\phi_{IH,FH} \in \mathfrak{L}^P$ as it can be built using Def. 2. The set of strings in $L(\phi_{IH,FH})$ are given below in (3.9).

[2] This pattern is based on the unattested First-Last Assimilation pattern discussed by Lai (2015).

(3.9)

In $L(\phi_{IH,FH})$	Not in $L(\phi_{IH,FH})$
L, H, LL, HH, LLLL,	HL, HHL, HLLL, HLHL,
HLLH, HLHH,	HHHL, HLLLLLL, ...
HLLLLH,...	

For example, HLLH $\in L(\phi_{IH,FH})$, because it satisfies $\rtimes$H and it also satisfies H$\ltimes$. In contrast, HLLL $\notin L(\phi_{IH,FH})$, because it satisfies $\rtimes$H but not H$\ltimes$. Note that any string in Σ^* that doesn't begin with H, e.g., LLLL, is in $L(\phi_{IH,FH})$, because $\phi_{IH,FH}$ only says anything about strings that begin with H (as an implication is always true if its antecedent is false). Thought about in these terms, the set of strings *not* in $L(\phi_{IH,FH})$ is exactly the set of strings that begin with H but do not also end in H.

To the best of my knowledge, no tonal pattern requires that the last TBU of the word must agree with the first TBU just in case it is H-toned. However, if we take statements in $\mathfrak{L}^P$ to be our hypothesis for phonological well-formedness, then we predict such a pattern to exist. Thus, $\mathfrak{L}^P$ *overgenerates* an unattested pattern. This can be said to be because of its *global* nature: $\phi_{IH,FH}$ establishes a dependency on TBUs on either end of the word.

In contrast, the statement ϕ_{KJ} described the attested L_{KJ} pattern by specifying a set of substrings (namely, HLL, HLH, HH, and LL$\ltimes$) which do not appear in any string in the language. The following shows that such statements are fundamentally *local*, in a well-defined way, and that we can build a restricted logic out of this kind of statement that discludes patterns like $L(\phi_{IH,FH})$.

3.4.2 Conjunctions of Negative Literals

The statement ϕ_{KJ} belongs to a particular subclass of statements in $\mathfrak{L}^P$ with some interesting properties. This class is called the *conjunctions of negative literals*, which I'll denote as $\mathfrak{L}^{NL}$. *Literal* refers to the most basic kind of propositional statement, in this case statements representing the presence of a single substring, as defined in 2a. A formal definition of $\mathfrak{L}^{NL}$ is below.

Definition 5 (Conjunction of negative literals) *Given a propositional logic $\mathfrak{L}^P$, a statement $\phi \in \mathfrak{L}^P$ is a* conjunction of negative literals *if it has the following structure:*

$$\phi = \neg w_0 \wedge \neg w_1 \wedge \neg w_2 \wedge \ldots \wedge \neg w_n$$

where $w_0, w_1, w_2, ..., w_n$ are literals*, or strings meeting Def. 2a of a statement in $\mathfrak{L}^P$.*

Let $\mathfrak{L}^{NL}$ denote the set of conjunctions of negative literals in $\mathfrak{L}^P$.

As ϕ_{KJ} does, a conjunction of negative literals specifies a list of *banned substrings* which will not appear in any string in the corresponding language. We shall generalize this concept to banned sub*structure* constraints when moving from strings to graphs in the following chapters.

One last thing to keep in mind about statements in $\mathfrak{L}^{NL}$ is that because each statement is finite, there is some longest substring which it bans. For example, in ϕ_{KJ}, the longest substrings it bans (HLL, HLH, LL$\ltimes$) are of length 3. For sets characterizable by banned substructure constraints, it will sometimes be important to remember the size of the largest substructure being banned. Following tradition, I will refer to this value with k. Thus, for ϕ_{KJ}, $k = 3$.

3.4.3 Comparing logics

The reason that the idea of banned substructures is of interest is that logics which can *require* substructures are more powerful than conjunctions of negative literals, which can only ban substructures. In this way, $\mathfrak{L}^{NL}$ is a very *restrictive* set of grammars, which will be argued momentarily to be an important quality of $\mathfrak{L}^{NL}$ as a hypothesis for well-formedness constraints in phonology.

First, note that by Definitions 2 and 5, $\mathfrak{L}^{NL}$ is a strict subset of $\mathfrak{L}^P$.

Theorem 1 $\mathfrak{L}^{NL} \subset \mathfrak{L}^P$

Proof: By Defs. 2 and 5, every statement in $\mathfrak{L}^{NL}$ is in $\mathfrak{L}^P$, but there are statements in $\mathfrak{L}^P$ that are not in $\mathfrak{L}^{NL}$. The statement $\phi_{IH,FH}$ in (3.8), for example, is in $\mathfrak{L}^P$ but does not match Def. 5 for a statement in $\mathfrak{L}^{NL}$. $\square$

The statement $\phi_{IH,FH}$ also demonstrates that $\mathfrak{L}^{NL}$ is *strictly less expressive* than $\mathfrak{L}^{P}$; that is, there are languages we can describe with $\mathfrak{L}^{P}$ that we cannot describe with $\mathfrak{L}^{NL}$. The language $L(\phi_{IH,HF})$ is one such language. I omit the proof, but the intuition is as follows.[3] We need to ban all strings of the shape HL^n, as no such string satisfies $\phi_{IH,FH}$. We cannot ban $\rtimes$H, as this would incorrectly also ban strings of the shape HL^{n}H, which *are* in $L(\phi_{IH,HF})$. We also cannot simply ban L$\ltimes$ as a substring, as that would incorrectly remove strings like LLL from $L(\phi_{IH,HF})$. We can then try and ban entire strings, such as $\rtimes$HL$\ltimes$, $\rtimes$HLL$\ltimes$, etc. However, we would need an *infinite* number of such statements, as for every negative literal $\neg \rtimes \text{HL}^n \ltimes$ for some n, there is some string HL$^{n+1} \notin L(\phi_{IH,FH})$ that would satisfy that negative literal. In this way, any statement in $\mathfrak{L}^{NL}$ is bound to fail to describe $L(\phi_{IH,FH})$. Again, this is because $\phi_{IH,FH}$ *requires* a H$\ltimes$ substring just in case a $\rtimes$H substring is also present.

3.4.4 $\mathfrak{L}^{P}$ and $\mathfrak{L}^{NL}$ and formal and natural languages

Thus, we have shown that $\mathfrak{L}^{NL}$ is less expressive than $\mathfrak{L}^{P}$. In fact, this is well-known. The languages describable by $\mathfrak{L}^{NL}$ are the *Strictly Local* (SL) languages (also known as the languages *Locally Testable in the Strict Sense*; McNaughton and Papert, 1971). The languages describable by $\mathfrak{L}^{P}$ are the *Locally Testable* (LT) languages, which are a strict superset of the SL languages (McNaughton and Papert, 1971).

These classes are very *natural* in that the logical characterizations in $\mathfrak{L}^{P}$ and $\mathfrak{L}^{NL}$ are not the only way of obtaining them. For example, the SL languages are describable by tiling grammars (Rogers et al., 2013) and finite-state acceptors whose

[3] Proving a pattern is not in $\mathfrak{L}^{NL}$ uses an abstract property shared by all patterns describable in $\mathfrak{L}^{NL}$, which is that of *suffix substitution closure* (Rogers and Pullum, 2011). This is a property of the patterns themselves and not dependent on the particular grammar formalism (such as the logical grammars outlined here). Essentially, a pattern has suffix substitution closure if there is some value k for which a string's membership in the pattern will not depend on information in the string that is separated by k or more symbols.

states correspond to substrings of size $k-1$ for some k (McNaughton and Papert, 1971).

The SL languages also have a few more properties that make them interesting. One, as pointed out by Rogers et al. (2013), the cognitive interpretation of evaluating the well-formedness of a string with respect to an SL language is straightforward. Recall $L(\phi_{KJ})$, for whom the value k of the longest banned substring was 3. For any strings in Σ^*, whether or not it is a member of $L(\phi_{KJ})$ can be checked by scanning the string with a window of size 3. If any banned substring appears in the window, we know it's out. Contrast the step-by-step evaluations of LLHL and *LHLL below in (3.10). ϕ_{kj} is repeated below in (3.10a) as reference for the banned substrings.

(3.10) a. $\phi_{KJ} = \neg\phi_{HLL} \wedge \neg\phi_{HLH} \wedge \neg\phi_{HH} \wedge \neg\phi_{LL\ltimes}$

b. $\underset{\checkmark\ \rtimes\mathrm{LL}}{\boxed{\rtimes\ \mathrm{L}\ \mathrm{L}}}\ \mathrm{H}\ \mathrm{L}\ \ltimes \qquad \rtimes \underset{\checkmark\ \mathrm{LLH}}{\boxed{\mathrm{L}\ \mathrm{L}\ \mathrm{H}}}\ \mathrm{L}\ \ltimes \qquad \rtimes\ \mathrm{L} \underset{\checkmark\ \mathrm{LHL}}{\boxed{\mathrm{L}\ \mathrm{H}\ \mathrm{L}}} \ltimes \qquad \rtimes\ \mathrm{L}\ \mathrm{L} \underset{\checkmark\ \mathrm{HL}\ltimes}{\boxed{\mathrm{H}\ \mathrm{L}\ \ltimes}}$

c. $\underset{\checkmark\ \rtimes\mathrm{LH}}{\boxed{\rtimes\ \mathrm{L}\ \mathrm{H}}}\ \mathrm{L}\ \mathrm{L}\ \ltimes \qquad \rtimes \underset{\checkmark\ \mathrm{LHL}}{\boxed{\mathrm{L}\ \mathrm{H}\ \mathrm{L}}}\ \mathrm{L}\ \ltimes \qquad \rtimes\ \mathrm{L} \underset{✖\ \mathrm{HLL}}{\boxed{\mathrm{H}\ \mathrm{L}\ \mathrm{L}}} \ltimes$

Because, as in (3.10c), we know a string is ill-formed as soon as we see a single banned substring, there is no need for any memory—the scanning procedure only needs to pay attention to what is currently in the window. Thus, the SL languages are fundamentally *local* because their well-formedness depends solely on each individual substring. This is in contrast to the LT languages, which require some memory. For example, a scanning mechanism evaluating strings with respect to $L(\phi_{IH,LH})$ would, upon encountering a final L$\ltimes$ substring, have to remember whether or not a $\rtimes$H substring appeared earlier in the word. This makes evaluation of LT languages *global*, as opposed to local.

This local characteristic of the SL languages also makes them easily learned from positive data (in the sense of Gold, 1967), as long as the learner has *a priori* knowledge of k (García et al., 1990; Heinz, 2007). Briefly, such a learner only has to keep track of the substrings of length k seen in each example of the target language, and can infer that any substrings of length k it has *not* seen are banned substrings.

Given that evaluation and learning in the SL languages is so simple, it is significant that many phonotactic patterns in *natural* language are SL (Heinz, 2007, 2010a; Rogers et al., 2013). It has been shown here how Kumamoto Japanese is one such pattern. Many segmental coocurrence restrictions are SL as well (Heinz, 2007, 2010a). For example, Halle (1978) points out that English speakers reliably judge *plast* as a possible English word (even if they have never heard it before), but **ptak* as not. This can be attributed to the initial *[pt] sequence of *ptak*, which we can ban with the negative literal $\neg \rtimes$ pt.

In contrast, LT patterns such as $L(\phi_{IH,FH})$, in which only the word ends agree in a particular feature, are, to the best of my knowledge, unattested as well-formedness constraints. Thus, $\mathfrak{L}^{NL}$ and the SL languages provide us with a strong *hypothesis* for well-formedness constraints:

(3.11) **The $\mathfrak{L}^{NL}$ Hypothesis:** Surface well-formedness constraints in phonology are local over strings.

Local, again, is used here in the technical sense as defined in this chapter. This is thus a restrictive hypothesis because it constrains the range of grammars available to speakers to conjunctions of negative substring literals—i.e., statements in $\mathfrak{L}^{NL}$. As mentioned above, as a theory of well-formedness it also comes with straightforward cognitive interpretations of string evaluation and learning.

3.5 Expressivity and Structure

However, the $\mathfrak{L}^{NL}$ Hypothesis is far too strong. There are many attested *long-distance* patterns in natural language phonology that are not SL (Heinz, 2010a; Heinz et al., 2011; Rogers et al., 2013). This section gives a concrete example in the Hirosaki Japanese pattern discussed in the previous chapter. However, while $\mathfrak{L}^{NL}$ is too restrictive of a hypothesis for surface well-formedness in phonology, as shown in the last section we also do not want to increase the power of logic to $\mathfrak{L}^P$. This section introduces $\mathfrak{L}^{NL}_T$, which has the same syntax as $\mathfrak{L}^{NL}$ but is interepreted over *tiers*. This logic is more expressive than $\mathfrak{L}^{NL}$, but does not overgenerate in the same way that $\mathfrak{L}^P$ does. This section thus introduces an important concept for this dissertation: that, when faced with the need for a more expressive grammar, it is a more restrictive choice to enrich the representation than to increase the power of the logic.

3.5.1 A long-distance generalization

As motivation to move beyond $\mathfrak{L}^{NL}$ and $\mathfrak{L}^P$, consider the tone pattern of Hirosaki Japanese (Haraguchi, 1977). Recall from Chapter 2 §2.2.2.2 that words in Hirosaki Japanese can only have one H-toned or F-toned mora, and that F-toned morae can only appear word-finally (see p. 39). For now, let us simplify somewhat and focus on the generalization that in Hirosaki Japanese must have one and only one H-toned mora, which may appear anywhere in the word. The full version of the pattern, including the restriction involving falling tones, will be discussed in more detail in §3.6.1. A formal language over $\Sigma = \{\mathrm{H}, \mathrm{L}\}$ representing the well-formed strings of toned morae in (this simplified version of) Hirosaki Japanese is given below in (3.12). I'll denote this formal language as L'_{HJ}, as the true Hirosaki Japanese pattern L_{HJ} will be discussed in §3.6.1.

(3.12)

In L'_{HJ}	Not in L'_{HJ}
H, HL, LH,	L, LL, HH, LLL,
LHL, LLH, HLL,	HLH, HHL, HHH,
LLLH, LLHL,	LHH, LLLL, HLLL,
LHLL, HLLL,	…, HLLLH,
LLLLH, …	LLLLL, …

The formal language L'_{HJ} is the set of strings that have exactly one H; any string with no Hs or two or more Hs is not in L'_{HJ}.

This pattern cannot be described with either $\mathfrak{L}^{NL}$ *or* $\mathfrak{L}^P$. I start with $\mathfrak{L}^{NL}$. Every string must contain a H, but this H may be in any position. From a negative standpoint, this means that the pattern does not contain strings of all L morae; i.e., L, LL, LLL, etc. For any string of a particular length, we can of course write a literal that bans this string. For example, $\phi_{L\text{-}string}$ and $\phi_{LL\text{-}string}$ in (3.13) below pick out monomoraic and bimoraic L-tone strings, respectively.

(3.13) a. $\phi_{L\text{-}string} = \rtimes\text{L}\ltimes$ b. $\phi_{LL\text{-}string} = \rtimes\text{LL}\ltimes$ c. $\neg\phi_{L\text{-}string} \wedge \neg\phi_{LL\text{-}string}$

The conjunction of negative literals in (a)c can thus ban the strings L and LL. However, it cannot ban the string LLL; for this, the negative literal ¬LLL is necessary. In general, to ban *any* string of length n morae of all L tones, we would need a literal like $\phi_{L\text{-}string}$ or $\phi_{LL\text{-}string}$ for each n. Attempting to write a conjunction of thus these results in an infinite statement, as in (3.14) below.

(3.14) $\neg\phi_{L\text{-}string} \wedge \neg\phi_{LL\text{-}string} \wedge \neg\phi_{LLL\text{-}string} \wedge \neg\phi_{LLLL\text{-}string} \wedge \ldots$

Thus, L'_{HJ} cannot be captured with a statement in $\mathfrak{L}^{NL}$, because it is impossible to enforce that an H be present if it may appear anywhere in the word (note that this is in contrast to Kagoshima Japanese, where it was possible to force Hs if they were in a particular position).

To capture the generalization that each word must contain a H, we can use a statement in $\mathfrak{L}^{P}$, which can require substrings. The relevant literal is given in (3.15).

(3.15) a. $\phi_H = \text{H}$

However, $L(\phi_H) \neq L'_{HJ}$, because ϕ_H is satisfied by any string that has *at least* one H, not *exactly* one H:

(3.16)

In $L(\phi_H)$	Not in $L(\phi_H)$
H, HL, LH, HH	L, LL, LLL,
LLH, LHL, LHH,	LLLL, LLLLL,
HLL, HLH, HHL, ...	...

We can of course, for example, ban local sequences of H, or Hs separated by some finite number of Ls, by negating literals like the following:

(3.17) $\phi_{HH} = \mathrm{HH}$, $\phi_{HLH} = \mathrm{HLH}$

However, any string *HL^{n}H which violates the 'only one H' generalization, banning it requires a literal ϕ_{HL^nH}. Thus, just as statements in $\mathfrak{L}^{NL}$ cannot ban all strings containing no Hs, negative literals cannot ban all strings containing two or more Hs. Moving to propositional logic does not help, as positive constraints are of no use in banning a second H.

3.5.2 Increasing power versus enriching structure: $\mathfrak{L}^{NL}_T$

It is at this point that we must choose one of two options: move to a higher logic, or stay with a banned substring logic and enrich the representation over which the substrings are interpreted. There are more powerful logics than $\mathfrak{L}^P$, such as *first order* logic and *monadic second order logic*. However, as noted by Rogers et al. (2013), the range of languages these logics can describe properly includes the LT languages, and so they overgenerate both in the way that $\mathfrak{L}^P$ does and in additional ways.

Instead, fixing the power of the logic while enriching the representation allows for increased expressivity without succumbing to the overgeneration problem of increased logical power. The following presents one method for doing this. The remainder of this section presents a logical interpretation of the *Tier-based Strictly Local* (TSL) languages (Heinz et al., 2011), which can capture L'_{HJ} as given above.

Heinz et al. define the TSL languages as SL-like languages defined over a *tier*, or subset of the alphabet. The inspiration for the TSL languages is what Nevins (2010) calls *relativized locality* in natural language phonology: locality does not always operate over the surface string of segments, but among *related* segments. The idea is that banned substring constraints operate not over the surface string, but over a 'projection' of that string that operates only a subset of the alphabet.

For some alphabet Σ, let T denote some *tier* or subset of Σ. Let $erase_T$ be a function which takes a string $w \in \Sigma^*$ and returns a string in T^* in the following way:

Definition 6 (The $erase_T$ function) (Heinz et al., 2011) *For an alphabet Σ and a tier $T \subseteq \Sigma$, for any $w \in \Sigma^*$:*

$$erase_T(w) \overset{def}{=} \begin{cases} \lambda & \text{if } w = \lambda \\ erase_T(u) \bullet \sigma & \text{if } w = u \bullet \sigma; u \in \Sigma^*, \sigma \in T \\ erase_T(u) & \text{if } w = u \bullet \sigma; u \in \Sigma^*, \sigma \notin T, \sigma \in \Sigma \end{cases}$$

Intuitively, $erase_T$ is very simple—it goes through a string and erases any symbols not on the tier. To be explicit, it is defined here recursively to match Def. 1 of Σ^*. If $w = \lambda$, then $erase_T(w) = \lambda$. If $w = u \bullet \sigma$, then there are two options. If $\sigma \in T$, then $erase_T$ concatenates this to whatever it returns for u. If $\sigma \notin T$, then $erase_T$ leaves σ behind and simply outputs the result of $erase_T(u)$. In this way, symbols not on T are erased from the string. For example, if $\Sigma = \{a, b, c\}$ and $T = \{a, b\}$ then $erase_T(caccbca) = aba$.

The tier T and its function $erase_T$ can be seen as additional structure because they allow for a different order on the symbols than of a string in Σ^* itself. For example, in the string *caccbca* above, $erase_T(caccbca)$ reveals that *aba* are adjacent relative to T.

We can build a version of $\mathfrak{L}^{NL}$ which uses this concept of a tier. First, we define the full set of propositions of such a language parallel to $\mathfrak{L}^P$, although we won't be using all of them. For any T let $\rtimes T^* \ltimes$ be the set of strings in T delineated with $\rtimes$ and $\ltimes$ (just as $\rtimes \Sigma^* \ltimes$). Then let $\mathfrak{L}^P_T$ be defined exactly as $\mathfrak{L}^P$, with the one difference that string literals in $\mathfrak{L}^P_T$ are substrings of $\rtimes T^* \ltimes$, whereas they were of $\rtimes \Sigma^* \ltimes$ in $\mathfrak{L}^P$.

Definition 7 (Syntax of $\mathfrak{L}^P_T$) *Given a tier $T \subseteq \Sigma$, the following are sentences in $\mathfrak{L}^P_T$:*

a. *For any $\phi = u$ where u is a substring of some word in $w \in \rtimes T^* \ltimes$, $\phi \in \mathfrak{L}^P_T$*

b. *For any $\psi, \psi_1, \psi_2 \in \mathfrak{L}^P_T$, $(\neg\psi), (\psi_1 \wedge \psi_2), (\psi_1 \vee \psi_2), (\psi_1 \rightarrow \psi_2)$ are all in $\mathfrak{L}^P_T$, as in Def. 2.*

Nothing else is in $\mathfrak{L}^P_T$.

The interpretation of statements in $\mathfrak{L}^P_T$ then uses the $erase_T$ function. In $\mathfrak{L}^P$, for a string w its satisfaction of a string literal was evaluated directly on whether or not that literal was a substring of $\rtimes w \ltimes$. For $\mathfrak{L}^P_T$, it is evaluated instead over $\rtimes erase_T(w) \ltimes$.

Definition 8 (Semantics of $\mathfrak{L}_T^P$) *Given a sentence $\phi \in \mathfrak{L}_T^P$ and a word w,*

a. if $\phi = u$ and u is a substring of $\rtimes erase_T(w)\ltimes$, then $w \models \phi$

b. if ϕ is equal to $(\neg\psi), (\psi_1 \wedge \psi_2), (\psi_1 \vee \psi_2)$, or $(\psi_1 \rightarrow \psi_2)$ for any $\psi, \psi_1, \psi_2 \in \mathfrak{L}_T^P$, then satisfaction of ϕ by w is as defined in Def. 3.

Again, the semantics of $\mathfrak{L}_T^P$ are almost identical to those of $\mathfrak{L}^P$, with the exception that the literals of $\mathfrak{L}_T^P$ are evaluated relative to the tier. Note that $\mathfrak{L}_T^P$ is actually just a generalization of $\mathfrak{L}^P$, as when $T = \Sigma$, $\rtimes T^*\ltimes = \rtimes\Sigma^*\ltimes$ and the $erase_T$ function does not change the string.

To continue the example with $\Sigma = \{a, b, c\}$ and $T = \{a, b\}$, consider the statement $\neg ab$. By Def. 7, this statement is in $\mathfrak{L}_T^P$. Consider then the string *caccbca*. It does not satisfy $\neg ab$, because ab is a substring of $erase_T(caccbca) = aba$. In this way, we can take the idea of banned substrings and apply it to *non-local* structures, although in a restricted way.

From $\mathfrak{L}_T^P$ we automatically get a set of conjunctions of negative literals $\mathfrak{L}_T^{NL}$, as Definition 5 applies to *any* propositional logic. $\mathfrak{L}_T^{NL}$ thus represents a set of grammars specifying banned substrings over a tier. As we will see with L'_{HJ} momentarily, $\mathfrak{L}_T^{NL}$ can describe languages that $\mathfrak{L}^{NL}$ cannot (when $T \neq \Sigma$). However, it does not overgenerate in the way that $\mathfrak{L}^P$ does; because $\mathfrak{L}_T^{NL}$ cannot specify structures, it cannot describe $L_{\phi_{IH,FH}}$ for any T (Heinz et al., 2011). Thus, $\mathfrak{L}_T^{NL}$ occupies something of a middle ground between $\mathfrak{L}^P$ and $\mathfrak{L}^{NL}$ which captures something important about natural language phonology.[4] This is because the class of languages $\mathfrak{L}_T^{NL}$ describes, the TSL languages, have been argued to be sufficient for long-distance phonotactics in segmental phonology, including vowel harmony, consonant harmony, and long-distance dissimilation (Heinz et al., 2011; McMullin and Hansson, to appear). The L'_{HJ} pattern

[4] Formally, it is not exactly 'in the middle'—the class of languages describable by $\mathfrak{L}_T^{NL}$ is more expressive than that of $\mathfrak{L}^{NL}$, but incomparable to that describable by $\mathfrak{L}^P$(that is, it can describe some languages that $\mathfrak{L}^P$ can, but not all, and vice versa)(Heinz et al., 2011).

offers a concrete example of how $\mathfrak{L}^{NL}_T$ can model long-distance well-formedness conditions. Recall that every string in L'_{HJ} must have exactly one H. If we set $T = \{H\}$, then we can capture this generalization by constraining the strings that appear on the tier of H tones.

The two literals in (3.18) pick out an empty string and a sequence of Hs, respectively. The conjunction of negative literals in (3.19) bans both of these substrings. Note that $\phi_{\rtimes\ltimes}$, although we have not used it before, is a valid statement in $\mathfrak{L}^P_T$, as $\rtimes\lambda\ltimes = \rtimes\ltimes \in \rtimes T^*\ltimes$ and $\rtimes\ltimes$ is a substring of itself.

(3.18) a. $\phi_{\rtimes\ltimes} = \rtimes\ltimes$ b. $\phi_{HH} = \text{HH}$

(3.19) $\phi_{\textit{One-H}} = \neg\phi_{\rtimes\ltimes} \wedge \neg\phi_{HH}$

If $T = \{H\}$, then $erase_T$ erases any Ls from a string. For any string w of *only* Ls, $\rtimes erase_T(w)\ltimes$ returns $\rtimes\ltimes$, and so w will contain $\rtimes\ltimes$ and thus not satisfy $\neg\phi_{\rtimes\ltimes}$. Conversely, if w contains more than one H, erasing all of the Ls will render two Hs adjacent, and w will thus not satisfy $\neg\phi_{HH}$. The table below illustrates this in detail for several different strings in Σ^*.

(3.20)

w	$\rtimes erase_T(w)\ltimes$	$w \models \neg\phi_{\rtimes\ltimes}$?	$w \models \neg\phi_{HH}$?	$w \in L(\phi_{\textit{One-H}})$?
LLLLLL	$\rtimes\ltimes$	**no**	yes	**no**
HLLLLH	$\rtimes$HH$\ltimes$	yes	**no**	**no**
LHLHHL	$\rtimes$HHH$\ltimes$	yes	**no**	**no**
LLLHLL	$\rtimes$H$\ltimes$	yes	yes	yes
HLLLLL	$\rtimes$H$\ltimes$	yes	yes	yes

In essense, $\phi_{\textit{One-H}}$ is only satisfied by strings whose value for $erase_T$ is equal to $\rtimes$H$\ltimes$. This is exactly the set of strings which contain only one H. Thus, when $\phi_{\textit{One-H}}$ is interpreted as a statement in $\mathfrak{L}^{NL}_T$, $L(\phi_{\textit{One-H}}) = L'_{HJ}$.

3.5.3 Interim Summary

So far, this chapter has reviewed three kinds of logical grammars for surface well-formedness constraints: propositional $\mathfrak{L}^P$ and conjunctions of negative literals $\mathfrak{L}^{NL}$ over plain strings, and conjunctions of negative literals $\mathfrak{L}_T^{NL}$over tiers. As $\mathfrak{L}^{NL}$ represented the idea of *local* well-formedness by listing a set of banned substrings, it presented a strong hypothesis for phonology, given in (3.11) as the $\mathfrak{L}^{NL}$ Hypothesis. The preceding section showed how the $\mathfrak{L}^{NL}$ Hypothesis was too strong, as it could not capture long-distance well-formedness constraints, such as L'_{HJ}.

However, $\mathfrak{L}_T^{NL}$ *could* describe L'_{HJ}, and has been argued elsewhere to cover long-distance phonotactics in segmental phonology. As $\mathfrak{L}_T^{NL}$ maintains the restrictive nature of banned substring constraints, it is an attractive hypothesis for phonology. We can thus revise the $\mathfrak{L}^{NL}$ Hypothesis to the $\mathfrak{L}_T^{NL}$ Hypothesis:

(3.21) **The $\mathfrak{L}_T^{NL}$ Hypothesis:** Surface well-formedness constraints in phonology are local over tiers.

However, while (3.21) may be a satisfactory hypothesis for segmental phonology, the following section shows that it is not satisfactory for tone.

3.6 Long-distance Tone Patterns and Local String Logics

The previous chapter noted three tonal patterns as representative of the 'long-distance' character of tone patterns: the surface pattern resulting from Unbounded Tone Plateauing and the accent patterns of Hirosaki Japanese and Wan Japanese. This section gives formal credence to this claim, as it shows that, when viewed as patterns over strings, they are non-local over both $\mathfrak{L}^{NL}$ and $\mathfrak{L}_T^{NL}$. This thus shows that the $\mathfrak{L}_T^{NL}$ Hypothesis is too strong for tone patterns, which helps to motivate negative literal constraints over *autosegmental representations*, the focus of the remainder of the dissertation.

3.6.1 Hirosaki Japanese (full)

The discussion in the previous section of L'_{HJ} concerned only a distinction between H and L, but in reality Hirosaki Japanese has three tones on the surface, H, L, and a falling tone F (Haraguchi, 1977). The distribution of these tones is as follows.

At most one H or F can appear in a word, and F may only appear word-finally. Each word must contain either a H or F, and cannot contain both. This full pattern is given as L_{HJ} below.

(3.22)

In L_{HJ}	Not in L_{HJ}
H, F, LH, LF, HL,	L, LL, HH, HF,
LLH, LLF, LHL, HLL,	LLL, FLL, LFL, HLF,
LLLH, LLLF, LLHL, LHLL,	LLLL, LLFL, LFLL,
HLLL, ...	FLLL, HLLF, ...

As discussed in the previous chapter, Haraguchi (1977) gives these patterns an accentual analysis, in which a nonfinal H or final F indicates an accented mora, and final H is the default tone pattern for accentless words. I do not take issue with this analysis, but I will instead focus on the formal properties of the surface pattern.

Intuitively, $\mathfrak{L}^{NL}_T$ cannot capture L_{HJ} because T needs to be $\{\mathrm{H}, \mathrm{F}\}$, but F has to be restricted to word-final position. The reasons for this are as follows. First, like L'_{HJ}, we may only have one H, but in L_{HJ} there is the further condition that there may be only one H *or* F, neither both (e.g., *HLF, *HLLF). In order to capture this, we set $T = \{H, F\}$ and augment $\phi_{\textit{One-H}}$ from (3.19) above in the following way:

(3.23) $\phi_{\textit{One-H-or-F}} = \neg\phi_{\rtimes\ltimes} \wedge \neg\phi_{HH} \wedge \neg\phi_{HF} \wedge \neg\phi_{FH} \wedge \neg\phi_{FF}$

Where ϕ_{HF}, ϕ_{FH}, and ϕ_{FF} represent HF, FH, and FF substrings on the tier, respectively. Parallel to $\phi_{\textit{One-H}}$, $\phi_{\textit{One-H-or-F}}$ is only satisfied for words for which $erase_T$ returns either $\rtimes$H$\ltimes$ or $\rtimes$F$\ltimes$.

However, $\phi_{\textit{One-H-or-L}}$ is also satisfied by strings like *LFLL, *LFLL, *FLLL, which are not in L_{HJ} because F is not in final position. Restricting F to word-final position requires a constraint ¬FL as well. However, if $T = \{H, F\}$, as was shown above to be necessary to capture the 'only one H or F' generalization, then by Def. 7, ¬FL is not a valid statement in $\mathfrak{L}^{P}_T$ or $\mathfrak{L}^{NL}_T$, because L $\notin T$.

It is possible describe the pattern with the *intersection* of the TSL language with an SL one: $L(\phi_{\textit{One-H-or-F}})$, where $\phi_{\textit{One-H-or-F}}$ is interpreted as a statement in $\mathfrak{L}^{NL}_T$, and $L(\neg\text{FL})$, where $\neg$FL is interpreted as a negative literal in $\mathfrak{L}^{NL}$. This misses a generalization though. Most phonologists would agree that the reason H and F both can't appear in the string is because F is, in reality, an HL sequence on the tonal tier, and so a word with both an H and HL will contain two H tones.[5] If we think about the generalizations in these terms, the constraint on F to word-final position also takes a different character. H is thus allowed to appear anywhere in the word, but there are specific restrictions on when it is allowed to combine with L. As we shall see in later chapters, this behavior is described very naturally with local constraints over autosegmental representations.

3.6.2 Wan Japanese Type β

As discussed in the preceding chapter, words in Wan Japanese have one of two tonal types, referred to in the literature as Type α and Type β (Breteler, 2013; Kubozono, 2011b). The pattern of Type β words depends on morphological information, and it is on this from which it draws its complexity.

To review, words without suffixes are pronounced with an H^{n}LHL pattern, where the initial string of H^{n}L is shortened or removed in words too short to fully realize it (see (2.42), p. 42).

(3.24) HL, LHL, HLHL, HHLHL, HHHLHL, ...

Words with suffixes show a H^{n}LHH pattern before the suffix, and then H^{m}L on the suffix. The first n is dependent on the length of the stem morpheme, whereas

[5] Haraguchi (1977) shows that both non-final Hs and final Fs are the surface realization of an accented mora. Final Hs are thus the default tone pattern for unaccented words. One might wonder how incorporating the accent into the alphabet—specifically, replacing non-final H and F with Ĥ—may affect the formal properties of the pattern. Interestingly, it does not: we would still need to state that each word have exactly one of either Ĥ or H (the unaccented final H), and that H is relegated to word-final position. Thus, the same kind of constraints are required whether or not the accent is incorporated into the representation.

m is dependent on the length of the suffix morpheme. Here, '-' denotes the relevant morpheme boundary (see (2.44), p. 43).

(3.25) HH-L, HH-HL, HH-HHL, ..., LHH-L, LHH-HL, LHH-HHL, ..., HLHH-L, HLHH-HL, ..., HHLHH-L, HHLHH-HL, ...

Let L_{WJ} refer to the union of the sets of strings represented in (3.24) and (3.25). A statment in $\mathfrak{L}^{NL}$ can capture (3.24), but no statement in $\mathfrak{L}^{NL}$ nor $\mathfrak{L}^{NL}_{T}$ can capture the entirety of L_{WJ}. Consider the statement in (3.26) below.

(3.26) $\neg \rtimes\mathrm{LL} \wedge \neg\mathrm{H}\ltimes \wedge \neg\mathrm{HHL}\ltimes$

This statement accurately describes the facts of (3.24)—a string can't start with a span of two Ls, a word cannot end on an H, and a word cannot end on a HHL sequence.

However, the last negative literal $\neg$HHL$\ltimes$ is not true of strings in (3.25), e.g., LHH-HHL, which end on a HHL sequence. We can instead use $\neg$LHHL$\ltimes \wedge \neg$HHHL$\ltimes$, which bans any final HHL sequence not immediately preceded by the morpheme boundary, but this would incorrectly ban strings such as LHH-HHHL, which is in L_{WJ} if we assume that suffixes have a H^{n}L pattern.

The correct statement is that *if* a morpheme boundary '-' is present, *then* a final HHL may appear (or, if the '-' appears closer to the end of the word, then -HL$\ltimes$ or -L$\ltimes$). If it is *not* present, then final HHL is *not* licensed. The following statement in $\mathfrak{L}^{P}$captures this:

(3.27) $\big(\text{-} \rightarrow (\mathrm{HHL}\ltimes \vee \text{-HL}\ltimes \vee \text{-L}\ltimes)\big) \wedge \big(\neg\,\text{-} \rightarrow \neg\mathrm{HHL}\ltimes\big)$

However, as already discussed in (3.4.4), such statements are beyond the power of logics like $\mathfrak{L}^{NL}$ and $\mathfrak{L}^{NL}_{T}$, which can only ban substrings. In intuitive terms, this is because the '-' may appear in any position in the string, but its presence bears on the realization of the last three TBUs. This is thus not a *local* dependency. Thus, the dependence of Wan Japanese on morphological information keeps it outside the expressivity of both $\mathfrak{L}^{NL}$ and $\mathfrak{L}^{NL}_{T}$.

3.6.3 Unbounded tone plateauing

Recall from Chapter 2 that in unbounded tone plateauing (UTP; Kisseberth and Odden, 2003; Hyman, 2011b; Jardine, to appear), underlying H tones merge, creating a plateau of H tones between them. This results in a surface pattern in which strings can only include one plateau of high tones. The following data are repeated from (2.33) and (2.34) from the previous chapter (p. 37).

(3.28) a. /mu-tém-a+bi-sikí/ mutémá+bísíkí 'log chopper'
LHHHHH

b. /bikópo byaa-walúsiimbi/ bikópó byáá-wálúsiimbi 'the cups of Walusimbi'
LHHHHHHHLLL

c. /tw-áa-mú-láb-a walúsimbi/ tw-áá-mu-lab-a walúsimbi
'we saw him, Walusimbi' HHLLL LHLL

d. /tw-áa-láb-w-a walúsimbi/ tw-áá-láb-wá wálúsimbi
'we were seen by Walusimbi' HHHHHHLL

e. /tw-áa-láb-a byaa=walúsimbi/ tw-áá-láb-á byáá-wálúsimbi
'we saw those of Walusimbi' HHHHHHHHLL

The set of strings which belong to this pattern, which I'll denote L_{UTP}, are given below in (3.29).

(3.29) L_{UTP} = LLL, LLLL, LHL, LLHL, LHLLLL, LLLLHL, HHH, LHHHL, LHHHHH, LLLHHL, LLHHHL, ...

As UTP creates a plateau between *any* underlying H tones, surface strings in a UTP generalization cannot contain Hs separated by L tones.

(3.30) Strings not in L_{UTP}

HLH, HLLH, HLHH, HLLLH, HHHHHLH, HLLLLLLH, ...

The correct generalization for the surface pattern of UTP is thus that all strings must have at most one unbroken plateau of H tone morae. As such, we must adopt some strategy for banning any number of L tones in between H tones.

Exactly as explained in §3.5.1 for all L-tone words in Hirosaki Japanese, this cannot be done with negative literals in $\mathfrak{L}^{NL}$. We can start with ¬HLH, bans a HLH substring. This substring appears in some of the strings in (3.30), as shown in (3.31) below. However, while ¬HLH can ban strings with Hs separated by exactly one L (as shown in (3.31a)), but not Hs separated by more than one L (as shown in (3.31b)).

(3.31) a. ***HLH**, ***HLH**H, *HHHH**HLH**, ...
b. *HLLH, *HLLLH, *HLLLLLLH, ...

To ban Hs separated by exactly two Ls, we could use the following negative literal, but again, it would be unable to pick out Hs separated by three or more Ls:

(3.32) a. $\neg\phi_{HLLH} = \neg$HLLH
b. ***HLLH**
c. *HLLLH, *HLLLLLLH, ...

Thus, UTP cannot be described by statements in $\mathfrak{L}^{NL}$. In fact, because of this issue, it cannot be described by $\mathfrak{L}^{P}$, either—adding full propositional constraints does not get around the fact that we need to ban Hs separated by any length of Ls. Finally, $\mathfrak{L}^{NL}_{T}$ is of no use either, as the interaction between Hs and Ls is important, and isolating either in a tier cannot obtain the desired effect.

3.7 Conclusion

The previous chapter established that previous theories of tonal well-formedness either do not make clear typological predictions or over- or undergenerate in serious ways. This chapter began a search for a sufficient with clear, sufficient, and restrictive theory of well-formedness through formal languages and the logics that describe them. In terms of restrictiveness, a notion of *locality* was introduced that was based on the idea of banning substructures, which in this chapter were sub*strings*. In particular, the following hypothesis based on $\mathfrak{L}^{NL}_{T}$ was raised, originally in (3.21).

(3.33) **The $\mathfrak{L}^{NL}_{T}$ Hypothesis:** Surface well-formedness constraints in phonology are local over tiers.

This was based on $\mathfrak{L}_T^{NL}$'s relatively good fit to patterns in segmental phonology. However, this hypothesis was shown in the preceding section to be too strong, at least for tonal phonology.

As mentioned earlier in the chapter, there are two options when dealing with the inadequacy of a logical language with regards to the empirical facts. One is to increase the power of the logic. As we have already seen, this leads to overgeneration, even for tone. The remainder of this dissertation is concerned with *enriching the structure*, in a way that is inspired by the success of APRs in tone. We shall move from strings to *graphs*, but shall keep the idea of banned substructure constraints.

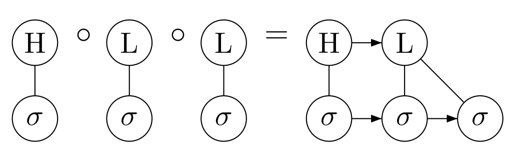
Chapter 4

AUTOSEGMENTAL PHONOLOGICAL GRAPHS

The preceding two chapters have motivated a need for a *local* theory of language-specific well-formedness constraints for autosegmental structures. The previous chapter showed how such a theory can be articulated using logical statements whose most basic statements are substructures, although in that chapter the relevant substructures were substrings of strings ($\mathfrak{L}^{NL}$) and tiers ($\mathfrak{L}^{NL}_{T}$). In order to develop such a logic for APRs, we must first explicitly define APRs, just as strings were formally defined in the preceding chapter. We can then define substructures of APRs, and thus logics using these substructures as literals, which will be the subject of the following chapter. The purpose of this chapter, then, is to explicitly define APRs using *graphs* to represent the information in APRs. This is possible because APRs were originally defined as graphs (Goldsmith, 1976; Coleman and Local, 1991).

Counter to previous formalizations of APRs, the current chapter defines APRs through *the concatenation of graph primitives*, which highlights heretofore unrecognized *string-like* properties of APRs. As shall be seen, this view of APRs has many advantages, but for the purposes of the current dissertation the most important is that it lets us relate APRs directly to surface strings. To give an example, the following shows how the APR corresponding to the surface string HLL can be generated from the concatenation ($\circ$) of primitives representing H and L TBUs. The exact notational conventions will be described in detail momentarily.

(4.1) a. HLL = 

b. g(HLL) = 

The notion of relating strings to graphs comes from the work of Courcelle et al. (2012) and Engelfriet and Vereijken (1997), who show how concepts of formal languages can be extended to graphs through generalizing string operations to graphs. By fixing a relationship between strings and APRs, this concatenation operation provides a way to directly compare the expressive power of local grammars over APRs and the grammars over strings introduced in the previous chapter; this will be of importance in Chapter 6, which examines the ability of APRs to capture the long-distance patterns shown in Chapter 3 to be beyond the power of these string grammars. Additionally, relating strings to APRs opens a path for developing a learning model for these local APR grammars directly from strings; this will be discussed further in Chapter 8. Chapter 7 discusses how a modified version of the concatenation operation introduced here can be used to build structures representing correspondence relations (McCarthy and Prince, 1995; Potts and Pullum, 2002).

Finally, there are also empirical consequences of looking at APRs through the concatenation of a finite set of primitives: it correctly predicts that languages can have unbounded spreading, but not unboundedly long contours. This fact is not made explicit in previous formalizations of APRs.

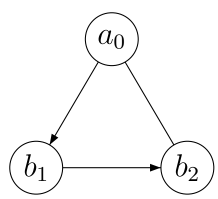
This chapter is structured as follows. §4.5 discusses the core theoretical and empirical consequences of concatenation for tonal phonology. However, in order to build up to this chapter, the preceding sections do the formal legwork. §4.1 defines the *labeled mixed graphs* that will be used throughout the dissertation to represent APRs, as well as some further mathematical preliminaries that are necessary for dealing with graphs. §4.2 formalizes the WFCs discussed in Chapter 2 as a set of axioms—it is this traditional axiomatic approach to which the concatenation approach is compared. §4.3 then defines a concatenation operation for labeled mixed graphs, and §4.4 shows how this concatenation operation, combined with a set of primitives with particular properties, can generate a set of graphs that follow the axioms set out in §4.2. §4.5 shows how this concatenation operation can be applied to model phenomena in tone, and §4.6 concludes. Much of the material in this chapter was first presented in Jardine and Heinz

(2015). However, it has been significantly rewritten for a more general audience.

4.1 Mathematical Preliminaries II: Graphs

Chapter 3 was concerned with *strings*, but in order to model the information contained in APRs, we must move to *graphs*, specifically *labeled mixed graphs*. 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 alphabet Σ. Formally, 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. As labeled mixed graphs are the main type of graph used in this dissertation, they will henceforth be referred to simply as *graphs*.

(4.2) a. G



b. $G = \Big\langle\ V = \{0, 1, 2\},$
$E = \{\{0, 2\}\},$
$A = \{(0, 1), (1, 2)\},$
$\ell(0) = a, \ell(1) = \ell(2) = b\ \Big\rangle$

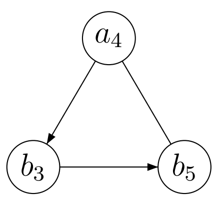
Consider the graph G in (4.2). It has three nodes, $0, 1$ and 2. When indicated, node indices here and throughout the dissertation will be indicated as subscripts on the labels. Node indices will come from the set $\mathbb{N}$ of natural numbers $\{0, 1, 2, ...\}$. Node 0 has the label a, and nodes 1 and 2 both have the label b. There is an undirected edge $\{0, 2\}$ between nodes 0 and 2, and there are two directed edges, one from 0 to 1 and one from 1 to 2.

Two notions will be helpful in defining the sets of graphs we want. One is that of a *partition*. A partition P of a set X 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 $X_i, X_j \in P$, $X_i \cap X_j = \emptyset$. Intuitively, a partition exhaustively breaks the elements of a set into non-overlapping subsets. A *directed path* through a graph G is a series of directed edges $(v, v_0), (v_0, v_1), (v_1, v_2), ..., (v_n, w) \in A$ between nodes v and w in V. Let $v \prec w$ denote when there is a path from v to w and let $v \preccurlyeq w$ be true when either $v \prec w$ or $v = w$.

We will be only considering graphs in which the directed edge relation is *asymmetric*; that is, $(x, y) \in A$ implies $(y, x) \notin A$. These will also be *simple* graphs, or those without *multiple edges* connecting the same pair of nodes. Thus, $\{x, y\} \in E$ implies $(x, y) \notin A$, and $(x, y) \in A$ implies $\{x, y\} \notin E$. Note also that the way edges are defined, we cannot have *loops*, or an edge or arc whose beginning and end nodes are the same. Let $GR(\Sigma)$ denote the set of all graphs following these properties whose labels are in Σ. For example, the graph in (4.2) is in $GR(\Sigma)$ for $\Sigma = \{a, b\}$.

A technical issue is that, following Courcelle et al. (2012)'s notion of *abstract graphs*, in this dissertation *isomorphic* graphs will be considered to be equal. Intuitively, isomorphic graphs have the same structure but their sets of nodes are distinct (although of the same cardinality). Formally, two graphs $G_1, G_2 \in GR(\Sigma)$ are isomorphic iff there is a mapping m from nodes in V_1 (the nodes of G_1; parts of a graph will be referenced using its subscript or prime symbol) to nodes in V_2 such that for all pairs of nodes $v, w \in V_1$, $\{v, w\} \in E_1$ iff $\{m(v), m(w)\} \in E_2$, $(v, w) \in A_1$ iff $(m(v), m(w)) \in A_2$, and for all nodes $v \in V_1$, $\ell_1(v) = \ell_2(m(v))$. For example, G' in (4.3) is isomorphic to G from (4.2). The mapping m from V to V' that shows this is given in (4.3b).

(4.3) a. G'



b. $m(0) = 4$

$m(1) = 3$

$m(2) = 5$

Thus, because G' is isomorphic to G, for the purposes of this dissertation, $G' = G$.

The following shows how APRs are, essentially, graphs in $GR(\Sigma)$ where E is the set of associations, A is the order on each tier, and Σ is the set of possible autosegmental units (H, L μ, σ, etc.). However $GR(\Sigma)$ is undesirable as a *theory* of autosegmental representation, as it includes graphs like G in (4.2) which are uninterpretable as APRs. Thus, the set of APRs must be some subset of $GR(\Sigma)$. In order to find out what that set is, we must formally establish the properties that make an APR an APR. The following thus formalizes the properties of APRs, discussed informally in Chapter 2.

The remainder of this chapter shows how to derive this set through the concatenation of elements from an alphabet of graph primitives.

4.2 The Top-down Approach: Properties of APRs as Axioms

In Chapter 2, §2.1 discussed the defining properties of APRs. This section defines these properties formally in terms of graphs. Formal treatments of these properties have existed since Goldsmith (1976)'s original definition of APRs, however, like Goldsmith, they 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. Second, spreading is achieved through two concatenation operations, one which draws a single association line from the last element of the melody tier in the first bistring to the first element of the timing tier of the second, and one which does the same from the timing tier to the melody tier. As a result, key properties like the NCC must be specified as axioms.

In sum, the defining properties of APRs have been traditionally defined in an axiomatic, *top-down* approach. This section will thus first introduce these properties as axioms. The remainder of the chapter will show how these properties can be *derived* from the concatenation operation defined later on, and thus not necessarily as specified as axioms.

4.2.1 Basic structure

The first formalization of APRs was in Goldsmith (1976), but the discussion here is based around Coleman and Local (1991) detailed definition of APRs, which makes explicit parts of the definition that Goldsmith leaves implicit. However, the notation used here will be based on conventions in graph theory, and thus is different from Coleman and Local (1991).

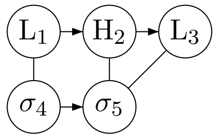
An APR is a set of elements with two binary relations over those elements. One relation is the linear order on each tier, and one relation is the association lines between these elements. As an example, take the following APR from Chapter 2, originally given in (2.5) on p. 20. In (4.4) below each element has been indexed with a subscript, which helps us distinguish, for example, the first syllable from the second syllable.

(4.4) $\mathrm{L}_1\mathrm{H}_2\mathrm{L}_3$

$\sigma_4\sigma_5$

In (4.4) there are five elements, $\{1, 2, 3, 4, 5\}$. This particular numbering is not essential, but it is important that we have some way of keeping each element distinct (remember that a set is a collection of *distinct* elements, so it is not helpful to refer to the set $\{\mathrm{L}, \mathrm{H}, \mathrm{L}, \sigma, \sigma\}$, because it is equivalent to $\{\mathrm{H}, \mathrm{L}, \sigma\}$). Of course it is important to keep track of what each element is; e.g. a H tone, a syllable, etc. Let $\Sigma = \{\mathrm{H}, \mathrm{L}, \sigma\}$. We can achieve this with a labeling function ℓ which maps elements in V to elements in Σ. The labeling function can be thought of as giving the 'value' of each element in the APR, where 'value' means whether it is a H tone, a syllable, etc. In (4.4), $\ell(2) = \mathrm{H}$, $\ell(1) = \ell(3) = \mathrm{L}$, and $\ell(4) = \ell(5) = \sigma$.

By now, it is perhaps clear that this labeling function and set of elements can be represented by a V and ℓ of a graph. What remains to be discussed is the linear order on each tier and the associations between tiers. These will be discussed separately in the following sections, but briefly, the order on each tier can be represented in a graph by a series A of ordered pairs, and the associations between elements can be represented by a set E of undirected edges in a graph. Thus, (4.4) can be represented as the following graph in $GR(\Sigma)$ where $\Sigma = \{\mathrm{L}, \mathrm{H}, \sigma\}$.

(4.5) a. 

b. $\langle \; V = \{1,2,3,4,5\},$
$E = \{\{1,4\},\{2,5\},\{3,5\}\},$
$A = \{(1,2),(2,3),(4,5)\},$
$\ell(1) = \ell(3) = \text{L}, \ell(2) = \text{H}, \ell(4) = \ell(5) = \sigma \; \rangle$

It is important to remember that *APGs represent the exact same information as APRs and thus are essentially equivalent objects*. Thus, although they are depicted differently here, APGs represent the same set of relationships between sound units that APRs do. Thus, the terms 'APGs' and 'APRs' can be used interchangeably, although from this chapter forth, the focus will be on APGs.

The following discusses the tiers and associations in more detail, and distinguishes APGs from other graphs in $GR(\Sigma)$ by means of axioms governing associations and tiers. For the sake of simplicity, the following discussion centers on APGs modeling APRs with only two tiers. There are two reasons for this. One, for the tonal patterns discussed in this dissertation, only two tiers are necessary. Two, two tiers is the simplest case with which we can begin to study the formal properties of APRs. As will be pointed out at the relevant points in the chapter, the discussion here can easily be extended to multi-tier APGs/APRs.

4.2.2 Tiers

Goldsmith (1976) defines a tier (which he called "levels") as a totally ordered sequence of elements $a^i_1 a^i_2 ... a^i_n$, where i is the index for tier L^i. Let us call the tonal tier in (4.4) L^1 and the timing tier L^2. Thus, L^1 is a string of three tone autosegments, $\text{H}_1\text{L}_2\text{H}_3$, and L^2 is a string of syllables $\sigma_4\sigma_5$. As in the graph in (4.5), we can represent the order on these strings with a set of ordered pairs. For example, $\{(1,2),(2,3)\}$ represents the order on L^1, as element 1 comes before 2 and 2 comes before 3. Similarly, the order on L^2 can be modeled with the set $\{(4,5)\}$.

It should be noted that some authors, e.g. Bird and Klein (1990), define an ordering relation *between* tiers. In an APR like (4.4), this means there would be an order among tones and syllables. This complicates the above definition of a tier, and it is unclear why such an ordering is necessary. As such, the definitions here will not include such an ordering (as is true for Goldsmith (1976)'s original definition of APRs).

To stipulate that APGs are bifurcated into two totally ordered tiers we can employ the following axiom.

Axiom 1 (Totally ordered tiers) *There is a partition* $\{V_0, V_1\}$ *on the set of nodes* V *such that for any two distinct nodes* $v_0, w_0 \in V_0$*, either* $v_0 \prec w_0$ *or* $v_0 \prec w_0$ *but both are not true. Likewise for* V_1*.*

This means that, for each of V_0 and V_1, there is a single directed path which connects all of their respective nodes and is not cyclic (because there can't both be a path from v_0 to w_0 and from w_0 to v_0). It is simple to show that this implies a total order on each set V_0 and V_1. The reader can confirm that the sets $\{1, 2, 3\}$ and $\{4, 5\}$ form such a partition on the set of nodes for the graph in (4.5); thus, it satisfies Axiom 1.

Recall from §2.1.2.1 that this dissertation will assume that tiers can only carry like segments. This can be specified axiomatically by saying that the partition of V from Axiom 1 resepects some partition of the alphabet Σ. Let this latter partition be known as the *tier partition*.

Axiom 2 (Tier partition on the alphabet) *There is a tier partition* $T = \{T_m, T_t\}$ *on* Σ *such that given the partition* $\{V_0, V_1\}$ *induced by the directed paths on each tier, for all* $v_0 \in V_0$*,* $\ell(v_0) \in T_m$ *and for all* $v_1 \in V_1$*,* $\ell(v_1) \in T_t$*.*

For example, for the $\Sigma = \{\mathrm{H}, \mathrm{L}, \sigma\}$ from (4.5) above, consider the tier partition $T = \{T_m, T_t\}$ where $T_m = \{\mathrm{H}, \mathrm{L}\}$ and $T_t = \{\sigma\}$. (The meaning of the subscripts t and m will become clear in a moment.) The graph in (4.5) respects this partition, as for the ordered sets $\{1, 2, 3\}$ and $\{4, 5\}$ just discussed, $\ell(1), \ell(2), \ell(3) \in T_m$ and for $\ell(4), \ell(5) \in T_t$. Essentially, the tier partition simply breaks the alphabet down into the groups of symbols that are allowed to appear on each tier in the graph. From now on, let us refer to the node partitions corresponding to T_t and T_m as V_t and V_m, respectively.

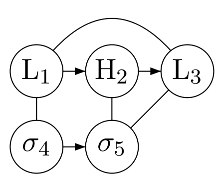
The first approximation of the definition of APGs is the subset $APG(\Sigma, T) \subset GR(\Sigma)$ of graphs which follow Axioms 1 and 2 and respect the tier partition T on Σ. However, there are a few more important axioms.

4.2.3 Associations

We next consider the set of associations in an APR. Goldsmith (1976)'s defines the set of association lines between two levels as a totally ordered set of pairs of elements on each level. As we have seen, these pairs can be represented in a graph by the undirected edges of E. We can use undirected edges as, to the best of my knowledge, no AP analysis has ever hinged on the direction of an association. Note, however, that there is no order on these edges. However, this is not an issue. As Coleman and Local (1991) point out, this total order on the associations is unnecessary if the NCC is present. This section will define the NCC as an axiom momentarily, and so the total order is unnecessary. The following section, §4.3, shows how to define the set of APGs through concatenation such that neither is necessary.

First, it is important to ensure that associations are only between tiers. Consider the following graph in $GR(\Sigma)$.

(4.6) a.



b. $\left\langle \begin{array}{l} V = \{1,2,3,4,5\}, \\ E = \{\{1,4\},\{2,5\},\{3,5\},\{1,5\}\}, \\ A = \{(1,2),(2,3),(4,5)\}, \\ \ell(1) = \ell(3) = \mathrm{L}, \ell(2) = \mathrm{H}, \ell(4) = \ell(5) = \sigma \end{array} \right\rangle$

The graph in (4.6) follows Axioms 1 and 2 but we clearly do not want it in $APG(\Sigma, T)$, as it has an association between the two L tones. The following axiom thus bans intra-tier associations:

Axiom 3 *For any edges $v, w \in V$, if there is a directed path between v and w, then $\{v, w\} \notin E$.*

Graphs obeying Axioms 1 through 3 thus have the basic structure of APRs, comprising tiers of like autosegments and associations between these tiers. We can now move on to applying the more substantive restrictions on APRs to APGs. For example, §2.1.2.3 introduced Goldsmith (1976)'s Well-Formedness Condition on association lines (p. 25):

(4.7) The Well-formedness Condition (Goldsmith, 1976, (24))

a. All TBUs are associated with at least one tone;
All tones are associated with at least one TBU.

b. Association lines do not cross.

It is easy to write an axiom for (4.7a)

Axiom 4 (Full specification) *For any $v \in V$, there is some $\{v, w\} \in E$ for some $w \in V$ $(v \neq w)$.*

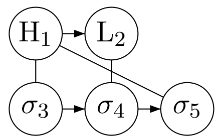
Axiom 4 simply requires that each element in the graph participates in an edge (i.e., an association).

It was discussed in that section that (4.7a), which forces full specification, was later shown to be too strong (e.g. by Pulleyblank (1986)). However, for the tone patterns discussed in Chapter 2 it was enough to consider it universal, and so for the moment let us only consider graphs that conform to Axiom 4. Language-specific violations of (4.7a)/Axiom 4, namely underspecification and floating tones, will be discussed in §4.5.3.

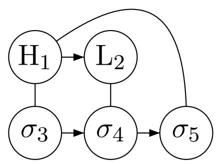
The other major constraint on the associations is (4.7b), the NCC. As various definitions of the NCC have been proposed, it will be discussed in its own section.

4.2.4 NCC

Consider the graph in (4.8), a graph version of the NCC-violating (2.16c) in Chapter 2 (p. 25).

(4.8) a. 

b. $\langle \quad V = \{1,2,3,4,5\}\,,$
$E = \{\{1,3\},\{1,5\},\{2,4\}\},$
$A = \{(1,2),(2,3),(4,5)\},$
$\ell(1) = \ell(3) = \mathrm{L}, \ell(2) = \mathrm{H}, \ell(4) = \ell(5) = \sigma \quad \rangle$

c. 

Both (4.8a) and (4.8c) are drawings of the *same* graph defined in (4.8b), but the association lines only 'cross' in (4.8a). This highlights the fact that there is nothing intrinsically 'line-crossing' about (4.8b), at least when our understanding of line-crossing is limited to drawings on paper. This is exactly Coleman and Local (1991)'s point that we cannot naïvely interpret the NCC as a constraint on drawings of APRs, as APRs are formal objects distinct from how we choose to depict them.

Goldsmith (1976) does provide a formal definition of the NCC based on the idea that association lines preserve contiguity on each tier. However, Sagey (1986) points out a serious flaw in this formalization, showing that, given an equal number of autosegments on each tier, it allows for autosegments on one tier to be associated to autosegments on the opposite end of the other tier, resulting in massive line-crossing. Sagey (1986), Hammond (1988), and Bird and Klein (1990) all offer alternative definitions, but they are tied to specific semantic interpretations of association as overlap. As this dissertation is concerned with APRs as formal objects, it is necessary to have a clear formal definition of the NCC that applies to these objects. This has the advantage of holding no matter what semantic interpretation we wish to give APRs.

Coleman and Local (1991), in passing, give exactly such a definition of the NCC. First, they define crossed lines, which can then be used for a formal definition of the NCC.

(4.9) "Two association lines (a^i, b^j) and (c^i, d^j) are said to *cross* iff $a^i <^i c^i$ and $d^j <^j b^j$." (Coleman and Local, 1991, p.304)

(where a^i, c^i are elements on level L^i, b^j, d^j are elements on level L^j, and $<^i$ and $<^j$ are the orders on L^i and L^j, respectively)

(4.10) "[I]n an APR, there is no pair of association lines which cross."
(Coleman and Local, 1991, p.304)

Essentially, these two statements are an axiom stating the NCC. Axiom 5 translates it into the notation in this chapter.

Axiom 5 (The No-Crossing Constraint) *There are no nodes v, w, v', w' such that $\{v, v'\} \in E$, $\{w, w'\} \in E$, $v \prec w$, and $w' \prec v'$.*

Kornai (1991, 1995), in his formally grounded study of APRs, also states a very similar well-formedness axiom.

4.2.5 OCP

The final axiom we shall consider is the OCP. Chapter 2 discussed that, while there have been two versions of the OCP posited by researchers, under conditions of full specification they are identical. As for now we are assuming that Axiom 4 (which requires full specification) holds, we can use the simplest version of the OCP and know we are satisfying both. This simplest version, originally listed in (2.9) in Chapter 2 (p. 22), is repeated below in (4.11).

(4.11) The Obligatory Contour Principle (McCarthy, 1986, p.208)
At the melodic level, adjacent identical elements are prohibited.

Axiom 6 encodes the OCP as an axiom constraining the labeling of nodes. As only holds at the *melodic* level, let us choose only one of the tiers V_m for the OCP to hold.

Axiom 6 (OCP) *For one tier V_m corresponding to T_m in the tier partition, for all $x, y \in V_m$, $(x, y) \in A$ implies $\ell(x) \neq \ell(y)$.*

This states that, for one tier V_m, any nodes connected by an arc cannot have identical labels. As V_m is specified to correspond to T_m in a tier partition, we can consider that for a tier partition $T = \{T_m, T_t\}$, all graphs in $APG(\Sigma, T)$ conforming to Axiom 6 obey the OCP on the same tier. Note that for V_t, which represents the timing tier, the OCP does not necessarily hold. Thus, (4.5) obeys Axiom 6 when $T_m = \{L, H\}$ and $T_t = \{\sigma\}$, because the adjacent nodes labeled σ are on tier V_t.

Again, this definition only considers one melody/TBU tier pair. To extend Axiom 6 to multiple melody tiers, Σ and V could be partitioned into $\{T_m, T_t, ..., T_n\}$ and $\{V_m, V_t, ..., V_n\}$, respectively. In this case, Axiom 3 would specify a *single* tier in which all undirected edges must have one end. Axiom 6 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.

4.2.6 Interim conclusion: the top-down approach

Let $APG(\Sigma, T)$ refer to the set of graphs in $GR(\Sigma)$ satisfying Axioms 1 through 6 for a particular tier partition $T = \{T_m, T_t\}$ over Σ. As these Axioms have been shown to be identical to the conditions on APRs listed in Chapter 2, $APG(\Sigma, T)$ is thus the set of graphs corresponding to the set of APRs that are well-formed according to these conditions. As previously noted, this axiomatic approach of defining the set of well-formed APRs has been the traditional one, taken by Goldsmith (1976), Coleman and Local (1991), Sagey (1986), Bird and Klein (1990), Kornai (1991, 1995), and others.

However, as noted in the introduction, this view of APRs has no way of relating these representations to their surface string counterparts, which, as shall be seen in later chapters, is crucial for comparing the relative expressivity of APR grammars to string grammars (Chapter 6), as well as useful for understanding how these grammars can be learned from surface strings (Chapter 8).

The remainder of this chapter shows how APRs can instead be viewed as, like strings, the concatenation of a fixed set of primitives, and that Axioms 1 through 6 can be *derived* through this operation—in other words, the major properties of APRs can be seen as the result of the concatenation of a fixed set of primitives.

4.3 A Concatenation Operation for Graphs

Defining APRs axiomatically obscures their relationship to strings. The set of strings Σ^* over an alphabet Σ was in Definition 1 in Chapter 3 defined through the concatenation of symbols in Σ. The following section, §4.4, shows how a set $APG(\Gamma)$ of graphs can be similarly defined through the concatenation of a set of symbol-like graph primitives. The present section defines this operation.

First, §4.3.1 gives an informal overview of how concatenation works. §4.3.2 then gives the formal details of the operation.

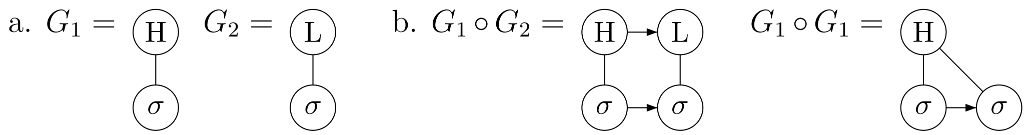
4.3.1 Concatenation of APGs: An informal overview

While the linear structure of strings allows for a straightforward notion of concatenation, graph structures include complex, non-linear relationships. A concatenation operation over graphs must somehow decide how the nodes in the two concatenated graphs will be related. Courcelle et al. (2012) discuss in general how this problem may be solved:

> We will consider two natural ways to "concatenate" two graphs. One way is to "glue" them together, by identifying some of their vertices. The other way is to "bridge" them (or rather, "bridge the gap between them"), by adding edges between their vertices. Clearly, to obtain single valued operations, we have to specify which vertices must be "glued" or "bridged". By means of labels attached to vertices, we will specify that vertices with the same label must be identified, or that edges must be created between all vertices with certain labels. (Courcelle et al., 2012, p. 6)

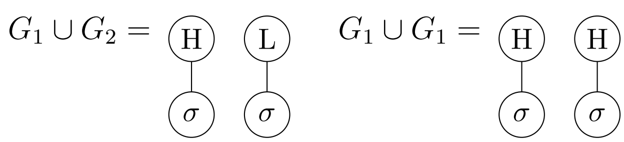
For APGs, the tier structure and labeling thus give us exactly the information we need to concatenate two graphs. The last member of each tier in the first graph will be 'bridged' via an arc to the first member of the respective tier in the second graph, unless they are identical melody units, which are instead 'glued,' or merged. Before delving into the details of the operation, (4.12) below gives an example.

In (4.12) below, G_1 and G_2 are primitives over an alphabet $\Sigma = \{\mathrm{H}, \mathrm{L}, \sigma\}$ and tier partition $T = \{T_m, T_t\}$ for $T_m = \{\mathrm{H}, \mathrm{L}\}$ and $T_t = \{\sigma\}$. The primitives G_1 and G_2 are given in (4.12a), and example graphs resulting from concatenating G_1 and G_2 are given in (4.12b). The concatenation operation for graphs is denoted with $\circ$.

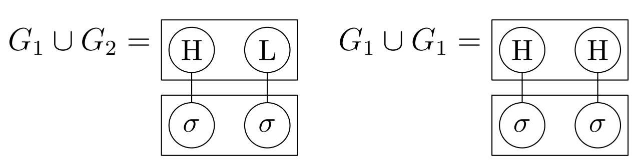
(4.12)


In (4.12b) there are two examples of concatenation. In the first, $G_1 \circ G_2$ results in the addition of arcs extending the nodes in G_1 to the nodes on their respective tiers in G_2. In contrast, $G_1 \circ G_1$ results in an arc between the σ nodes of the two instances of G_1 but the H nodes have been merged. This previews how the concatenation operation achieves autosegmental structure: the addition of arcs builds strings of timing tier units and distinct melody units, while the merging of like nodes results in one-to-many spreading relationships.

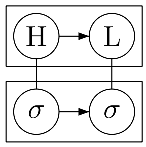
To go into a little more detail, yet still keep the discussion rather informal, concatenation can be defined as the following process. For two graphs to be concatenated, we take their *disjoint union* ($\cup$), which combines the two graphs, keeping their respective nodes distinct. Using the graphs above as examples,

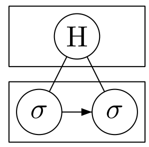
(4.13)


We then identify the *ends*, or pairs consisting of the last node on a tier in the first graph and the first node on that tier in the second graph. These pairs are highlighted in the examples below.

(4.14)


Whether or not a pair of end nodes is merged or bridged is decided based on their label. Pairs of melody nodes whose labels are not identical, and all pairs of timing tier nodes are bridged with an arc between them (as in $G_1 \circ G_2$ below). Pairs of melody nodes whose labels are identical are merged (as in the melody tier in $G_1 \circ G_1$ below).

(4.15) $G_1 \circ G_2 =$ 

$G_1 \circ G_1 =$ 

The following sections show how this operation preserves the crucial properties of APRs established in Chapter 2 and in §4.2 in this chapter—thus, the set of well-formed APRs can be seen as generated from the concatenation of a finite set of primitives. It also shows how other properties of this set of APRs can vary depending on the properties of these primitives.

First, however, the remainder of this section formally defines concatenation, drawing from the work of Courcelle et al. (2012) and Engelfriet and Vereijken (1997). While the content of §4.3.2 is not essential for understanding the ideas put forward in the remainder of the chapter, the proofs in §4.4 do make use of the definitions therein.

4.3.2 Concatenation of APGs: a formal excursus

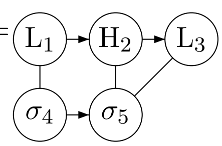
The concatenation operation is defined over graphs in $GR(\Sigma)$ for some tier partition $T = \{T_m, T_t\}$ of Σ. Integral to this definition are two functions, `first` : $GR(\Sigma) \times T \to \mathbb{N}$ and `last` : $GR(\Sigma) \times T \to \mathbb{N}$ which pick out the first and last nodes on a particular tier in a graph, respectively, if such nodes exist.[1] These will be the nodes which will either be 'bridged' or 'glued' by concatenation. If such nodes do not exist for a particular graph and tier, the functions are left undefined for that graph. `first` and `last` are defined formally below.

[1] The function notation `first` : $GR(\Sigma) \times T \to \mathbb{N}$ means that `first` takes as arguments a graph in $GR(\Sigma)$ and some T_i in tier partition T and returns a number in $\mathbb{N}$. Examples will be seen momentarily.

Definition 9 (first and last)

$$\begin{aligned}
\texttt{first}(G, T_i) &\stackrel{def}{=} v \in V \text{ s.t. } \ell(v) \in T_i, \forall v' \in V, \ell(v') \in T_i \rightarrow v \preccurlyeq v' \\
&\quad \text{if such a } v \text{ exists; undefined otherwise} \\
\texttt{last}(G, T_i) &\stackrel{def}{=} v \in V \text{ s.t. } \ell(v) \in T_i, \forall v' \in V, \ell(v') \in T_i \rightarrow v' \preccurlyeq v \\
&\quad \text{if such a } v \text{ exists; undefined otherwise}
\end{aligned}$$

As per Definition 9, $\texttt{first}(G, T_i)$ returns, if it exists, a node v whose label is in T_i such that for all other nodes whose label is in T_i there is either a directed path from v to v' or $v = v'$. $\texttt{last}(G, T_i)$ is similar, although it picks out a v such that all other v' whose labels are in the same tier have a either directed path to v or are equal to v. For example, consider our running example of $\Sigma = \{\mathrm{H}, \mathrm{L}, \sigma\}$ and tier partition $T = \{T_m, T_t\}$ for $T_m = \{\mathrm{H}, \mathrm{L}\}$ and $T_t = \{\sigma\}$. For G in (4.16) below, repeated from (4.5), $\texttt{first}(G, T_m) = 1$, $\texttt{last}(G, T_m) = 3$, $\texttt{first}(G, T_t) = 4$, and $\texttt{last}(G, T_t) = 5$.

(4.16) $G =$


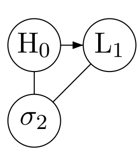
Concatenation can then be broken down into multiple steps as follows. First, we define the graph $G_{1,2}$ as the disjoint union of G_1 and G_2. For any two graphs G_1, G_2, let $G_{1,2}$ be defined as in Definition 10:

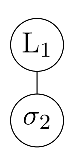
Definition 10 ($G_{1,2}$) *For any graphs G_1 and G_2,*

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

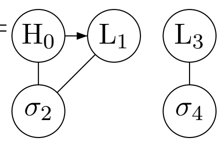
Where each $\cup$ denotes a disjoint *union.*

For example, take the following G_1 and G_2 in $GR(\Sigma)$.

(4.17) $G_1 =$


$G_2 =$


Since the indices overlap in G_1 and G_2, to get a disjoint union we can simply take an isomorphic copy of G_2 with different indices (recall that we consider isomorphic copies to be equal). Thus, in this case $G_{1,2}$ is the graph in (4.18), where the nodes $\{1, 2\}$ from G_2 have been re-indexed as $\{3, 4\}$, respectively.

(4.18) $G_{1,2} =$ 

The next step is to determine which nodes are 'glued' and which nodes are 'bridged'. Intuitively, we want to fuse the 'ends' of the tiers—the last nodes in G_1 and the first nodes in G_2 on each tier. First, we single out pairs comprising the last node in G_1 and the first node in G_2 for each tier. Call this set of pairs R.

Definition 11 **(R)** *For a graph $G_{1,2}$ made up of the disjoint union of graphs G_1 and G_2,*

$$R \stackrel{def}{=} \{ \quad (v, v') \in V_1 \times V_2 \mid \begin{array}{l} v = \texttt{last}(G_1, T_i), \\ v' = \texttt{first}(G_2, T_i), \\ \text{for some } T_i \in T \end{array} \quad \}$$

We then split R into two kinds of pairs: the ones we 'glue', or merge together, and the ones we 'bridge', or draw an arc between. Merging will occur between pairs of nodes in R whose labels are identical, excluding nodes whose labels are in T_t in the tier partition. Let this set of pairs be denoted R_{ID}. All other pairs in R not in R_{ID} will have arcs drawn between them. Let this set of pairs be denoted $R_{\overline{ID}}$.

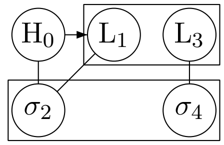
Definition 12 **(R_{ID} and $R_{\overline{ID}}$)**

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

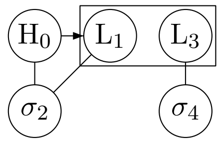
$$R_{\overline{ID}} \stackrel{def}{=} R - R_{ID}$$

The sets of pairs R, R_{ID}, and $R_{\overline{ID}}$ for the graph in (4.18) are listed below in (4.19). The pairs in each are highlighted as boxes in the diagram for the graph.

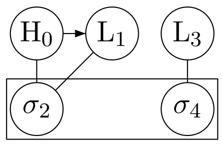
(4.19) a. $R = \{ \ (1,3), \ (2,4) \ \}$



b. $R_{ID} = \{ \ (1,3) \ \}$



c. $R_{\overline{ID}} = \{ \ (2,4) \ \}$



For a set of nodes V and a relation $R \subseteq V \times V$, Engelfriet and Vereijken (1997) define a new set of nodes V/R where each pair in R has been merged to a single node. This is done by defining V/R as a partition on V based on R.[2] A rough definition is given below.

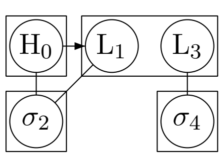
Definition 13 (V/R) *V/R is a partition on V such that*

a. *for all nodes $v_1 \in V$* not *in some pair in R, there is a node $v \in V/R$ corresponding to $\{v_1\}$. Let $[v_1]_R$ denote this v.*

b. *for all pairs $(v_1, v_2) \in R$, there is a node $v \in V/R$ corresponding to $\{v_1, v_2\}$. Let $[v_1]_R = [v_2]_R$ denote this v.*

To merge the nodes in R_{ID}, we simply take $V_{1,2}/R_{ID}$. The boxes in (4.20) below show the $V_{1,2}/R_{ID}$ for (4.18).

(4.20)



$V_{1,2}/R_{ID} = \{ \ \{0\}, \{1,3\}, \ \{2\}, \{4\} \ \}$

[2] For readers familiar with the terms, V/R is the quotient of V relative to the smallest equivalence relation containing R. Thus Definition 13 uses some notation usually associated with equivalence relations.

In (4.20), nodes 0, 2, and 4 from $G_{1,2}$ in (4.18) each get their own set in $V_{1,2}/R_{ID}$, because as seen in (4.19), none of them are in a pair in R_{ID}. Conversely, 1 and 3 are grouped together, as they form a pair in R_{ID}.

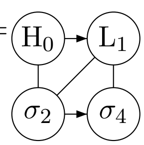
The final step in the concatenation of graphs G_1 and G_2 thus is to merge nodes in $V_{1,2}$ over R_{ID}, maintaining the edge, arc, and labeling relationships from $G_{1,2}$, and adding pairs from $R_{\overline{ID}}$ as arcs. This can be seen in the following definition for $G_1 \circ G_2$.

Definition 14 **($G_1 \circ G_2$)** *For any graphs G_1 and G_2,*

$$G_1 \circ G_2 \overset{def}{=} \langle\ V_{1,2}/R_{ID},\ E,\ A,\ \ell\ \rangle$$

Where $E = \{\{[v]_{R_{ID}}, [w]_{R_{ID}}\} \mid \{v, w\} \in E_{1,2}\}$,
$A = \{([v]_{R_{ID}}, [w]_{R_{ID}}) \mid (v, w) \in A_{1,2} \cup R_{\overline{ID}}\}$, and
$\ell([v]_{R_{ID}}) = \ell(v)$

The concatenation of G_1 and G_2 from (4.17) is given in (4.21).

(4.21) $G_1 \circ G_2 =$ 

Note that each node in (4.21) corresponds to a box in (4.20); 0 in (4.21) corresponds to $\{0\}$ in (4.20), 1 corresponds to $\{1, 3\}$, 2 corresponds to $\{2\}$, and 4 corresponds to $\{4\}$.[3] The edges in $G_1 \circ G_2$ respect the edges in $G_{1,2}$; $\{3, 4\}$ in $G_{1,2}$, for example, is preserved as $\{1, 4\}$ between the corresponding nodes in $G_1 \circ G_2$. Same goes for the arcs. Additionally, $G_1 \circ G_2$ contains an additional arc $(2, 4)$ corresponding to the pair $(2, 4) \in R_{\overline{ID}}$ from $G_{1,2}$.

We can now concatenate graphs in $GR(\Sigma)$. The following section utilizes this to derive a set of APGs following the axioms defined in §4.2.

[3] Recall that isomorphic graphs are considered equal; thus this re-indexing of the nodes is possible.

4.4 The Bottom-up Approach: APGs Through Concatenating Primitives

Having defined a concatenation operation for APGs, it is now possible to define $APG(\Gamma)$, a set of graphs derived from a set Γ of symbols. In other words, $APG(\Gamma)$ corresponds directly to the set Γ^* of strings over Γ. It is shown in this section that $APG(\Gamma)$ follows the axioms in §4.2 defined in that chapter for $APG(\Sigma, T)$. Thus, $APG(\Gamma)$ complements the axiomatic definition of the set of APRs in that it follows the same axioms, but it is directly relatable to strings.

4.4.1 APG primitives

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. Recall from Chapter 3 that Σ^*, the set of all strings over an alphabet Σ, was defined as follows (repeated from Definition 1 on p. 81):

Definition 15 *Given an alphabet Σ, $w \in \Sigma^*$ iff:*

- $w = \lambda$, *or*
- $w = u^\bullet\sigma$; $\sigma \in \Sigma$, $u \in \Sigma^*$

Nothing else is in Σ^*.

Given that we now have a way of concatenating graphs in $GR(\Sigma)$, the set of APGs can be defined similarly. We only need to define two notions, that of an alphabet for APGs and that of the *empty graph*. We can define the empty graph, denoted G_λ, as follows:[4]

Definition 16 (The empty graph) *The* empty graph G_λ *is the graph whose sets of nodes, edges, arcs, and labeling function are all empty;*

$$G_\lambda = \langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle$$

[4] Some researchers advise against using an empty graph with 0 nodes (Harary and Read, 1974), but for the purposes here, it will be seen to be extremely useful. Such an empty graph is also made use of in Engelfriet and Vereijken (1997).

The following theorem states that G_λ is the identity element for the concatenation operation.

Theorem 2 (G_λ is identity with respect to concatenation) *For any graph G, $G_\lambda \circ G = G \circ G_\lambda = G$.*

Proof: (sketch) This almost immediately follows from the fact that the concatenation of two graphs is a modification of their disjoint union. Thus, no new nodes are added to G. As there are no nodes to merge or to bridge, no other modifications are made to G, and so the resulting graph is identical to G. $\square$

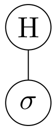
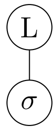
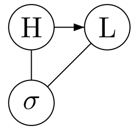
The next step is to define an alphabet of *graph primitives* out of which we build the set of APGs. This corresponds to the alphabet of symbols for strings. To ensure that this set of APGs conforms to the structural properties of APRs, it is necessary that the graph primitives in our alphabet meet certain properties. These properties are given below in Definition 17.

Definition 17 (APG primitive) *Over an alphabet Σ and tier partition $T = \{T_t, T_m\}$, an* APG primitive *is a graph $G \in GR(\Sigma)$ which has the following properties:*

a. V_t is a singleton set $\{v_t\}$
b. The arcs A form a total order over the nodes of V_m
c. All $e \in E$ are of the form $\{v_m, v_t\}$, where $v_m \in V_m$

The parts of the definition can be interpreted as follows. Definition 17a requires that in a primitive, there can only be one node on the timing tier V_t. (Recall that T_t is the set of symbols that can appear on the timing tier, and V_t is the corresponding set of nodes in a graph labeled with symbols from T_t.) The melody tier, V_m, may consist of multiple nodes, but Def. 17b requires that all of the arcs in A form a total order over V_m. Def. 17c requires that all of the edges are between the node in V_t and the nodes in V_m.

Some example APG primitives over the alphabet $\Sigma = \{\mathrm{H}, \mathrm{L}, \sigma\}$ partitioned into tiers $T_m = \{\mathrm{H}, \mathrm{L}\}$ and $T_t = \{\sigma\}$ is given below in (4.22). In these and following graphs, the particular indexing of the nodes is not important, and thus shall be omitted.

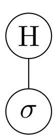
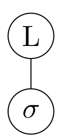
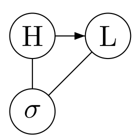
(4.22) a.  b.  c. 

Note in (4.22), all the graphs have only one node labeled σ, thus they satisfy Def. 17a. The graph in (4.22c) has two nodes with labels from T_m, but note that there is a single arc ordering them. Thus all of the graphs in (4.22) satisfy Def. 17b. Finally, all of the edges in all of the graphs in (4.22) satisfy Def. 17c.

We can then explicitly relate such a set of graphs to an alphabet of string symbols. This alphabet will be referred to as Γ (to differenatiate it from the Σ of the graph labels).

Definition 18 (Alphabet of APG primitives) *An* alphabet of APG primitives *over* $GR(\Sigma)$ *is a finite set* Γ *of symbols and a naming function* $g : \Gamma \to GR(\Sigma)$ *such that for all* $\gamma \in \Gamma$, $g(\gamma)$ *satisfies Definition 17 for an APG primitive.*

For example, let $\Gamma = \{\mathrm{H}, \mathrm{L}, \mathrm{F}\}$. The graphs in (4.22) can be related to the symbols in Γ as in (4.23) below. This pair Γ and g is thus an alphabet of APG primitives.

(4.23) $g(\mathrm{H}) =$  $g(\mathrm{L}) =$  $g(\mathrm{F}) =$ 

As is perhaps clear, the symbols in Γ can represent the surface realizations of TBUs, where their corresponding graphs represent their underlying autosegmental structure. For example, if we interpret F as a falling tone, then $g(\mathrm{F})$ in (4.23) represents the autosegmental decomposition of that tone into a syllable associated with an HL sequence of tones. The last step is to show that this relationship holds over *strings* in Γ as well.

The strings in Γ^* represent a class of graphs, denoted $APG(\Gamma)$. $APG(\Gamma)$ is defined by extending g to strings in Γ^* using the concatenation operation $\circ$ for graphs.

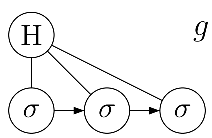
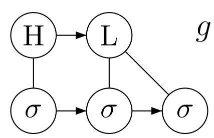
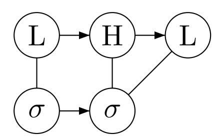
Definition 19 **($APG(\Gamma)$)** *For an alphabet of APG primitives Γ with naming function g, extend g to strings in Γ^* as follows. For $w \in \Gamma^*$, $g(w) \stackrel{def}{=}$*

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

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

Note that this definition is written parallel to Definition 15; thus, $APG(\Gamma)$ can be defined parallel to a set of strings.

In terms of interpreting Γ as linear realizations of tones over TBUs and APG primitives as their autosegmental counterparts, this means that *linear strings of TBUs can be directly related to their autosegmental interpretations.* For example, take the strings $\mathrm{HHH}, \mathrm{HLL}, \mathrm{HF} \in \Gamma^*$. By the definitions introduced so far,

(4.24) $g(\mathrm{HHH}) =$  $g(\mathrm{HLL}) =$  $g(\mathrm{HF}) =$ 

§4.5 goes into more detail about interpreting $APG(\Gamma)$ for different Γ and g pairs as sets of APRs. First, however, to show that we can truly treat any such $APG(\Gamma)$as a set of APRs, it is important to establish that it obeys the key properties of APRs.

4.4.2 Properties of $APG(\Gamma)$

For any alphabet of APG primitives Γ and g following definition Definition 18, $APG(\Gamma)$ follows all of the axioms in §4.2 defined for APRs. The following theorems establish this. Sketches of their proofs are given here; for full proofs, see (Jardine and Heinz, 2015). See (Jardine and Heinz, 2015) also for a proof that concatenation is associative over any such $APG(\Gamma)$. This is an important property for relating $APG(\Gamma)$ to strings, but has been omitted here for the length and technical nature of the proof.

The first theorem states that any graph in $APG(\Gamma)$ follows the basic structure of APRs; it has two tiers whose labels are in disjoint subsets of the labelling alphabet Σ, and all undirected edges are between nodes in these two tiers.

Theorem 3 ($APG(\Gamma)$ obeys the basic properties of APRs) *For any $G \in APG(\Gamma)$, G satisfies Axiom 1 (that $\sim_A$ partitions V into at most two sets V_t and V_m which are totally ordered by the arcs), 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: (sketch) That V_t and V_m are totally ordered follows from the fact that parts (a) and (b) of Definition 17 ensure that the tiers in each primitive are totally ordered and the fact that concatenation preserves this order on each tier (because merging or bridging at the edges of the graph does not change the total order induced by the arcs).

That G satisfies Axiom 2 follows directly the fact that concatenation only adds arcs between nodes whose labels are in the same $T_i \in T$.

That G follows Axiom 3 follows directly from Part (c) of Definition 17 and the fact that concatenation adds no new undirected edges to E. □

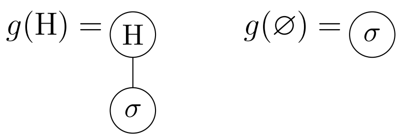
The following theorem regards Axiom 5, or the NCC. Axioms 4 and 6, stating full specification and the OCP, will be discussed in a moment, as they are dependent on the content of the alphabet of APG primitives.

Theorem 4 ($APG(\Gamma)$ obeys the NCC) *For any $G \in APG(\Gamma)$, G satisfies the NCC (Axiom 5: There are no nodes v, w, v', w' such that $\{v, v'\} \in E$, $\{w, w'\} \in E$, $v \prec w$, and $w' \prec v'$.).*

Proof: (sketch) By part a of Definition 17, there is only one timing tier node v_t per APG primitive. As such, any primitive clearly satisfies the NCC. Any edges resulting from the concatenation of another primitive are to a node v'_t *following* v_t in the order of arcs, as concatenation adds an arc from v_t to v'_t (and concatenation does not add edges). This holds for any number of concatenations, and so the NCC is preserved for any graph in $APG(\Gamma)$. □

The final two axioms are dependent on the graphs in the alphabet of APG primitives. For example, the reader can confirm both of the graphs in (4.25) follow Definition 17 of an APG graph primitive, however one has no edges.

(4.25)



Axiom 4 states that for all $v \in V$, there must be an edge $\{v, w\} \in E$ for some other $w \in V$. In autosegmental terms, this is full specification—each autosegment is associated to at least one other autosegment. However, if $\Gamma = \{\mathrm{H}, \varnothing\}$ and g is as defined in (4.25), then $APG(\Gamma)$ clearly fails Axiom 4, as $g(\varnothing)$ includes a timing tier node which has no edges. As concatenation does not add edges or merge timing tier nodes, any graph corresponding to a string containing $\varnothing$ also fails Axiom 4.

However, if $\Gamma = \{\mathrm{H}, \mathrm{L}, \mathrm{F}\}$ and g is defined as in (4.23), then it can be shown that any graph in $APG(\Gamma)$ does. Theorem 5 states this.

Theorem 5 (Concatenation preserves full specification) *If, for all $\gamma \in \Gamma$, $g(\gamma)$ obeys Axiom 4, then for all $G \in APG(\Gamma)$, G obeys Axiom 4.*

Proof: (sketch) This follows from the fact that neither node-merging nor bridging in concatenation has any effect on existing edges in the primitives. □

The axiom stating the OCP is the same way. *If* all of the APG primitives obey the OCP, then the OCP is preserved for $APG(\Gamma)$. Some examples in which the OCP is not preserved in the primitives are given in §4.5.2.

Theorem 6 (Concatenation preserves the OCP) *If $g(\gamma)$ for all $\gamma \in \Gamma$ satisfy the OCP (Axiom 6), then for any $G \in APG(\Gamma)$, G satisfies Axiom 6.*

Proof: (sketch) This follows directly from the fact that concatenation merges adjacent end nodes on V_m. □

The conditional nature of these final two theorems allows Axioms 4 and 6 to be violable based on the chosen set of primitives—this allows full specificity and the OCP to be violated on a language- or analysis-specific basis through the choice of primitives. Examples will be given in §4.5.

It should be noted that while this discussion has been limited to two tiers, the core properties of concatenation explored here will likely extend to representations with more tiers. Extending the analysis to feature geometry (Clements and Hume, 1995; Sagey, 1986), for example, provides an interesting complication, as which association lines denote not only associations between featural autosegments and timing tier nodes but also link featural autosegments and 'organizational' nodes (such as PLACE for the features LABIAL, CORONAL and DORSAL). 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. However, the central concepts in graph concatenation of 'gluing' (to derive the OCP) and 'bridging' nodes will still hold, and thus it will be likely straightforward to derive the crucial APR properties over multiple tiers similar to the way it was done for two tiers here.

4.4.3 Discussion: The bottom-up versus the top-down approach

Traditionally, APRs have been defined in a 'top-down' manner, by specifying constraints on graph-like structures, as was made explicit in the axioms for $APG(\Sigma, T)$ in §4.2. However, Theorems 3 through 6 have just shown that a bottom-up approach, i.e. concatenation of a finite set of primitives, yields a set $APG(\Gamma)$ of APGs which also follows these axioms.

As the two approaches achieve the same axioms, they can be seen as complementing each other. However, there are some reasons for preferring the bottom-up approach. One, it naturally captures some empirical facts about tone patterns, namely that the length of contour melodies and strings of floating tones are always bounded for a particular language. More importantly, the bottom-up approach allows us a principled way of directly comparing autosegmental representations to their string counterparts. This answers the question, raised in the previous chapter, of how to enhance string representations in order to achieve the right level of expressiveness for tone patterns. Both of these points are discussed in more detail in the following sections.

4.5 Empirical Consequences of Concatenation

The preceding section discussed the properties of APGs and concatenation in abstract terms. This section discusses more concretely how viewing tonal representations as the concatenation of graph primitives naturally explains some empirical facts. It is important to keep clear that, just as with the axiomatic approach in §4.2, $APG(\Gamma)$ is primarily intended to represent the *universal* set of well-formed APRs, although some variation with respect to the primitives languages choose from will be discussed in §4.5.3 (and then again in Chapter 5, §5.5). Language-specific constraints regarding the well-formedness of melodies and association will be discussed in detail in the next two chapters.

4.5.1 Multiple association

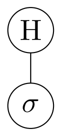
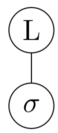
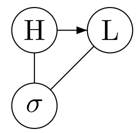
The cornerstone of autosegmental analyses, as illustrated in Chapter 2, is the idea of a single autosegment being associated to multiple other autosegments. Concatenation achieves this asymmetrically; multiple association of melody units to timing units can be repeatedly generated through concatenation, whereas multiple association of timing units to melody units must be specified in the primitives.

To illustrate, consider patterns over the H and HL melodies in Mende, originally discussed in §2.1.1 (p. 16).

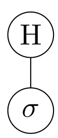
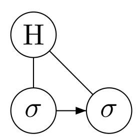
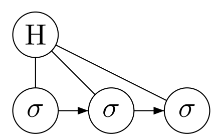
(4.26) Mende H and HL melody words (repeated from (2.1))

a.	kɔ́	‘war’	b.	pɛ́lɛ́	‘house’	c.	háwámá	‘waist’
		H			HH			HHH
d.	mbû	‘owl’	e.	ngílà	‘dog’	f.	félàmà	‘junction’
		F			HL			HLL

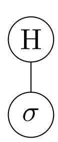
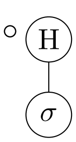
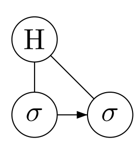
In terms of toned syllables, we have strings of H, L, and F-toned syllables. Let $\Gamma = \{\mathrm{H}, \mathrm{L}, \mathrm{F}\}$ be an alphabet of APG primitives with g defined as it was in (4.23):

(4.27) $g(\mathrm{H}) =$  $g(\mathrm{L}) =$  $g(\mathrm{F}) =$ 

The set of graphs $APG(\Gamma)$ generated from the concatenation of these primitives contains the APGs corresponding to the autosegmental representations to the strings in (4.26). As discussed in the previous section, g gives us an explicit way of relating these strings of syllables to APGs. For example, the H melody strings are mapped by g to the following APGs:

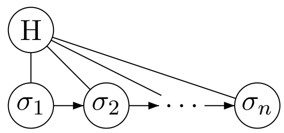
(4.28) $g(\mathrm{H}) =$  $g(\mathrm{HH}) =$  $g(\mathrm{HHH}) =$ 

The first example, $g(\mathrm{H})$, is straightforward: it is just the graph primitive associated with the symbol H. For the string HH of two H-toned syllables, $g(\mathrm{HH})$ is the concatenation of two $g(\mathrm{H})$ graphs, $g(\mathrm{H}) \circ g(\mathrm{H})$. In this situation, by the definition of concatenation, the H nodes on the melody tier in each $g(\mathrm{H})$ primitive are ‘glued’ together, as illustrated below:

(4.29) $g(\mathrm{HH}) = g(\mathrm{H}) \circ g(\mathrm{H}) =$  $\circ$  $=$ 

This results in a single H node being associated to two σ nodes. Likewise, g(HHH) results in a single H node associated to three σ nodes. In fact, there is no bound on the number of timing tier units to which a single melody tier unit may be associated. For any n, we can generate a H node associated to n σ nodes by taking $g(\mathrm{H}^n)$:

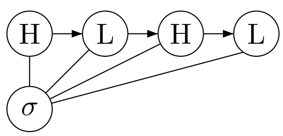
(4.30) $g(\mathrm{H}^n) =$



This is a welcome result: such ‘unbounded spreading’ is well known in tone (Yip, 2002; Hyman, 2011b).

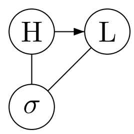
In contrast, it is impossible to generate the following APG by concatenating primitives in Γ:

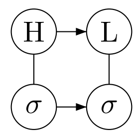
(4.31) *

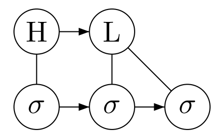


This is because concatenation only ‘bridges’ timing tier units (in this case, σ nodes), and does not glue them. Thus, unless a contour appears in the set of APG graph primitives, it does not appear in $APG(\Gamma)$. This matches empirical fact: languages may have unbounded spreading, but they do not have ‘unbounded contouring’. For example, Mende has forms like (4.30) for at least $n = 4$ (Dwyer, 1978), but no forms like (4.31) (it does have R-F contours, which will need to be added as a primitive. See §5.1, p. 150, in Chapter 5). Thus, viewing the set of autosegmental representations as the concatenation of a finite number of primitives makes the correct prediction that any language will have a finite number of possible contours, because each contour needs to be specified as one of a finite set of primitives.

This brings us to the HL melodies. The APGs corresponding to the HL melody forms from (4.26) are as follows:

(4.32) $g(\mathrm{F}) =$ 

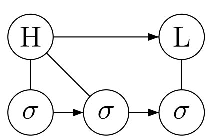
$g(\mathrm{HL}) =$ 

$g(\mathrm{HLL}) =$ 

The contour F is realized simply as the graph primitive $g(\mathrm{F})$. The string HL is realized as $g(\mathrm{H})$ and $g(\mathrm{L})$ bridged on each tier—again, gluing of nodes does not occur because H and L nodes have nonidentical labels. However, gluing does occur in the last two L nodes of $g(\mathrm{HLL})$, resulting in multiple association to the second two σ nodes.

These APGs raise two important points. One, some may object to this method of representation, as it generates $g(\mathrm{F})$ in a way distinct from how $g(\mathrm{HL})$ is generated, even though both APGs have the same melody. However, the purpose of Γ and g is to relate surface strings to their autosegmental representations—under this view, g does in fact relate F with HL, or any string whose corresponding graph has a HL melody.

Two, note that $g(\mathrm{HHL})$ is also a graph in $APG(\Gamma)$ whose melody is HL:

(4.33) $g(\mathrm{HHL}) =$ 

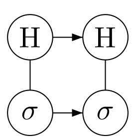
As was discussed in 2.1.1 in Chapter 2, HHL is not a valid string of syllables in Mende (see p. 17), and so (4.33) is not a valid APG in Mende (recall that Mende has been analyzed with left-to-right association). However, this is not a problem, as $APG(\Gamma)$ is the *universal set* of well-formed APGs given Γ. Importantly, *concatenation does not represent the grammar of a particular language.* Language-specific patterns over APGs can be viewed as *restrictions* on $APG(\Gamma)$, which will be defined in the following chapter, Chapter 5, by extending the logical constraints over strings in Chapter 3 to graphs. This is, on a fundamental level, no different than autosegmental analyses in more traditional frameworks: language-specific rules or constraint rankings produce a set of language-specific APRs which are a subset of the universal set of possible well-formed APRs (because they conform to the tier structure, NCC, etc.).

This raises a question: is there a single, universal Γ and g? Or can Γ and g be language specific? The strongest possible hypothesis is that all languages must choose from a very restrictive Γ that allows for $APG(\Gamma)$ to follow all of Axioms 1 through 6. Indeed, the patterns surveyed in Chapters 5 and 6 are all subsets of such a $APG(\Gamma)$ (to be defined in Chapter 5). However, it is well-known that there are clear cases in which Axiom 4 (full specification) and Axiom 6 (the OCP) are not obeyed. The following subsections discuss these cases. The conclusion is that languages may pick from subset of a universal Γ, with g largely being universal save for one complicating issue of downstep.

4.5.2 OCP violations

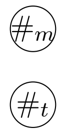
It was just seen how the association of a melody tier node to multiple timing tier nodes is generated through concatenation by the merging of like melody tier nodes. This is because concatenation preserves the OCP, as proven in §4.4.2. Thus, with an alphabet of APG primitives as in (4.27), concatenation cannot produce a graph as in (4.34) (and thus it is not a member of $APG(\Gamma)$).

(4.34)



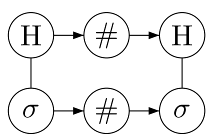
Is such a representation necessary? To deal with OCP violations *across* morpheme boundaries we can add a morpheme boundary symbol # to Γ whose graph interpretation inserts a morpheme boundary on both the timing and melody tier (as is commonly done in autosegmental analyses).[5]

[5] Note that there is no association line between the two nodes, as is sometimes included (see Goldsmith (1976) and Pulleyblank (1986) for discussion). In Goldsmith (1976), the primary motivation for including association lines between morphological markers is to block association past them. Note that, assuming concatenation, association through boundary markers like in (4.35) is impossible even without this association line. Morpheme boundaries over which association can occur must thus be represented by primitives with no node on the melody tier. This difference will be discussed in

(4.35) $g(\#) =$ 

Here, the m and subscript marks that $\#_m$ behaves as part of the melody tier, and thus will be bridged through concatenation to other elements on this tier. Likewise, $\#_t$ behaves as part of the timing tier. For typographical clarity, this part of the label will be omitted from subsequent diagrams.

With a morpheme boundary primitives as in (4.35), an OCP violation across morpheme boundaries will be in $APG(\Gamma)$, as in the following graph $g(\text{H\#H})$:

(4.36) $g(\text{H\#H}) =$ 

As for OCP violations within morpheme boundaries, Hyman (2014) states (p. 371) that "[c]ases of tautomorphemic OCP violations are extremely rare and include ... a few tone cases, e.g. Shambala [a.k.a. Kishambaa–AJ] (Odden, 1982, 1986) ... While such cases need to be scrutinized carefully, they are rare, and alternative interpretations are sometimes available." Hyman's 'Shambala' is the case in Kishambaa (Odden, 1986) mentioned in §2.1.2.2 in Chapter 2 (p. 24). For at least the tautomorphemic OCP violation in Kishambaa, we can take advantage of the fact that OCP violations are signaled by downstep in the surface string. As detailed in Odden (1986), Kishambaa nouns show the following contrast:

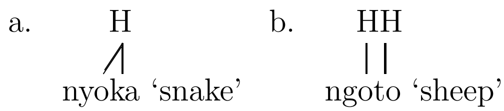
(4.37) a. nyóká 'snake' (HH)

b. ngóꜜtó 'sheep' (HꜜH)

Odden (1986) posits the following autosegmental diagrams for these forms.

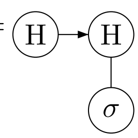
---

more detail in Chapter 6 with respect to competing representational assumptions of Wan Japanese.

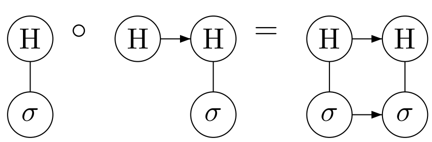
(4.38) 

He concludes that these are true OCP violations, and not the result of a latent floating L tone (floating tones will be discussed momentarily), because there is no independent evidence for such a floating tone. Thus, at least for Kishambaa, it appears that $APG(\Gamma)$ should include a graph like (4.34).

Crucially, this tautomorphemic OCP violation is marked in the surface string. Thus, to obtain a graph like (4.34) we can add a ${}^{!}$H symbol representing a downstepped H to Γ. This can then be pointed to the OCP-violating graph primitive in (4.40) below corresponding to a H tone following another H.

(4.39) $g({}^{!}\mathrm{H}) =$ 

The graph $g(\mathrm{H}{}^{!}\mathrm{H})$, corresponding to the surface string for (4.37b) ngó${}^{!}$tó ‘sheep’, is thus equal to (4.34).

(4.40) 

In conclusion, because OCP violations can be derived from symbols in the surface string, it appears that they (at least for tone) can be subsumed into a universal set of APGs $APG(\Gamma)$ for a Γ including # and ${}^{!}$H with g for these symbols as defined above. However, a factor complicating downstep is that it can also be signaled by floating tones, as shall now be discussed.

4.5.3 Floating tones

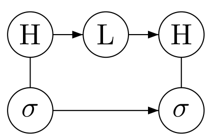
‘Floating tones’ are tones autosegments which are not associated to any timing tier unit (and thus not directly realized in the surface string). They are often associated with downstep, as in Dschang-Bamileke (Pulleyblank, 1986). The string of syllables in the following phrase is, at a slower speech rate, pronounced as HLH.

(4.41) sə́ŋ è sə́ŋ (HLH)
‘bird of bird’

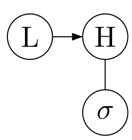
At faster speech rates, the vowel of the morpheme /è/ ‘of’ is elided, but the presence of its tone is still felt, as the tone of the second /sə́ŋ/ ‘bird’ is downstepped:

(4.42) sə́ŋ s$^{!}$ə́ŋ (H$^{!}$H)
‘bird (of) bird’

The accepted analysis of this downstep is that the L tone of /è/ ‘of’ is still present in the representation, although it has lost its TBU. As an APG, the representation of (4.42) is as follows:

(4.43)


Again, this graph is not in $APG(\Gamma)$ for any Γ we have defined thus far. However, we can again take advantage of the downstepped $^{!}$H and assign it a primitive representing a H tone following a L tone.

(4.44) $g(^{!}\mathrm{H}) =$


The graph (4.43) is thus equal to $g(H^{!}\mathrm{H})$. This makes explicit the correspondence between the differences in the autosegmental representation and difference in the HLH and H$^{!}$H strings, which is not done in Pulleyblank (1986) (and no other author, to the best of my knowledge).[6]

[6] However, note that $g(\mathrm{L}^{!}\mathrm{H}) = g(\mathrm{LH})$; i.e., this difference between strings is lost in the autosegmental representation. As shall be discused momentarily, a full representational theory of downstep is beyond the scope of this dissertation.

This raises another issue: if g is universal, then it can only point to a single primitive. The problem of $g(^{!}\text{H})$ possibly pointing to a floating tone or an OCP violation (as in (4.40)) can be resolved by rejecting Odden (1986)'s argument that positing a floating L tone to explain downstep in Kishambaa is *ad hoc*. If $g(^{!}\text{H})$ universally points to a floating L tone, then we appear to have a universal way of relating strings to APGs. This would thus constitute a theory-internal counterargument to Odden's claim. Alternatively, we could argue against the floating tone analysis in Dschang, and say that the downstep is caused by an OCP violation over the phrasal domain.

However, this and other issues related to floating tones and downstep will not be addressed in detail here. One other issue is down*drift*, the automatic lowering of successive H tones in a domain (Pulleyblank, 1986; Yip, 2002). This dissertation will not attempt a representational theory of downstep, as its main focus, to be seen in the following chapters, is to show that tone patterns can be described by fundamentally local constraints over APGs. Regardless of what a full representational theory of downstep/downdrift is, this locality result will likely not change.

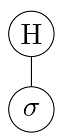
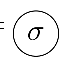
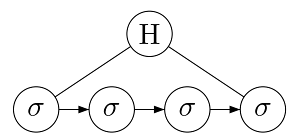
4.5.4 Interim conclusion: concatenation and tone

This section has shown that the concatenation operation defined in §4.3 generates sets of APGs which can model attested tone patterns. As a theory of representation, $APG(\Gamma)$ is extremely restrictive, perhaps too restrictive. For example, it does not straightforwardly account for downstep. However, its advantage is made clear: it provides a concrete way of relating surface strings to APRs. It will also be seen as a sufficient theory of representation for the tone patterns explored in the following chapters. The extensions hinted at for the OCP and downstep discussed in the preceding sections can be explored in future work.

One additional type of APR not covered in this section which can be addressed in future work is one which uses underspecification of TBUs. This is common in systems which have what is called a 'privative' distinction between specified (commonly with a H tone) and unspecified TBUs. This is usually invoked in underlying or intermediate representations of tone. One example is this intermediate representation from Hewitt and Prince (1989)'s analysis of Northern Karanga Shona, first given in Chapter 2 in (2.63) in §2.3.1.1 (p. 54):

(4.45)
$$\begin{array}{cccc} \text{H} & & \text{H} & \\ | & & | & \\ \sigma & \sigma & \sigma & \sigma \end{array}$$

A corresponding APG cannot be generated with the concatenation operation defined in this chapter. The Γ from (4.25) allows for H-specified syllables and underspecified syllables, but any two H nodes would end up on the 'ends' of the melody tier and be merged by concatenation. This can be seen in the following graph $g(\text{H}\varnothing\varnothing\varnothing\text{H})$:

(4.46) $g(\text{H}) =$  $g(\varnothing) =$  $g(\text{H}\varnothing\varnothing\varnothing\text{H}) =$ 

Thus, the concatenation operation defined here is inadequate for representations such as in (4.45). However, such representations could be generated by a *second* concatenation operation which only bridges nodes, and does not glue them. Exactly such an operation is discussed with respect to underlying APRs in Chapter 7, §7.3.1, 228.

4.6 Conclusion

This chapter served several important purposes. The overarching goal was to formally define a universal set of autosegmental representations, which can then in the following chapter lead to a theory of local grammars for autosegmental representations. The first part of this chapter formally defined the properties that are important to APRs in the traditional, axiomatic approach.

The second part of the chapter then showed how these properties can emerge from a concatenation operation over graphs. In particular, the result here is consistent with Coleman and Local (1991)'s observation that if we have an order on associations derived from the order on the respective tiers, then the NCC is unnecessary. It was then shown how this concatenation operation captures the basic tonal ideas of multiple association, and specifically that it naturally captured the idea that unbounded spreading is empirically common, but unbounded contouring is unattested.

In theoretical terms, this notion of concatenation is also important in that it will be made use of besides simply defining the set of well-formed APRs. Chapter 6 shall use concatenation to directly compare APR grammars to string grammars, which is necessary to show that local APR grammars can describe the long-distance tone patterns Chapter 3 showed were beyond the power of local (and tier-based local) string grammars. Chapter 7 will make extensive use of concatenation to develop representations for phonological transformations from underlying to surface forms. Finally, Chapter 8 will show how concatenation allows for learning APR grammars directly from strings.

Of course, there are many remaining questions, some of which have been hinted at throughout this chapter. For one, this chapter has shown how important autosegmental properties can be seen as the result of concatenation, but to what extent are the top-down and bottom-up approaches equivalent? That is, do they define the same set of structures? As illustrated in §4.5.2 and §4.5.3, it depends on what axioms are followed in the top-down approach, and the properties of the primitives used in the bottom-up approach. A full survey of under what conditions axiomatic and concatenation-based definitions describe the same set of structures will thus be left to future work. Additionally, in §4.4.2, it was outlined how concatenation might be extended to segmental feature-geometric representations. How concatenation can be fruitfully applied to these and other kinds of phonological representations remains an interesting open question.

Finally, as noted in §4.5.1, it is important to keep in mind that concatenation does *not* represent the derivation of language-specific association paradigms. As reviewed in Chapter 2, languages display a wide range of patterns of associations from tones to TBUs. Thus, they cannot be derived from a single, universal operation. Instead, the following two chapters show how these language-specific patterns can be defined by generalizing the idea of logical constraints from Chapter 3 from graphs to strings. The conclusion will be that *banned substructure constraints*—which are the most restrictive kind of constraint over graphs as well—can capture these language-specific patterns.

Chapter 5

BANNED SUBGRAPH GRAMMARS AND TONE MAPPING

This and the following chapter contain the central result of this dissertation, which is that the language-specific variation in tone patterns reviewed in Chapter 2 is fundamentally *local* over APRs in the computational sense defined for strings in Chapter 3. Chapter 2 introduced two important aspects of variation in tone patterns with respect to autosegmental representations: directional, quality-specific, and positional restrictions on tone-TBU associations and long-distance generalizations arising from interactions between the melody and timing tiers. It then showed how previous analyses in derivational frameworks and Optimality Theory had difficulties in capturing this variation, particularly in the proliferation of language-specific rules and association paradigms in the case of derivational theories and in the through the over- and under-generation of ALIGN constraints.

The goal of this chapter is to introduce a theory of graph constraints which is *local* in the way defined over strings in Chapter 3 and contrast it, with respect to variation in tone-mapping patterns, with the analyses in derivational frameworks and Optimality Theory. In Chapter 3, we saw how banned substructure constraints form a restrictive theory of patterns over strings. This chapter extends this idea to graphs, by defining *conjunctions of negative literals over graphs*. It then shows how these negative literals can describe the language-specific association patterns of Mende, Hausa, Kukuya, and Northern Karanga Shona discussed in Chapter 2.

This theory thus compares favorably to previous explanation of these patterns, as it is a *local* theory of association patterns over APRs, without resorting to adhoc rules or globally-evaluated directionality paradigms or ALIGN constraints used in rule-based and optimization-based theories of autosegmental phonology. It also makes clear, yet *restrictive* typological predictions about the type of association constraints that are possible in tonal phonology. This theory is restrictive because, like the banned substructure constraints for strings introduced in Chapter 3, the well-formedness of a structure is evaluated only according to the well-formedness of its individual substructures. Thus, a theory of well-formedness based on CNLs over graphs does not predict patterns that count over representations, like ALIGN. One undesirable typological prediction of CNLs over graphs, contour-specific directionality generalizations, is discussed. However, it is shown how such a constraint can be eliminated from the typology by further constraining the theory to only consider particular *types* of subgraphs—in particular, subgraphs which do not contain any cycles. This issue highlights an additional strength of banned subgraph grammars: we can get an idea of the types of substructures over which humans appear to be evaluating phonological representations.

The structure of this chapter is as follows. First, §5.1 lays out the problem of picking out subsets of the APGs definable by the concatenation operation discussed in Chapter 4, and then 5.2 shows how this problem can be solved by a logical language of conjunctions of negative literals where the literals are subgraphs. §5.3 then shows how statements in this language can describe the tone mapping patterns reviewed in Chapter 4. §5.4 compares this analysis with previous analyses in derivational and OT frameworks, and §5.5 shows how restricting the definition of subgraph literals can further restrict the theory. §5.6 concludes.

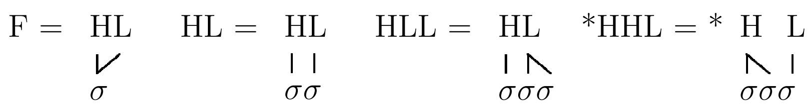
5.1 Specifying Graph Sets

The previous chapters showed a way of generating a 'universal' set of APGs, i.e., the set of all APGs that can be generated from a particular set of primitives. Any language-specific association pattern is thus a *subset* of this set of universal APGs. For example, recall the basic directional Mende pattern originally described in Chapter 2. The pattern exhibits five melodies, H, L, HL, LH, and LHL, and both contours and plateaus of like-toned syllables appear on the right edge of the word. Here I extend the general pattern beyond three syllables (recall the discussion on the infinitude of phonological patterns in Chapter 3, §3.2, p. 82.) and represent a syllable with a R-F falling contour with a single C. Call this pattern 'Simple Mende' (c.f. the pattern taking into account less-attested melodies and association patterns discussed in Chapter 2, §2.2.1.3, p. §31).

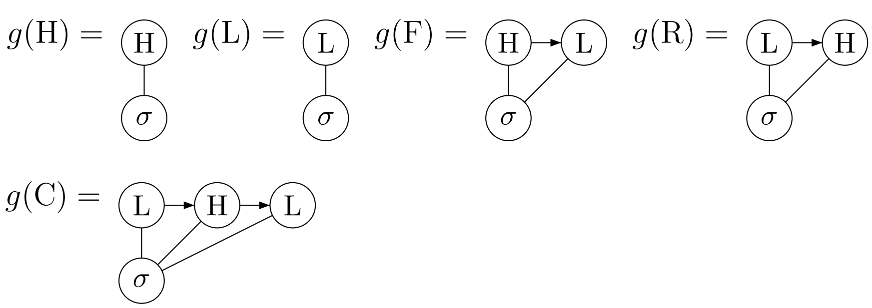
(5.1) Simple Mende string patterns

(H melody)	H,	HH,	HHH,	HHHH,	...
(L melody)	L,	LL,	LLL,	LLLL,	...
(HL melody)	F,	HL,	HLL,	HLLL,	...
(LH melody)	R,	LH,	LHH,	LHHH,	...
(LHL melody)	C,	LF	LHL,	LHLL,	...

Again, in autosegmental terms the generalization is that multiple association can only occur on the right edge of the word. Example APRs from HL melody forms are repeated below from (2.3) and (2.7) in Chapter 2 (pp. 18 and 21).

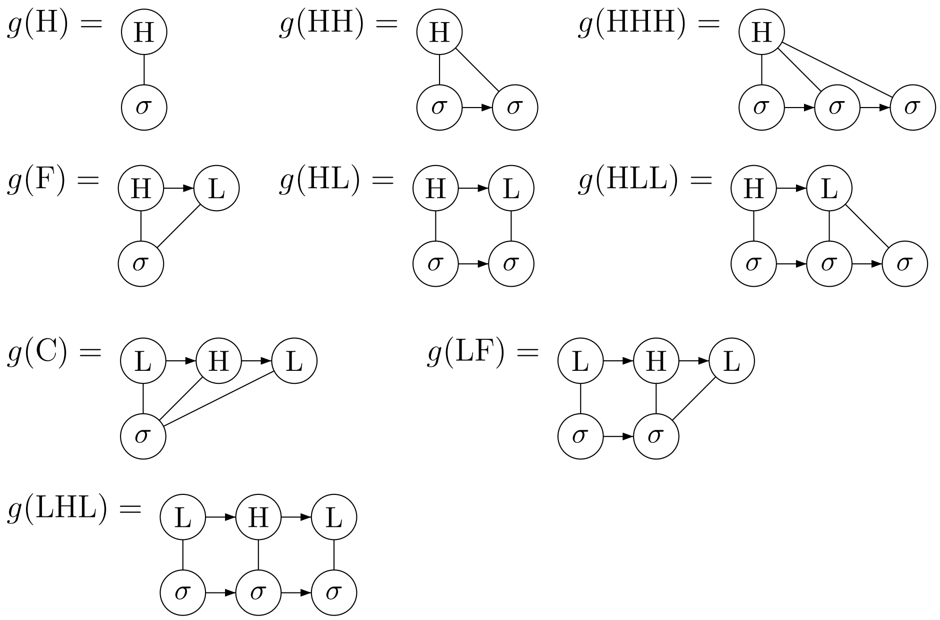
(5.2)


We can generate APGs corresponding to the Mende APRs with the alphabet of APG graph primitives $\Gamma = \{\mathrm{H}, \mathrm{L}, \mathrm{F}, \mathrm{R}, \mathrm{C}\}$ and the function g mapping Γ to graphs in $GR(\Sigma)$ for $\Sigma = \{\mathrm{H}, \mathrm{L}, \sigma\}$ as in (5.3).

(5.3)


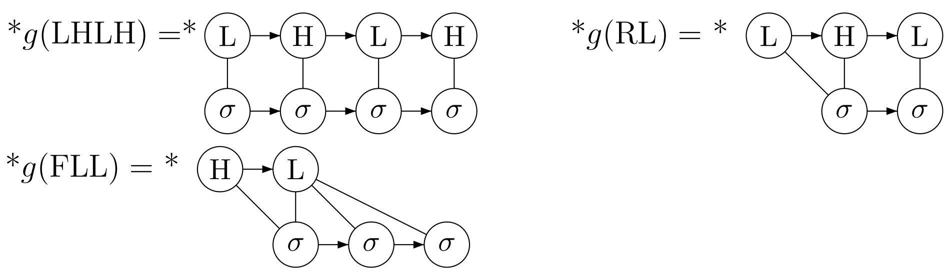
We can generate *all* of the APGs corresponding to the APRs for the string patterns in (5.1) by concatenating these primitives. In other words, the Mende pattern is a subset of $APG(\Gamma)$. Examples from the H, HL, and LHL melody rows are given below in (5.4).

(5.4) Simple Mende graphs



However, we cannot generate *only* the set of Mende APRs from concatenation. For example, $APG(\Gamma)$ also contains the following graphs. These are not valid APGs in the Mende pattern, as indicated with the traditional ungrammaticality asterisk (*).

(5.5)



In other words, the Simple Mende pattern is a *proper* subset of $APG(\Gamma)$. This is also true for the full Mende pattern and the patterns in Hausa, Kukuya, and N. Karanga Shona.

Thus, a *grammar* for each pattern must bifurcate $APG(\Gamma)$ into language-specific sets of well-defined and ill-defined APGs. The problem addressed in this chapter is this: how can we do this in a *local* way, as defined formally in Chapter 3? In Chapter 2, we saw how rule-based and optimization-based grammars achieved this through global, directional rules and constraints. This chapter shows how this can be done locally by defining each pattern of APGs exclusively through banned subgraphs. I now define these banned subgraph grammars.

5.2 Conjunctions of Negative Graph Literals

Chapter 3 defined $\mathfrak{L}^{NL}$ and $\mathfrak{L}^{NL}_{T}$, logical grammars over strings which defined patterns by specifying banned substrings. For example, the tone pattern of Kagoshima Japanese was shown in §3.3.2 (p. 89) to be describable by the following statement in $\mathfrak{L}^{NL}$:

(5.6) $\phi_{KJ} = \neg\phi_{HLL} \wedge \neg\phi_{HLH} \wedge \neg\phi_{HH} \wedge \neg\phi_{LL\ltimes}$

For example, the statement $\neg\phi_{HLL}$ bans a substring HLL, which represents a H-toned syllable farther from the right edge of the word than one syllable. This helps to capture the tone pattern of Kagoshima Japanese, in which H tones only appear on the final or penult syllable.

We may write similar statements describing graph patterns by banning *subgraphs*; this logical language will be called $\mathfrak{L}^{NL}_{G}$. However, in order to define $\mathfrak{L}^{NL}_{G}$ it is necessary to first define the concept of a subgraph, as well as the graph version of the border symbols $\rtimes$ and $\ltimes$, which were seen in Chapter 3 to play a large part in logical grammars over strings.

5.2.1 Subgraphs

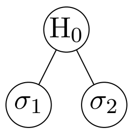
Like a substring, a subgraph is simply a piece of another graph. Formally,

Definition 20 (Subgraph) *For a graph $G = \langle V, E, A, \ell\rangle$ a* subgraph *of G is (isomorphic to) a graph $G' = \langle V', E', A', \ell'\rangle$ for which $V' \subseteq V$, $E' \subseteq E$, $A' \subseteq A$, and $\ell : V' \to \Sigma$ is a labeling function such that for all $x \in V'$, $\ell(x) = \ell'(x)$.*

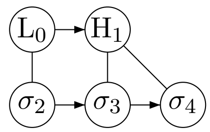
Essentially, a subgraph of a graph G is a subset of the nodes, edges, and arcs of G. Recall also that we are considering isomorphic graphs equal, so it doesn't matter if the set of nodes is technically distinct as long as the edge and arc relations and labeling function hold.

Take the following two graphs in $GR(\Sigma)$. Graph (5.7a) is a subgraph of (5.7b). The node indices are labeled here for reference.

(5.7) a.

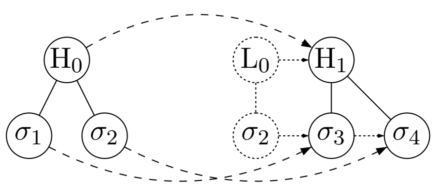


b.



Graph (5.7a) is a subgraph of (5.7b); that is, it is isomorphic to a graph that is a subset of the nodes, edges, arcs, and labeling function of (5.7b). This is depicted visually in (5.8).

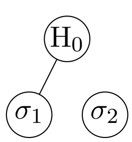
(5.8)



In (5.8), nodes 0, 1, and 2 from graph (5.7a) are mapped to nodes 1, 3, and 4 in (5.7b), respectively. Importantly, the edges $\{0, 1\}$ and $\{0, 2\}$ in (5.7a) are preserved under this mapping—their correspondents in (5.7b) are $\{1, 3\}$ and $\{1, 4\}$, respectively. Thus, (5.7a) is a subgraph of (5.7b).

Importantly, (5.7a) is what is known as a *connected* subgraph—there exists a path of edges and arcs from every node to every other node. A graph which is not connected has some node or set of nodes which are 'cut off' from all of the other nodes in the graph. An example of such a graph is given in (5.9).

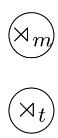
(5.9)

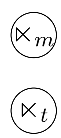


In (5.9), there is no path from node 2 to either of nodes 0 or 1. It is thus not connected. We shall only consider *connected subgraphs* for $\mathfrak{L}_G^{NL}$. This provides one restriction on the structure of the subgraphs used as literals in $\mathfrak{L}^{NL}_G$—they must be *local* in the sense that they represent a contiguous part of the larger graph. For other possible restrictions, see §5.5. Henceforth, 'subgraph' will refer only to connected subgraphs.

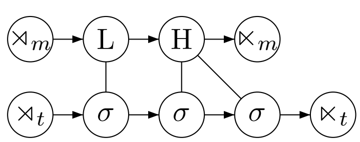
5.2.2 Boundary symbols for APGs

With subgraphs defined, there is one more concept that we need in order to create a logical language for graphs. Recall from Chapter 3 that it was important to be able to refer to beginnings and ends of strings with the boundary symbols $\rtimes$ and $\ltimes$ (e.g., in $\phi_{LL\ltimes}$ from (5.6)). We must thus define g for these symbols. We've already seen how to do this for morpheme boundaries in §4.5.2 of the previous chapter, so this is an instance of the same idea. For $\rtimes$ and $\ltimes \notin \Gamma$, we define $g(\rtimes)$ and $g(\ltimes)$ to be the following graphs:

(5.10) $g(\rtimes) =$ 

$g(\ltimes) =$ 

Where $\rtimes_t$ and $\ltimes_t$ are added to the set T_t on the partition of the labeling alphabet Σ (again, corresponding to autosegments on the timing tier) and likewise $\rtimes_m$ and $\ltimes_m$ are added to T_m. Thus, for example, using Γ and g as defined above, $g(\rtimes\text{LHH}\ltimes)$ is as follows:

(5.11) 

The nodes introduced by $g(\rtimes)$ and $g(\ltimes)$ thus indicate the beginning and the end of each tier. As the distinction between boundary symbols on their respective tiers, i.e., $\rtimes_t$ versus $\rtimes_m$, the subscripts indicating their tier will be henceforth omitted.

5.2.3 $\mathfrak{L}^{NL}_G$

With the notions of subgraphs and $g(\rtimes)$ and $g(\ltimes)$ defined, we can define a $\mathfrak{L}^{NL}_G$ as a logical language for graphs the same way as the string-based Here, I outline the basic concept of $\mathfrak{L}^{NL}_G$ in the interest of showing how it can be fruitfully applied to association patterns like those in Mende. Chapter 6 discusses the expressivity of $\mathfrak{L}^{NL}_G$ in more detail.

Recall that the logical languages from Chapter 3 were built on substring literals; satisfaction for a string of a particular statement was ultimately decided by its composite substrings. For example, recall Definition 2 of $\mathfrak{L}^P$, the full propositional logic over strings in Σ^* from which $\mathfrak{L}^{NL}$ is derived (Σ here denotes an alphabet of string symbols). A literal in $\mathfrak{L}^P$ was defined as any substring of a word in $\rtimes\Sigma^*\ltimes$:

(5.12) Def. 2a: If $\phi = u$ for a substring u of some word $w \in \rtimes\Sigma^*\ltimes$, then $\phi \in \mathfrak{L}^P$

For a string w, satisfaction for of such a literal in $\mathfrak{L}^P$ was given in Definition 3a as follows:

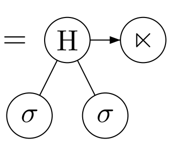
(5.13) Def. 3a: If $\phi = u$ and u is a substring of $\rtimes w \ltimes$, then $w \models \phi$

Thus, for example, ϕ_{HLL} = HLL is a string literal in $\{\text{H}, \text{L}\}$; LHLLL$\models\phi_{HLL}$ because LHLLL contains HLL as a substring; LLLH$\not\models \phi_{HLL}$ because it does not.

In the same way, we can define a *graph literal* and *satisfaction* of a graph literal. For the purposes of this chapter, I will refrain from defining a full propositional logic $\mathfrak{L}^P_G$ over graphs; more powerful logics over graphs will be discussed briefly in Chapter 8.

Definition 21 (Graph literal) *Given an alphabet of APG primitives* Γ *and* g *over graphs in* $GR(\Sigma)$*, a* graph literal *is a statement* $\phi = G$ *where* $G \in GR(\Sigma)$ *is a connected subgraph of* $g(w)$ *for some* $w \in \rtimes\Gamma^*\ltimes$

For example, using the Γ and g defined earlier in this chapter, the following is a valid graph literal because it is a subgraph of $g(\rtimes\text{LHH}\ltimes)$ (i.e., (5.11)):

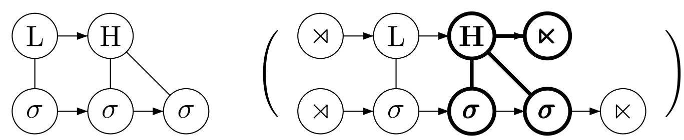
(5.14) $\phi_{FH^2} =$ 

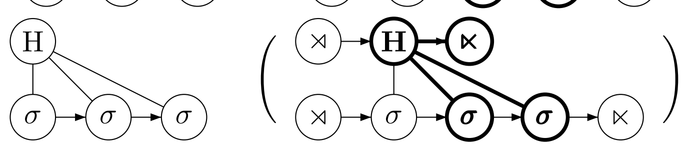
Note that graph literals are graphs in $GR(\Sigma)$, not the set $APG(\Gamma)$ of autosegmental graphs built out of concatenating primitives in Γ. In other words, graph literals need not conform to the properties of APGs. Thus, (5.14) is not a valid APG (its syllable nodes are unordered) but it is a valid graph literal.

Satisfation of a graph literal can be defined parallel to that for string literals:

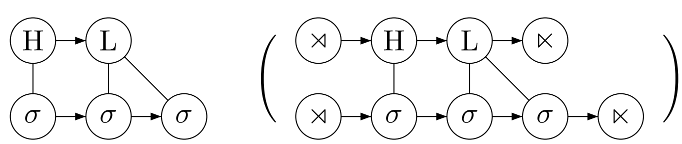
Definition 22 (Satisfaction of a graph literal) *For a graph literal $\phi = G$ for some $G \in GR(\Sigma)$, a graph $g(w) \in APG(\Gamma)$ for a string $w \in \Gamma^*$* satisfies ϕ *(written $g(w) \models \phi$) if G is a subgraph of $g(\rtimes w \ltimes)$*

For example, the following graphs satisfy ϕ_{FH^2} from (5.14) (in parentheses are $g(\rtimes w \ltimes)$ for each, with the relevant subgraph highlighted):

(5.15) $g(\text{LHH}) =$ 

$g(\text{HHH}) =$ 

In autosegmental terms, ϕ_{FH^2} represents a final H tone which is multiply associated. Note that the constraint is satisfied by a graph in which the H is associated to *more* than two σ nodes, as in $g(\text{HHH})$. Any graph which does not have a final, multiply associated H does not satisfy (5.14), such as the following graph.

(5.16) $g(\text{HLL}) =$ 

A *negative graph literal* is a statement $\neg\phi$ such that ϕ is a graph literal; just as negative literals were interpreted in Chapter 3, for a graph G, $G \models \neg\phi$ iff $G \not\models \phi$. A negative literal thus specifies a banned substructure; as with ϕ_{KJ} in (5.6), multiple banned substructures are specified through conjunction. This concept can be extended to graphs straightforwardly as follows:

Definition 23 (Conjunction of negative literals over graphs) *A* conjunction of negative graph literals *is a statement ϕ such that ϕ has the following structure:*

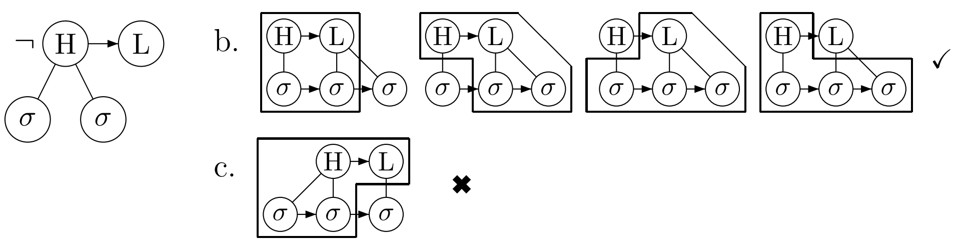
$$\phi = \neg G_0 \wedge \neg G_1 \wedge \neg G_2 \wedge \ldots \wedge \neg G_n$$

where $G_0, G_1, G_2, ..., G_n$ are graph literals meeting Definition 21, and satisfaction of $\neg(G_i)$ and $\psi_1 \wedge \psi_2$ is defined in the usual way.

Let $\mathfrak{L}_G^{NL}$ denote the set of conjunctions of negative graph literals.

For a statement ϕ let $APG(\phi)$ denote the set of graphs G such that $G \models \phi$. If ϕ is a conjunction of negative graph literals in $\mathfrak{L}_G^{NL}$, it specifies a series of subgraphs which cannot appear in any graph in $APG(\phi)$. Because these subgraphs are connected, such a statement is evaluated *locally*, just as in a conjunction of negative literals over strings. Let k be the largest number of nodes in a literal in ϕ; whether a graph G satisfies ϕ can be checked simply by looking at its subgraphs of size k. This is much like the 'scanning' for forbidden substrings in string grammars as was originally illustrated in (3.10) in Chapter 3 (p. 94). The subgraph version of this is illustrated schematically in (5.17b) and (c) below. To evaluate with respect to the forbidden subgraph in (5.17a), we scan for subgraphs of size 4.

(5.17) a. $\neg$ b. c.



This subgraph scanning can be also be done efficiently. Ferreira (2013) gives an efficient algorithm for listing, given an input graph G, its connected subgraphs of a fixed k. The connectedness of the subgraphs is crucial: the algorithm uses this to recursively bifurcate G into 'searched' nodes and 'unsearched' nodes. An APG-specific algorithm may improve on Ferreira (2013)'s result, however this will be left for future work—the crucial point here is that it is the *local*, i.e., connected nature of the constraints that makes them efficiently computable (more on this is discussed in Chapter 8).

Thus, $\mathfrak{L}_G^{NL}$ provides us with a way of specifying subsets of $APG(\Gamma)$ through banned substructure constraints. The following two chapters test to what extent statements in $\mathfrak{L}_G^{NL}$ can be applied to the typology of language-specific association patterns in tone. As with the string logics discussed in Chapter 3, $\mathfrak{L}_G^{NL}$ forms a strong hypothesis about tonal well-formedness of association in tone:

(5.18) **The $\mathfrak{L}_G^{NL}$ Hypothesis:** Surface well-formedness constraints in tonal phonology are local over autosegmental structures.

Just as the $\mathfrak{L}^{NL}$ Hypothesis offered in Chapter 3 (p. 95), this is a strong hypothesis because it restricts the range of grammars available to speakers to conjunctions of negative graph literals. As with banned substructure constraints in strings, as a theory of well-formedness it comes with straightforward cognitive interpretations of evaluation and learning (to be discussed more in Chapter 8).

This chapter and the next review the tone patterns discussed in Chapter 2 and show them to be describable by constraints in $\mathfrak{L}_G^{NL}$, thus confirming the hypothesis (although, as discussed in Chapter 6 for Wan Japanese and in Chapter 8 for what I call the 'superstructure problem', for some patterns this requires certain representational assumptions). The remainder of this chapter is concerned with tone-mapping patterns, first showing how they can be described by constraints in $\mathfrak{L}_G^{NL}$, then using these analyses to illustrate the superiority of $\mathfrak{L}_G^{NL}$ as a theory of tonal well-formedness when compared to derivational theories or OT.

5.3 Tone Mapping Patterns with Local Graph Constraints

This section shows how the directional, quality-specific, and positional association well-formedness generalizations of tone mapping patterns in Mende, Hausa, Kukuya, and N. Karanga discussed in Chapter 2 can all be described by banned substructure constraints over APGs. This means that each of these types of generalizations is *local*, in the computational sense defined in this chapter and in Chapter 3.

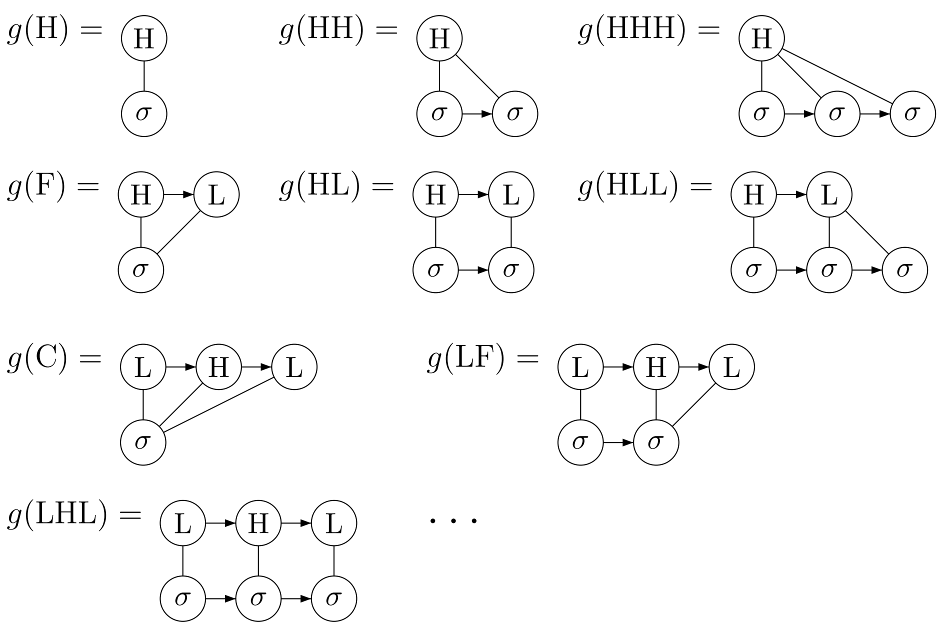
5.3.1 Simple Mende

I begin with Simple Mende, first mentioned in §5.1, and repeated below here. Again, in autosegmental terms this is a *directional* generalization: multiple association is only allowed on the right edge of the word.

(5.19) Simple Mende string patterns

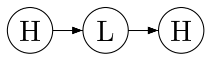
(H melody)	H,	HH,	HHH,	HHHH,	...
(L melody)	L,	LL,	LLL,	LLLL,	...
(HL melody)	F,	HL,	HLL,	HLLL,	...
(LH melody)	R,	LH,	LHH,	LHHH,	...
(LHL melody)	C,	LF	LHL,	LHLL,	...

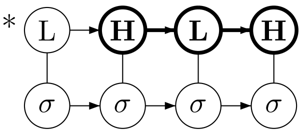
(5.20) Simple Mende graphs



Note again that this does not take into account the less-attested cases in Mende—for example, actual Mende allows both LHH and LLH (see (2.23) on p. 31 in Chapter 2, §2.2.1.3). However, it represents a pure version of this directional generalization, and is thus instructive to illustrate a analysis of it with $\mathfrak{L}_G^{NL}$. The other generalizations in Mende will be discussed momentarily in §5.3.4.

This set can be described by a small set of banned subgraphs. First, there are no HLH melodies. This structure is given below as ϕ_{HLH} in (5.21a). An example graph which we would like to exclude that contains ϕ_{HLH} is given in (5.21b).

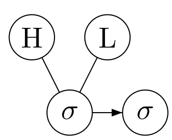
(5.21) a. ϕ_{HLH} = 

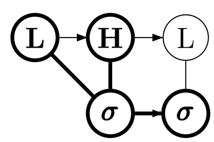
b. * 

*g(LHLH)

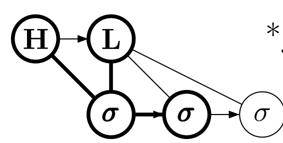
The graph in (5.21b) illustrates how banning an HLH sequence on the melody tier not only removes graphs with a single HLH sequence on the tonal tier, but also LHLH, LHLHLH, etc.; that is, any sequence of more than three autosegments. This is because any such sequence must (because the OCP is in effect) include a HLH.

Next, we also want to ban nonfinal contours. This can be done by by singling out the structure $\phi_{NF\text{-}Cont}$ in (5.22a). Note that for a contour to be nonfinal, its associated syllable must be followed by some other syllable. It is exactly this structure that $\phi_{NF\text{-}Cont}$ specifies. Note the lack of a directed edge between the H and L nodes in (5.22a); this means that both falling and rising contours are matched by this graph, as the reader can confirm in the examples in (5.22b) and (c).

(5.22) a. $\phi_{NF\text{-}Cont}$ = 

b. * 

*g(RL)

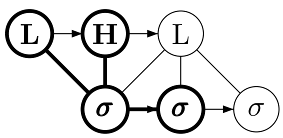
c. * 

*g(FLL)

Note that $\phi_{NF\text{-}Cont}$ also matches nonfinal LHL contours as well, as such a structure is a superstructure of (5.22a).[1] An example is given below. The graph in (5.23) below actually contains two instances of $\phi_{NF\text{-}Cont}$, as both an L following an H and an

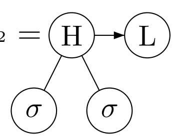
[1] Two similar constraints, *T_1T_2-σ_{nonfinal} and *$T_1T_2T_3$-σ_{nonfinal}, are proposed by Zhang (2000) to assign violations to nonfinal two- and three-tone contours, respectively. As Zhang does not formally define how associations are violated, it is not clear if candidates which violate *T_1T_2-σ_{nonfinal} also violate *$T_1T_2T_3$-σ_{nonfinal} as well.

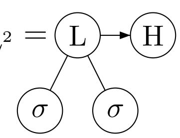
H following an L are all assigned to a nonfinal syllable, but just the first instance is highlighted.

(5.23) * 

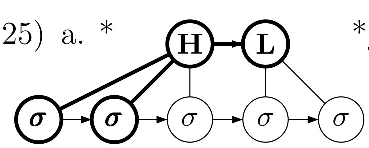
$*g$(CLL)

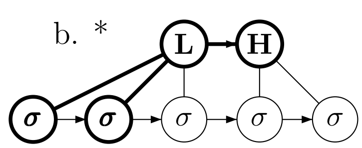
The third and final piece to the Simple Mende tone association pattern is left-to-right directionality. This, too, can be characterized by banned substructures. These structures are given below as the literals $\phi_{NF\text{-}H^2}$ and $\phi_{NF\text{-}L^2}$, which represent the situation in which a nonfinal (i.e., followed by another tone) H tone and a nonfinal L tone, respectively, are associated to more than one syllable. In naming these literals, subscripts represent the number of associations, thus H^2 refers to a H tone associated to two syllables. Normal-sized numbers shall later on be used to indicate the position of an association (e.g., $H2$ will indicate a H tone associated to the second syllable).

(5.24) a. $\phi_{NF\text{-}H^2} =$ 

b. $\phi_{NF\text{-}L^2} =$ 

These structures are, essentially, $\phi_{NF\text{-}Cont}$ flipped upside down. Both $\phi_{NF\text{-}Cont}$ and the structures in (5.24) prohibit a nonfinal autosegment to associating to more than one autosegment. In the case of (5.24), to get blanket left-to-right directionality we have to specify constraints for both H and L, as we need to ban both graphs below in (5.25) below.

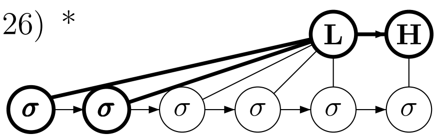
(5.25) a. * 

$*g$(HHHLL)
b. * 

$*g$(LLLHH)

Later on we shall see with Kukuya that languages may pick one or the other—this corresponds to Zoll (2003)'s notion of spreading that is 'dependent on tone quality'. Note that even though the structures in (5.24) only refer to two timing tier units, they ban graphs like in (5.25) with plateaus on the left edge no matter the length of their plateau. An explicit example is given in (5.26) below.

(5.26) *



$*g(\text{LLLLLH})$

Again, (5.26) contains multiple instances of $\phi_{\textit{NF-L}^2}$, but only the violation including the first two σ nodes is highlighted.

These last three subgraphs, $\phi_{\textit{NF-Cont}}$, $\phi_{\textit{NF-H}^2}$, $\phi_{\textit{NF-L}^2}$, are thus enough to capture the directional association pattern of Simple Mende. The conjunction of negative literals below in (5.27) bans each of these subgraphs.

(5.27) $\neg\phi_{HLH} \wedge \neg\phi_{\textit{NF-Cont}} \wedge \neg\phi_{\textit{NF-H}^2} \wedge \neg\phi_{\textit{NF-L}^2}$

The reader can confirm, as the examples above partly show, that the set of Simple Mende graphs is exactly the subset of $APG(\Gamma)$ which satisfies (5.27). Thus, the generalization that multiple association can only occur on the right edge is describable in $\mathfrak{L}^{NL}_{G}$.

5.3.2 Hausa

Recall from Chapter 2, §2.2.1.1 (p. 28) that Hausa exhibited the opposite directional generalization to Simple Mende: multiple association only occurs on the left. The following shows that such a pattern can be obtained with a conjunction of negative literals very similar to (5.27).

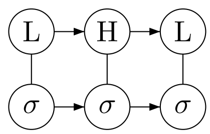
Recall that Hausa was the mirror image of Mende, with a few exceptions: HLH melodies are allowed, contours do not appear on monosyllables, and rising tones are not permitted.

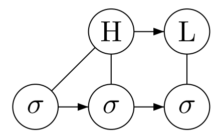
Examples of graphs which are valid in Hausa but not in Simple Mende are given in (5.29).

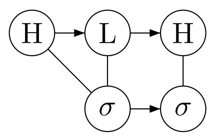
(5.28) Hausa string patterns

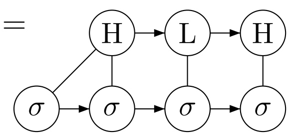
(H melody)	H,	HH,	HHH,	...
(L melody)	L,	LL,	LLL,	...
(HL melody)	HL,	HLL,	HHHL,	...
(LH melody)	LH,	LHH,	LLLH,	...
(LHL melody)	RL	LHL,	LLHL,	...
(HLH melody)	FH	HLH,	HHLH,	...

(5.29) Hausa graphs

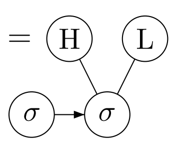
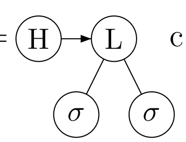
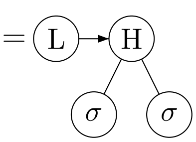
g(LHL) = 

g(HHL) = 

g(FL) = 

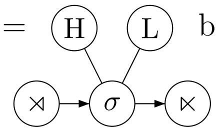
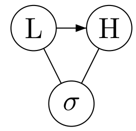
g(HHLH) = 

First, to get the left-edge directionality, we need constraints that work in the opposite direction of those in (5.22) and (5.24) for Simple Mende. The relevant structures are given in (5.30) below. $\phi_{NI\text{-}Cont}$ in (5.30a) specifies noninitial contours, parallel to $\phi_{NF\text{-}Cont}$ for Simple Mende, and (5.30b) $\phi_{NI\text{-}L^2}$ and (5.30c) $\phi_{NI\text{-}H^2}$ specify spreading of a noninitial tone autosegment, parallel to $\phi_{NF\text{-}L^2}$ and $\phi_{NF\text{-}H^2}$ from (5.24) do for nonfinal tones.

(5.30) a. $\phi_{NI\text{-}Cont}$ =  b. $\phi_{NI\text{-}L^2}$ =  c. $\phi_{NI\text{-}H^2}$ = 

As with the three directionality constraints in Simple Mende, banning these three subgraphs will be enough to capture the generalization that multiple association can only occur on the left edge.

The other generalizations in Hausa can be similarly dealt with. First, the absence of contours on monosyllables, and the absence of rising contours in general, can be captured by banning the subgraphs in (5.31a) (5.31b), respectively.

(5.31) a. $\phi_{\mathit{1\sigma\text{-}Cont}} =$  b. $\phi_R =$ 

There are actually two ways of capturing the absence of rising tones: either we ban ϕ_R in (5.31b), or we consider the set $APG(\Gamma)$ where Γ contains no 'R' symbol. This point will be returned to in §5.5.

Finally, as Hausa allows HLH melodies but no melody with four tones or more, we need to specify the following structures in ϕ_{HLHL} and ϕ_{LHLH}. These two structures will be contained by any melody tier of four autosegments or more.

(5.32) a. $\phi_{HLHL} = \text{H} \rightarrow \text{L} \rightarrow \text{H} \rightarrow \text{L}$

b. $\phi_{LHLH} = \text{L} \rightarrow \text{H} \rightarrow \text{L} \rightarrow \text{H}$

Hausa is thus the following statement banning all of the above structures:

(5.33) $\neg\phi_{\mathit{NI\text{-}Cont}} \wedge \neg\phi_{\mathit{NI\text{-}L^2}} \wedge \neg\phi_{\mathit{NI\text{-}H^2}} \wedge \neg\phi_{\mathit{1\sigma\text{-}Cont}} \wedge \neg\phi_R \wedge \neg\phi_{HLHL} \wedge \neg\phi_{LHLH}$

Thus, Hausa shows how both directional generalizations and generalizations regarding melodies and contours can be captured in $\mathfrak{L}_G^{NL}$.

5.3.3 Kukuya

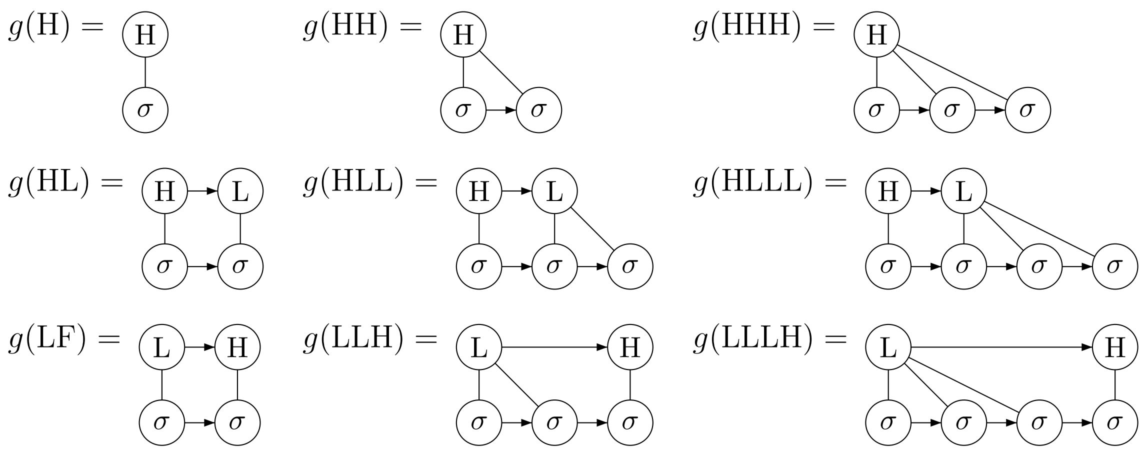
Quality-specific association generalizations are local in the same way. Recall that we saw one such generalization in Kukuya, in which multiple association of H was disallowed in the presence of an L tone. To review the data, Kukuya's tone pattern is almost identical to Simple Mende, yet instead of *LHH surface patterns, LLH is attested, as highlighted below.

(5.34) Kukuya string patterns

(H melody)	H,	HH,	HHH,	HHHH,	...
(L melody)	L,	LL,	LLL,	LLLL,	...
(HL melody)	F,	HL,	HLL,	HLLL,	...
(LH melody)	R,	LH,	**LLH**,	**LLLH**,	...
(LHL melody)	B,	LF	LHL,	LHLL,	...

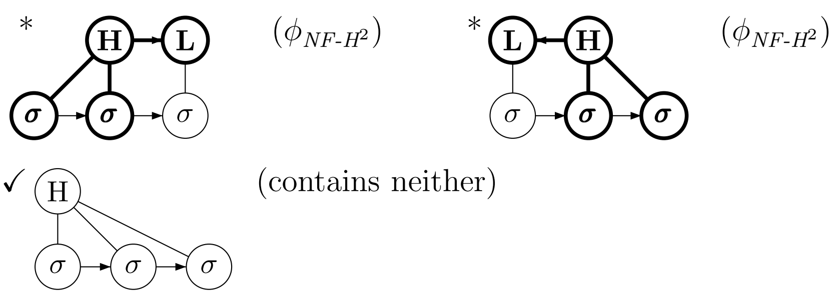
The crucial contrast to capture is that all graphs in Kukuya represent right-edge directional tone associations, with the exception of the LH melodies. This contrast is highlighted by the graphs below in (5.35).

(5.35) Kukuya graphs



When we look at the Kukuya surface strings in (5.34) and graphs in (5.35), we see that, unlike L, a H is not allowed to spread when it is either nonfinal or noninitial. However, we don't want to ban all multiple association of H, as g(HHH) shows it is possible if it is the only tone. Thus, starting with the constraints for Simple Mende in (5.27), we simply add $\phi_{NI\text{-}H^2}$, to also ban noninitial Hs from spreading, and remove $\phi_{NF\text{-}L^2}$, to allow a nonfinal L to spread (as in the last graph in (5.35)). That this bans any spreading of an H in the presence of L, but allows spreading in case H is the only tone on the melody tier, is illustrated below in (5.36).

(5.36)

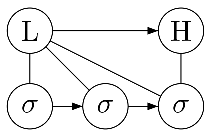


This collection of constraints is given below in (5.37). We will need to add to this, but the reader can confirm that all of the graphs in (5.35) and the graphs corresponding to the other strings in (5.34) conform to the following constraints.

(5.37) Preliminary Kukuya constraints:

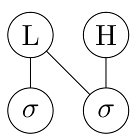
$$\neg\phi_{HLH} \wedge \neg\phi_{NF\text{-}Cont} \wedge \neg\phi_{NF\text{-}H^2} \wedge \neg\phi_{NI\text{-}H^2}$$

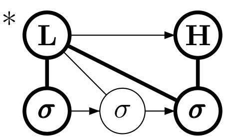
Crucially, these constraints capture that H and L behave independently with respect to spreading: H cannot spread if it is noninitial or nonfinal, whereas L can spread in both of these situations. Note that in the above constraints, we have removed all restrictions from L. This creates an interesting situation, not seen in the previous graph sets, in which a nonfinal L can spread on to the syllable of a final H to create a final contour, as in the following graph:

(5.38) * 

*g(LLR)

Such a tone sequence is not attested in Kukuya, so we need a further constraint to ban it. Such a constraint would ban the structure in the literal $\phi_{L^2\text{-}Cont}$, given below in (5.39a), which specifies a multiply-associated L participating in a contour.

(5.39) a. $\phi_{L^2\text{-}Cont} =$ 

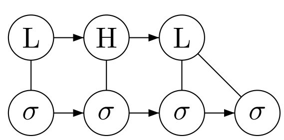
b.* 

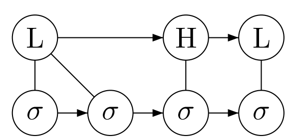
The final set of Kukuya constraints thus includes $\phi_{L^2\text{-}Cont}$:

(5.40) Final Kukuya constraints:

$$\neg\phi_{HLH} \wedge \neg\phi_{NF\text{-}Cont} \wedge \neg\phi_{NF\text{-}H^2} \wedge \neg\phi_{NI\text{-}H^2} \wedge \neg\phi_{L^2\text{-}Cont}$$

Note that this analysis allows both of these graphs for an LHL melody:

(5.41) a. g(LHLL) = 

b. g(LLHL) = 

The data available for Kukuya is ambiguous as to which of these is valid, as only trisyllableic roots are attested (Hyman, 2011a; Paulian, 1974). Note also that Zoll (2003)'s analysis allows for both of these APRs as well. As such, I will not add any further constraints which distinguish between these two.

5.3.4 Mende (continued)

Now that we've seen how the constraints governing the association of H and L can be chosen independently by banned substructure constraints, the full Mende pattern is easily describable in $\mathfrak{L}_G^{NL}$. Recall from §2.2.1.3 that, in additio to the surface patterns seen for Simple Mende in (5.1), HF, HHL, and LLH are also attested surface strings in Mende. Thus, the full set of Mende string patterns is as in (5.42) below, with the additional string patterns highlighted in bold (this chart is repeated from (2.24), on p. 32 of Chapter 2).

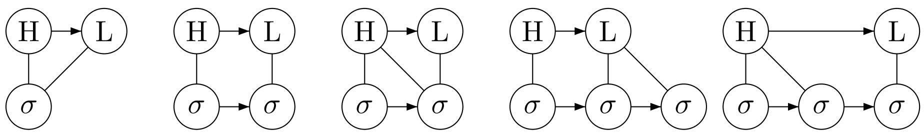
(5.42) Mende string patterns

(H melody)	H,	HH,	HHH,	HHHH,	...
(L melody)	L,	LL,	LLL,	LLLL,	...
(HL melody)	F,	HL,	HLL,	HLLL,	...
		HF,	**HHL**,	**HHLL**,	**...**
(LH melody)	R,	LH,	LHH,	LHHH,	...
			LLH,	**LLHH**,	**...**
(LHL melody)	B,	LF	LHL,	LHLL,	...

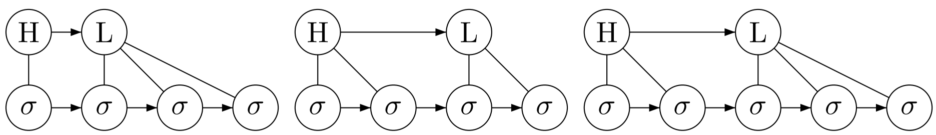
Recall that the generalization for this additional data is one of positional association: as long as it does not create a rising tone contour, an initial, nonfinal tone can associate to either the first syllable only or both the first and second syllables. In particular, this means that for 4σ forms for the HL and LH melodies are HLLL, HHLL and LHHH, LLHH, respectively, and not *HHHL and *LLLH. Thus, for example, for HL melodies, for a word n σs long, both HL^{n-1} and HHL^{n-2} are valid strings (as are HL^{n-1} and HHL^{n-2} for LH melody words). The set of valid HL and LH graphs are thus as follows:

(5.43) a. Mende HL melody graphs

g(F) g(HL) g(HF) g(HLL) g(HHL)

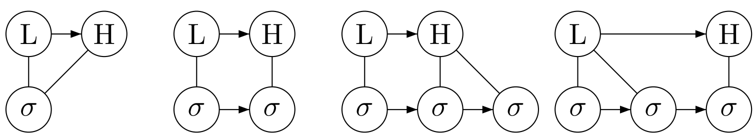


g(HLLL) g(HHLL) g(HHLLL) ...

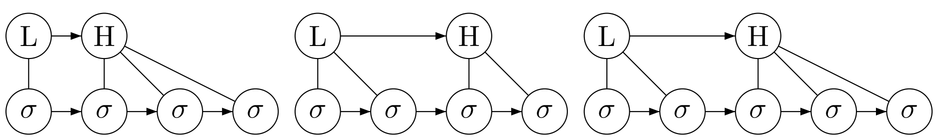


b. Mende LH melody graphs

g(R) g(LH) g(LHH) g(LLH)

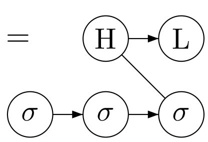


g(LHHH) g(LLHH) g(LLHHH) ...

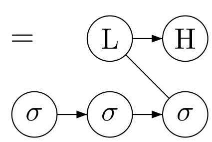


Mende can then largely be described by modifying $\phi_{NF\text{-}H^2}$ and $\phi_{NF\text{-}L^2}$ from (5.24) to the following literals, which specify a nonfinal tone associated to the third—or later—syllable in a word.

(5.44) a.$\phi_{NF\text{-}H3}$ =



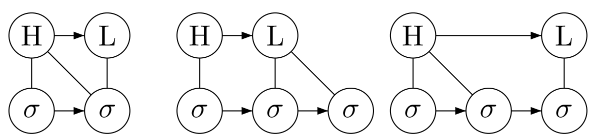
b.$\phi_{NF\text{-}L3}$ =



Banning $\phi_{NF\text{-}H3}$ and $\phi_{NF\text{-}L3}$ from (7.74), instead of $\phi_{NF\text{-}H^2}$ or $\phi_{NF\text{-}L^2}$, allows graphs in which a nonfinal tone is associated also to the second syllable. Such graphs were banned in Simple Mende, but need to be allowed in the full Mende pattern. For example, a HF sequence is valid in Mende, but a *HHF sequence is not. As highlighted in (5.45) below, this is due to the fact that *g(HHF) contains $\phi_{NF\text{-}H3}$, while g(HF) does not. Similarly, surface patterns like *HHHHL are banned because the initial H tone has, as can be seen in *g(HHHHL) in (5.45b) below, associated beyond the second syllable. This goes not only for HL melody words, but LHL melody words as well, as can be seen for the invalid graph *g(LLHL) in (5.45b).

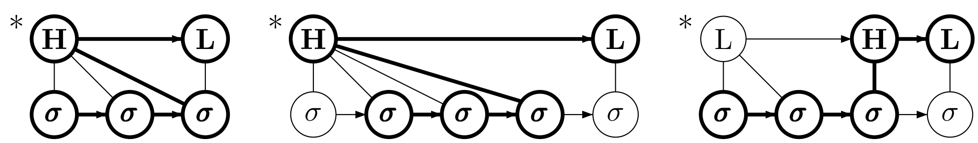
(5.45) a. HL melody graphs in Mende not containing $\phi_{NF\text{-}H3}$

g(HF) g(HLL) g(HHL)



b. Graphs not in Mende which contain $\phi_{NF\text{-}H3}$

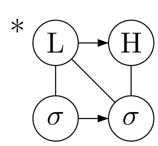
*g(HHF) *g(HHHHL) *g(LLHL)



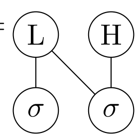
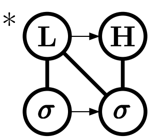
Thus, using $\phi_{NF\text{-}H3}$ and $\phi_{NF\text{-}L3}$ instead of $\phi_{NF\text{-}H^2}$ and $\phi_{NF\text{-}L^2}$ allow graphs that were not allowed in Simple Mende, but still correctly ban forms in which a nonfinal tone has associated to the third syllable or later. Note, interestingly, that specifying $\phi_{NF\text{-}H3}$ as a banned structure allows for both HF and HHL surface patterns, which were both banned in Simple Mende. This is due to a structural similarity between these two APRs, highlighted by the current analysis, which is that both include a nonfinal H associating to the second syllable.

One more change is needed in order to fully capture Mende tone association. Note from (5.42) that while the final-contoured bisyllabic HL melody form HF is attested, LR is not a valid realization of a bisyllabic LH melody.

(5.46) g(HF) *g(LR)



The generalization is simple: falling contours are allowed on words longer than one syllable long, but rising contours are not. (Note that this also holds true for bisyllabic LHL melody words, which are pronounced LF.) As was also seen in Kukuya, H and L are thus patterning independently of each other. Again, this is no problem for banned substructure grammars in $\mathfrak{L}^{NL}_{G}$. In fact, to deal with this restriction we can simply take the $\phi_{L^2\text{-}Cont}$ literal from the Kukuya analysis above, repeated below in (5.47a). The disyllable with a falling contour in (5.46b) above contains $\phi_{L^2\text{-}Cont}$, as shown below in (5.47b).

(5.47) a. $\phi_{L^2\text{-}Cont} =$  b. * 

The statement which captures the full set of graphs for Mende is thus a slight modification of the one in (5.27) for Simple Mende. It is given below in (5.48).

(5.48) Full Mende constraints:

$$\neg\phi_{HLH} \wedge \neg\phi_{NF\text{-}Cont} \wedge \neg\phi_{NF\text{-}H3} \wedge \neg\phi_{NF\text{-}L3} \wedge \neg\phi_{L^2\text{-}Cont}$$

For the additional patterns in Mende, we ban nonfinal tones from associating to the third or later syllable (with $\phi_{NF\text{-}H3}$ and $\phi_{NF\text{-}H3}$), and we ban rising contours in forms with two or more syllables (with $\phi_{2\sigma R}$). The other banned substructures from (5.27), ϕ_{HLH} (specifying HLH melodies) and $\phi_{NF\text{-}Cont}$ (specifying nonfinal contours), remain unchanged.

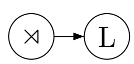
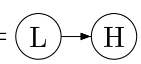
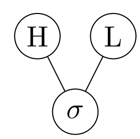
5.3.5 N. Karanga

Finally, I turn to the positional association generalization of N. Karanga. Recall from Chapter 2 §2.2.1.4 that H-toned Karanga verbs have different tone patterns in the Assertive and Non-assertive tenses. These patterns, originally listed in (2.27) and (2.28) in §2.2.1.4 (p. 34), are repeated below in (5.49). What is shared by both tenses is the main 'positional' generalization that the H tone spreads up until the third syllable.

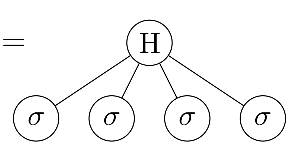
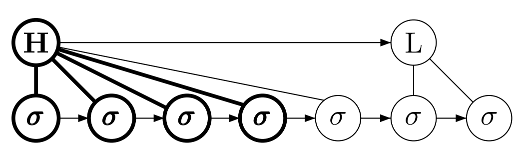
(5.49) N. Karanga verbs

ASSERTIVE	NON-ASSERTIVE
H	H
HH	HL
HHH	HLH
HHHL	HHLH
HHHLL	HHHLH
HHHLLL	HHHLLH
HHHLLLL	HHHLLLH

However, the two tenses have their differences, and so the following deals with the graph sets for each tense separately. First, the ASSERTIVE. For the melody tier, the generalization that needs to be captured is that only H or HL patterns are allowed. In terms of negative constraints, initial L tones and LH sequences are banned. These structures are singled out by the statements $\phi_{I\text{-}L}$ and ϕ_{LH}, respectively, below in (5.50a) and (b). Also, contours do not appear at all in N. Karanga. The graph in (5.50c) ϕ_{Cont} matches any contour.

(5.50) a. $\phi_{I\text{-}L} =$  b. $\phi_{LH} =$  c. $\phi_{Cont} =$ 

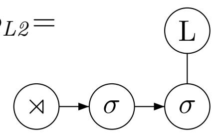
With the melody established, all that remains for the ASSERTIVE is to capture that the initial H associates to exactly three syllables if there are three or more syllables present, or all of the syllables if there are fewer than three. In terms of negative constraints, this can be broken down into a few sub-generalizations. First, a H cannot spread to more than three syllables. Such a structure is singled out by the quadrupally-associated H structure in ϕ_{H^4} below.

(5.51) a.$\phi_{H^4} =$  b. * 

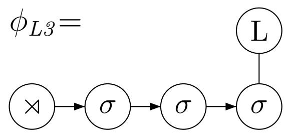
*g(HHHHHLL)

While it is true that the initial H does not associate to more than three syllables, it is also true that the initial H does not associate to *less* than three syllables, unless there are only one or two syllables in the word. Another way of stating this is that the L tone never associates to the second or third syllable (that it also can't associate to the first syllable is subsumed by the generalization that L can't be initial). These situations are described by the graphs in ϕ_{L2} and ϕ_{L3}. Examples of unattested associations containing these graphs are given in (5.53).

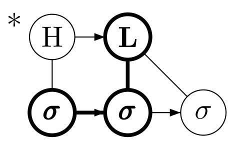
(5.52) a. ϕ_{L2}=



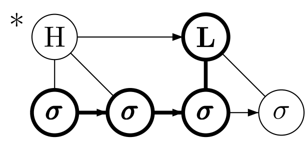
b. ϕ_{L3}=



(5.53) a. *g(HLL)



b. *g(HHLL)

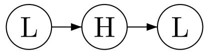


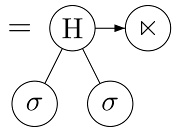
If a L is not allowed to associate to the first three syllables of the word, then an initial H must fill in these associations. However, by banning quadruple H associations, it will never associate past the third syllable. Thus, by banning graphs such as in (5.51a) and (5.53), we can get the set of graphs in which an initial H associates to exactly three syllables, if there are that many in the word. Thus, with the following conjunction of negative literals banning each subgraph so far mentioned, we can describe the ASSERTIVE pattern for H-toned verbs in N. Karanga.

(5.54) N. Karanga ASSERTIVE:

$$\neg\phi_{I\text{-}L} \wedge \neg\phi_{LH} \wedge \neg\phi_{Cont} \wedge \neg\phi_{H^4} \wedge \neg\phi_{L2} \wedge \neg\phi_{L3}$$

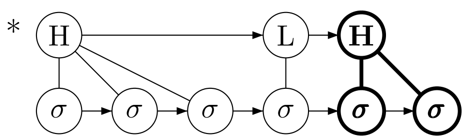
The NON-ASSOCIATIVE pattern is slightly more complex. First, there are three possible tone melodies, H, HL, and HLH, depending on the number of syllables in the word. Thus, we cannot, as in the ASSERTIVE, ban ϕ_{LH}. However, the generalization can be restated in negative terms as follows: there are no initial L tones, and no LHL sequences on the melody tier. The first of these structures was already discussed in $\phi_{I\text{-}L}$ from (5.50). The second can be specified by the literal ϕ_{LHL} below in (5.55a).

(5.55) a. $\phi_{LHL} =$ 

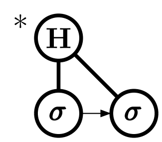
b. $\phi_{F\text{-}H^2} =$ 

Note that in (5.49), the final H is never associated to more than one syllable. The second structure specified in (5.55a), $\phi_{F\text{-}H^2}$, matches a multiply-associated final H. Banning this structure ensures not only that a final H only associates to one syllable, but also that bisyllabic or longer forms must include an L. Examples of this are given in (5.56) below.

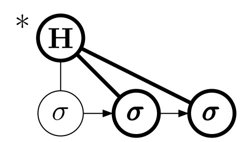
(5.56) a. $*g$(HHHLHH)



b. $*g$(HH)



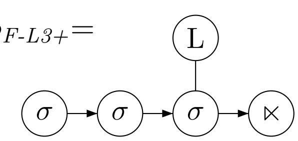
c. $*g$(HHH)



It is also true, as in the ASSOCIATIVE pattern, that H is not allowed to associate to more than four syllables (ϕ_{H^4}). As it turns out, these are the only restrictions on H associations that we need to consider.

For associations with L, there are two important generalizations. First, an L does not associate to the final syllable in words of three syllables or more. (It also technically does not associate to the final syllable in monosyllabic words, but this can be subsumed under the generalization that there is no initial L.) A structure in which this does occur is specified by $\phi_{F\text{-}L3+}$ below.

(5.57) $\phi_{F\text{-}L3+}=$

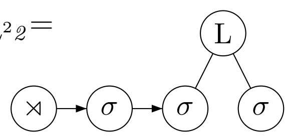


Banning $\phi_{F\text{-}L3+}$ ensures that an H must associate to the final syllable in trisyllabic or longer words.

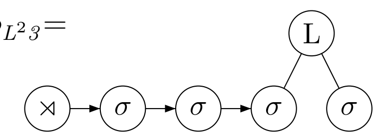
The second important generalization with regards to L is related to ϕ_{L2} and ϕ_{L3} from the ASSERTIVE pattern. Recall that in the ASSERTIVE pattern, L cannot associate to the second or third syllable, and that this motivated the ternary spreading of the H. In the NON-ASSERTIVE pattern, the L *can* associate to the second and third syllable, as in HL, HLH, and HHLH. Thus, the ternary spreading of the H must be motivated by some other constraint on L.

What is crucial is that the L does not ever associate to multiple syllables when it is associated to the second or third syllable. These two situations are captured by the graphs in (5.58) below.

(5.58) a. $\phi_{L^2 2}$=

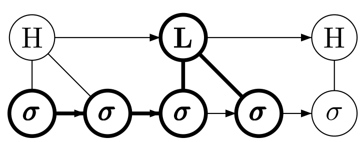


b. $\phi_{L^2 3}$=



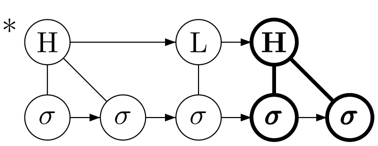
These structures, in conjunction with other structures discussed above, motivate tertiary spreading in the NON-ASSERTIVE pattern. Consider a form with five syllables in which the initial H only associates to the first two syllables. In such a case, there are two options for the associations of the remaining L and H tones. One is that the following L associates to the third and fourth syllables. This contains $\phi_{L^2 3}$, as highlighted below.

(5.59) *g(HHLLH) = *



The only possibility, then, which does not contain $\phi_{L2^2 3}$, is to keep L associated only to the third syllable, and have the second H associated to the fourth and fifth. However, as highlighted below in (5.60), this contains $\phi_{F\text{-}H^2}$, discussed above.

(5.60) *g(HHLHH) = *



Thus, graphs with more than four syllables in which an L is associated earlier than the fourth syllable in the word will necessarily contain one of the substructures specified above. If we ban all of these substructures, then we are ensured tertiary spreading of the H exactly in the situation in which there are five or more syllables. Of course, we do not want the H spreading to more than three syllables, so for the NON-ASSERTIVE tense as well we must ban the ϕ_{H^4} substructure representing association of an H to four or more syllables.

This concludes discussion of all of the literals needed to describe the pattern for H-toned NON-ASSERTIVE verbs. By banning the substructures they describe, we can get exactly the set of graphs corresponding to the set of strings for the NON-ASSOCIATIVE in (5.49). This conjunction of literals is as follows:

(5.61) $\neg\phi_{I\text{-}L} \wedge \phi_{LHL} \wedge \neg\phi_{F\text{-}H^2} \wedge \neg\phi_{F\text{-}L3+} \wedge \neg\phi_{L^2 2} \wedge \neg\phi_{L^2 3}$

Again, while the subgraphs used were slightly larger than those for other generalizations, the positional generalization in N. Karanga Shona is captured by banning the subgraph ϕ_{H^4} and the tense-specific constraints on where L can associate—ϕ_{L2} and ϕ_{L3} in the ASSERTIVE tense and $\phi_{L^2 2}$ and $\phi_{L^3 3}$ in the NON-ASSERTIVE.

5.3.6 Interim conclusion: $\mathfrak{L}^{NL}_G$ and language-specific association

This section begun to test the $\mathfrak{L}^{NL}_G$ Hypothesis by applying conjunctions of negative subgraph literals to pick out language-specific association generalizations resulting from the tone-mapping patterns discussed in Chapter 2. The result is consistent with the hypothesis—these patterns were found to be describable by logical constraints written in $\mathfrak{L}^{NL}_G$. The largest k value (again, for a subgraph the number of nodes) for any constraint was 6, required for (5.58) in N. Karanga. Most of the other constraints had k values of 4 or 5. This is significant, as this means these patterns are fundamentally *local*, in the sense defined in this chapter and Chapter 3, as well-formedness is entirely determined by the presence or absence of banned substructures. This is a generalization missed by previous analyses, as the following discusses in more detail.

5.4 Discussion

The previous section has shown how $\mathfrak{L}_G^{NL}$ presents a unified theory of directional, quality-specific, and positional association generalizations in tone-mapping patterns, as well as other generalizations, such as constraints on possible melodies. This section compares this theory with the previous analyses for these patterns given in Chapter 2, as well as one potential way of restricting the typological predictions of $\mathfrak{L}_G^{NL}$ even further.

5.4.1 Comparison with previous analyses

The analyses in the previous section of tone mapping patterns with constraints in $\mathfrak{L}_G^{NL}$ has shown that the language-specific well-formedness conditions over autosegmental structures stemming from directional, quality-specific, and positional association patterns are fundamentally local. This is a generalization that has been missed in previous analyses. For example, consider the directional association patterns of Mende and Hausa. The analyses discussed in Chapter 2 capture the difference between Mende and Hausa using a *global* notion of directionality—implemented in rule-based theories as parameter on the direction of association (as in §2.3.1.1) and in optimization-based theories as the ranking of directional Align constraints (as in §2.3.1.2).

However, a $\mathfrak{L}_G^{NL}$ theory of autosegmental well-formedness *does* preserve OT's focus on surface well-formedness. A clear example is found in the conditions on H spreading in Kukuya. In a rule-based framework, Hyman (1987)'s analysis of Kukuya depended on a language-specific rule to 'fix' a H that had, through left-to-right association, spread to multiple TBUs following an L tone:

(5.62) Kukuya L-Spreading (repeated from Chapter 2, (2.55), p. 51)

L H $\rightarrow$ L H $\rightarrow$ LLH

σ σ σ $\quad$ σ σ σ

Zoll (2003) rightly criticized this analysis for missing the surface generalization that H could not spread in the presence of an L tone. However, this insight is directly captured in the $\mathfrak{L}^{NL}_G$ analysis of Kukuya by the banning of the constraints $\phi_{NF\text{-}H^2}$ and $\phi_{NI\text{-}H^2}$. In this way, $\mathfrak{L}^{NL}_G$ preserves Zoll's insight that H and L can behave independently of some general directional association paradigm.

Additionally, adopting $\mathfrak{L}^{NL}_G$ as a theory of autosegmental well-formedness makes specific typological predictions: a well-formedness pattern should not be attested if it cannot be described in terms of the presence or absence of banned substructures. This is superior to derivation-based theories, which while originally had some predictive power due to simple parameters, such as right-to-left versus left-to-right association, have since lost this typological clarity through the proliferation of analysis-specific association paradigms, such as Leben (1978)'s complex well-formedness condition for Mende (see Chapter 2, §2.3.1.1, p. 52) or the various definitions of 'edge-in' association (see Chapter 2, §2.3.1.1, p. 2.62). The same goes for the OT analysis of N. Karanga, which required an analysis-specific constraint to capture the positional generalization that the initial H must spread, if possible, to the third syllable. By focusing instead on the local nature of these various well-formedness conditions, $\mathfrak{L}^{NL}_G$ gives us a unified theory of the typology.

Furthermore, this typology is *restrictive* because it makes the strong claim that evaluation of a structure is based only on the well-formedness of its substructures. This thus discludes unattested 'counting' patterns, such as the H centering pattern predicted by generalized ALIGN constraints (Eisner, 1997b). It also allows for a theory of learning these constraints, discussed in more detail in Chapter 8. This is not to say that $\mathfrak{L}^{NL}_G$ does not overgenerate at all as a theory of autosegmental well-formedness—one example will be given momentarily in §5.5. However, as that section shows, because the theory is based on substructures, is it is simple to further restrict its typological predictions by studying the *types* of substructures that are found in phonology.

Finally, one potential criticism of the analyses based in $\mathfrak{L}^{NL}_G$ given here is that the patterns analyzed in this section have traditionally been viewed as a mapping of a structure with an underlying, unassociated melody to a fully associated one. As $\mathfrak{L}^{NL}_G$, at least for the moment, is a theory of surface constraints, phonologists may thus question its validity. To this I have two responses.

First, this is not as drastic a departure from previous theories as it may seem. For example, Zoll (2003)'s OTM analyses from §2.3.1.2 entirely depended on the ranking orders of Markedness constraints governing surface well-formedness. The relative ranking of Faithfulness, governing the mapping from underlying to surface forms, played no role in language-specific variation. Thus, although Zoll does not explicitly state so, her analysis highlights how language-specific variation in these tone association patterns is, essentially, variation in surface well-formedness.

Second, although the present analysis focuses on the local character of this surface variation, it does not reject this view of association patterns as mappings. Chandlee (2014); Chandlee et al. (2015) show how locality over strings can be extended to string-to-string mappings, and can be exploited for learning. Chapter 8 of this dissertation thus begin such a local theory of mappings over APGs which takes advantage of the result in this and the following chapter that major tone patterns are local on the surface.

5.5 Further Restricting the Theory

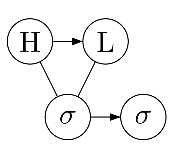
The preceding section showed how $\mathfrak{L}^{NL}_G$ is sufficiently expressive to capture language-specific variation in surface association for the tone-mapping patterns discussed in Chapter 2. The following chapter will show how it is also expressive enough to also capture the long-distance processes discussed in Chapter 2. It will also be discussed how $\mathfrak{L}^{NL}_G$ is the most restrictive logical language (compared to, e.g., a full propositional logic $\mathfrak{L}^P$ over graphs). However, as a theory of surface well-formedness $\mathfrak{L}^{NL}_G$ it overgenerates at least one type of unattested pattern: as discussed below, constraints in $\mathfrak{L}^{NL}_G$ can allow for independent behavior of contours. This issue, though, reveals another potential strength in graph-based logics over string-based ones. By restricting the *type* of subgraph allowed as a literal of the logic, we get a theory restrictive enough to excludes this pattern. This level of restrictiveness is not available simply by adjusting the power of the logic. It also raises an interesting question: if banned substructure constraints are the best theory for well-formedness in phonology, what *kind* of substructures are relevant? Unfortunately, a full answer to this question is beyond the purview of this dissertation, which is simply to explore the adequacy of banned substructure constraints over graphs as a theory of structural well-formedness in phonology. However, this section will discuss one possibility for restricting the theory further based on the type of graph allowed as a literal.

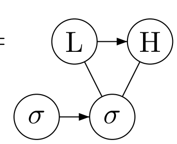
The above discussion argued that one of the strengths of $\mathfrak{L}^{NL}_{G}$ as a theory of surface well-formedness is that it was able to capture distinct association patterns for different tone phonemes, such as in Kukuya. In Kukuya, H cannot spread if it is noninitial or nonfinal, whereas L has no such restriction. This was captured in (5.61) by banning substructures corresponding to spreading nonfinal H and noninitial H autosegments, but by leaving L unrestricted.

However, this independence is not restricted in $\mathfrak{L}^{NL}_{G}$ to single tonal phonemes. Imagine a pattern in which falling contours only appear on the right edge of the word, but rising contours only appear on the left edge of the word. In terms of strings, this looks like this:

(5.63) F, HF, LF, LLLF, HHLF, ...
R, RH, RL, RLLL, RHHL, ...
*FH, *LR, *FLL, *LLR, ...

To the best of my knowledge, this pattern is unattested. However, describing the graph equivalent of this pattern is quite simple in $\mathfrak{L}^{NL}_{G}$. The following literals pick out a non-final F and a non-initial R. Both are very similar to (5.22a) from §5.3.1.

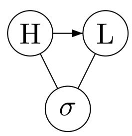
(5.64) a. $\phi_{NF\text{-}F} =$ 

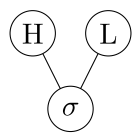
b. $\phi_{NI\text{-}R} =$ 

The following statement in $\mathfrak{L}^{NL}_{G}$ banning these two substructures effectively restricts falling tones to the right edge of the word and rising tones to the right:

(5.65) $\neg\phi_{NF\text{-}F} \wedge \neg\phi_{NF\text{-}R}$

However, there is a very natural concept in graph theory that can exclude this pattern from the predicted range of $\mathfrak{L}_G^{NL}$. Note that in the graph in $\phi_{NF\text{-}F}$ there is a *path* of edges and arcs from the H node to itself: the arc from H to L, the edge from L to the initial σ node, and then the edge from σ to H. (Such paths also exist for the L and initial σ nodes.) Such a path is called a *cycle*, and a graph which contains no such path is *acyclic*. As we've already stipulated in §5.2.1 that literals must be *connected* graphs, we can thus further add the stipulation that they are *acyclic*.[2] This would invalidate the graphs in (5.64) as literals. In fact, any subgraph distinguishing a HL contour from a LH contour would be banned, as the full graph representations of such contours necessarily includes a cycle (an example is given in (5.66a)). However, $\mathfrak{L}_G^{NL}$ constraints could still refer to contours in general, as a graph indicating a contour is simply a doubly-associated timing tier node (as shown in (5.66b)).

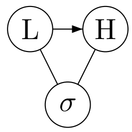
(5.66) a. 

b. 

Thus, restricting graph literals to acyclic graphs would remove any patterns in which contours behaved independently. Adopting such a theory of graph literals would have no effect on the analyses presented in this chapter, with one exception. Hausa was shown to have a contour-independent generalization: falling tones are allowed, while rising tones are not. Thus, the analysis in §5.3.2 invoked the cyclic literal ϕ_R in order to ban rising, but not falling tones (p. 164).[3]

[2] A connected, acyclic, undirected graph is called a *tree*. We can alternatively posit that the *underlying graph* (i.e., where the arcs have been replaced with edges) of each literal must be a tree).

[3] Contrast this with ϕ_{Cont} from §5.3.5 (p. 171), equivalent to (5.66b) above, which can be used to ban contours altogether, and is acyclic.

(5.67) $\phi_R =$ 

The same generalization occurs in Hirosaki Japanese, as will be mentioned in the next chapter. The one attested kind of contour-independent generalization, then, is whether or not certain contours are allowed at all. As ϕ_R from Hausa illustrates, this cannot be described in $\mathfrak{L}^{NL}_G$ assuming universal set of graph primitives $APG(\Gamma)$ for $\Gamma = \{$H, L, F, R, C$\}$. However, banning individual contours (i.e., keeping F but removing R) could be left up to the alphabet of APG primitives. In other words, unlike Mende, for which $\Gamma = \{$H, L, F, R, C$\}$, the Hausa pattern could be described with APGs over $\Gamma = \{$H, L, F$\}$—making a constraint against R contours unnecessary. This makes Γ and g language-specific, but not in an unreasonable way—a learner may only include an F contour in its alphabet of graph primitives if it appears in any input strings (more on learning from strings can be found in Chapter 8). Furthermore, allowing languages to choose a *subset* of some universal set of graph literals (e.g., $\{$H, L, F, R, C$\}$ used in this chapter), combined with a theory of graph literals which may only be acyclic, makes the correct prediction that the only kind of contour-specific constraint—e.g., a constraint that applies to R but not F—is to choose what contours are present in the language.

Again, a full exploration of the consequences of constraining the *type* of graph literals that should allowed in a phonological theory based in $\mathfrak{L}^{NL}_G$ is outside of the central goals of this dissertation. However, this section has shown that there is at least reasonable way of restricting a theory of phonological well-formedness based in $\mathfrak{L}^{NL}_G$ through constraining the type of graphs that are allowed as literals.

5.6 Conclusion

To conclude, this chapter has introduced a theory of language-specific autosegmental well-formedness in tone patterns through constraints on the APGs defined in Chapter 4 based on the logical language $\mathfrak{L}^{NL}_G$. It was then shown how statements in

$\mathfrak{L}_G^{NL}$ can successfuly describe the directional, quality-specific, and positional generalizations from the tone mapping patterns of Mende, Hausa, Kukuya, and N. Karanga Shona originally discussed in Chapter 2. The theory was then compared favorably to previous analyses of these patterns, for the following reasons. Like OT, it directly addresses the surface well-formedness generalizations at the center of the patterns. However, unlike previous derivational and OT analyses, it is fundamentally *local* in a well-defined way allows for a clear, yet restrictive typological theory. Furthermore, it was shown how $\mathfrak{L}_G^{NL}$ may be further restricted by only considering *acyclic* graph literals.

One additional strength of previous analyses discussed in Chapter 2 not addressed in this chapter was their ability to use autosegmental representations, which was their ability to capture long-distance patterns in a 'local' manner. The following chapter shows how $\mathfrak{L}_G^{NL}$ can also do this, and furthermore shows how not only that its expressivity can be understood in a precise way, but that this expressivity forms a tight fit to the attested typology of tone.

Chapter 6

BANNED SUBGRAPH GRAMMARS AND LONG-DISTANCE PHENOMENA

The purpose of this chapter is to show how $\mathfrak{L}_G^{NL}$ also provides a restrictive theory of long-distance tonal patterns. The previous chapter showed how the $\mathfrak{L}_G^{NL}$ Hypothesis, which stated that tone well-formedness patterns are fundamentally local over autosegmental structures, was borne out in with respect to tone mapping patterns, as the major kinds of language-specific well-formedness generalizations in these patterns were found to be describable in $\mathfrak{L}_G^{NL}$. It then argued that, through their focus on the local nature of the generalizations on the surface, analyses in $\mathfrak{L}_G^{NL}$ better capture the nature of the variation in these patterns than previous derivational and OT analyses.

These strengths of $\mathfrak{L}_G^{NL}$ derive from the notions of locality first introduced for strings with $\mathfrak{L}^{NL}$ and for string-based tiers in $\mathfrak{L}_T^{NL}$ in Chapter 3. However, that chapter also showed how the 'long-distance' tone patterns introduced in Chapter 2—namely, unbounded tone plateauing and the accent patterns of Hirosaki Japanese and Wan Japanese—are beyond the expressive power of $\mathfrak{L}^{NL}$ and $\mathfrak{L}_T^{NL}$. This thus raises a crucial question for $\mathfrak{L}_G^{NL}$—can it describe these long-distance patterns? If it is to be a viable theory of tonal well-formedness, it must, as Chapter 2 showed how these three patterns could be described in previous frameworks.

This chapter shows, through analyses of UTP, Hirosaki Japanese, and Wan Japanese with $\mathfrak{L}_G^{NL}$ constraints, that the answer to this question is positive. This means that the mathematical approach taken here is in agreement with traditional autosegmental work arguing that APRs allow for seemingly non-local phenomena to be analyzed locally (e.g., Odden, 1994). Of course, as shall be discussed below, an important difference is that work like Odden (1994) defines locality as strict adjacency, whereas $\mathfrak{L}^{NL}_G$ defines it in terms of substructures of a finite size. This means that, unlike Odden (1994)'s definition of locality, the $\mathfrak{L}^{NL}_G$ notion of locality is not dependent on underspecification. That is not to say, however, that it does not require any representational assumptions. As discussed below, there are two representational assumptions that allow the three long-distance patterns to be described locally with $\mathfrak{L}^{NL}_G$. One is adherence to the OCP, which allows spans of a particular tone to be treated on the melody tier as a single unit. Another, specific to the analysis of Wan Japanese, is allowing morphological information on the tonal tier. However, both of these assumptions are argued to be reasonable, and in fact the analyses here show, in a new way, how they are directly linked to locality.

The second goal of this chapter is to compare $\mathfrak{L}^{NL}_G$ to the string-based logical theories of phonological well-formedness discussed in Chapter 3. As $\mathfrak{L}^{NL}_G$ is sufficiently powerful to capture attested patterns that $\mathfrak{L}^{NL}$ or $\mathfrak{L}^{NL}_T$ could not, it thus more expressive than these classes of grammars. This thus presents a new way of looking at the expressivity of local autosegmental grammars versus local grammars over strings. Furthermore, it is shown how $\mathfrak{L}^{NL}_G$ does not overgenerate in the same way as $\mathfrak{L}^{P}$, the full propositional logic over strings. This logic was shown in Chapter 3 to generate patterns such as $L(\phi_{IH,FH})$, the unattested pattern in which the final TBU of a word must be high if the initial TBU is high as well. Thus, compared to string-based logics, the graph-based $\mathfrak{L}^{NL}_G$ is a better fit to the typology of tone.

This chapter is structured as follows. First, in order to facilitate comparison with string-based logical theories of well-formedness, §6.1 describes how to directly relate string sets and graph sets, drawing on the concatenation operation defined in Chapter 4. Then §6.2 gives analyses in $\mathfrak{L}^{NL}_G$ for L_{UTP}, L_{HJ}, and L_{WJ}, and §6.3 makes clear the representational assumptions which make these analyses possible. Finally, §6.4 discusses in detail how $\mathfrak{L}^{NL}_G$ compares favorably to string-based logics.

6.1 Relating Formal Languages and Graph Sets

As the focus of this chapter is the tone patterns shown in Chapter 3 to be long-distance when viewed as patterns over strings of TBUs, it is important to explicitly relate statements in $\mathfrak{L}_G^{NL}$ to string patterns. One reason for this is to compare the expressivity of $\mathfrak{L}_G^{NL}$ grammars to the string-based $\mathfrak{L}^{NL}$ and $\mathfrak{L}_T^{NL}$ of Chapter 3—in essence, this is an explicit study of the expressivity of locality over autosegmental representations when compared to locality over strings. The other reason for doing this is to get an understanding for the structural properties of autosegmental representations that allow for this difference in expressivity.

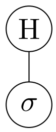
Statements in $\mathfrak{L}_G^{NL}$ can be directly related to string patterns thanks to the method given in Chapter 4 for directly comparing strings and APGs. Given an alphabet of APG primitives Γ, the set of strings in Γ^* specified by a statement $\phi \in \mathfrak{L}_G^{NL}$ can be defined as follows:

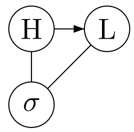
Definition 24 (The set of strings specified by a statement in $\mathfrak{L}_G^{NL}$) *For a statement $\phi \in \mathfrak{L}_G^{NL}$, the set $L(\phi) \subseteq \Gamma^*$ of strings in Γ^* specified by ϕ is defined as:*

$$L(\phi) = \{w \in \Gamma^* | g(w) \models \phi\}$$

In other words, it is the set of strings corresponding to the set of graphs specified by ϕ. Because g directly relates strings and graphs, this is not a fundamentally different way of thinking about ϕ—it just shifts the emphasis from the set of graphs ϕ specifies to its corresponding set of strings.

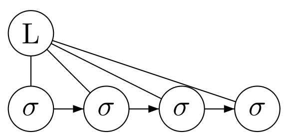
The remainder of this chapter shows that the string patterns shown in Chapter 3 not to be describable by banned substructure constraints in $\mathfrak{L}^{NL}$ or $\mathfrak{L}_T^{NL}$ are describable by statements in $\mathfrak{L}_G^{NL}$. In phonological terms, this is because the APGs reveal two important things about the underlying structure of strings. One, the APG primitives reveal the internal structure of string symbols. Thus, for example, given the example Γ that we have been using, H and F are explicitly related, because g(H) and g(F) both begin with a H node on the melody tier.

(6.1) $g(\mathrm{H}) =$ 

$g(\mathrm{F}) =$ 

This advantage will be made clear in the analysis of the cooccurrence restriction on H and F in Hirosaki Japanese.

Two, APGs reveal relationships between symbols in a string. Thus, in the string LLLLL, in g(LLLLL) each timing tier unit is related to the same melody node. (Of course, as to discussed further below, this is because the OCP is assumed to be in effect.)

(6.2) $g(\mathrm{LLLL}) =$ 

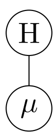
As shall be seen, this allows local statements in $\mathfrak{L}^{NL}_{G}$ to capture long-distance relationships between symbols in a string. As can be seen in (6.2), TBUs which are separated by some distance in the string representation are local in the graph if they are associated to the same melody node (or melody nodes within a fixed distance of each other). In all three of the analyses presented in this chapter, we shall see that this allows local constraints over the melody to model generalizations which appear long-distance when viewed over strings of TBUs.

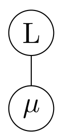
Importantly, however, as shown in Chapter 4, APGs are restricted by the NCC, and so cannot represent arbitrary relationships between units in a string. As a result, the local statements in $\mathfrak{L}^{NL}_{G}$ do not overgenerate in the way that more powerful string logics do. As an example, §6.4 shows how $L(\phi_{IH,FH})$, the unattested tone pattern shown to be overgenerated by $\mathfrak{L}^{P}$ in Chapter 3, is not describable by $\mathfrak{L}^{NL}_{G}$.

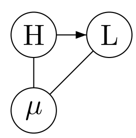
First, however, it is important to show how $\mathfrak{L}^{NL}_{G}$ can capture the non-local string patterns discussed in Chapter 3. This is the subject of the following section.

6.2 Analyses of Long-distance Phenomena with Banned Subgraph Grammars

This section shows how statements in $\mathfrak{L}_G^{NL}$ can capture the three non-local patterns introduced in Chapter 3: L_{UTP} (the surface pattern of unbounded tone plateauing), L_{HJ} (Hirosaki Japanese), and L_{WJ} (Wan Japanese). As shown above, this can be done by casting these languages as subsets of Γ^* and relating strings in these formal languages to graphs. Importantly, this chapter will use slightly different versions of Γ as in the previous chapter. All use some subset of $\{\mathrm{H}, \mathrm{L}, \mathrm{F}\}$ for Γ (none of the patterns involve rising tones), and additionally L_{WJ} requires the morpheme boundary '-'. Additionally, as the Japanese dialects described here use mora (μ) instead of the syllable as the TBU (Haraguchi, 1977; Breteler, 2013), g will map these string symbols to APG primitives whose labelling alphabet uses the symbol μ on the timing tier:

(6.3) $g(\mathrm{H}) =$ 

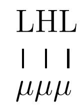
$g(\mathrm{L}) =$ 

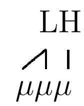
$g(\mathrm{F}) =$ 

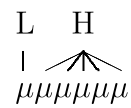
The values for $g()$ for the string symbols below will thus be assumed to be as in (6.3). The discussion of L_{WJ} will also detail the value of g(-).

6.2.1 Unbounded tone plateauing

Recall that the process of unbounded tone plateauing results in a surface pattern in which there may be only one, unbroken plateau of H tones across the domain. The examples below in (6.4) repeat the crucial data (originally from (2.36), p. 38, in Chapter 2) in terms of strings and APRs.

(6.4) a. ‘chopper’ mutéma LHL 

b. ‘log’ kisikí LLH 

c. ‘log chopper’ mutémá+bísíkí LHHHHH 

d. ” ” *mutéma+bisikí *LHLLLH 

As from Chapter 3, §3.6.3, this translates to the following string pattern, repeated from (3.29) and (3.30) (p. 106) in that section.

(6.5) a. Strings in L_{UTP}

LLL, LLLL, LHL, LLHL, LHLLLL, LLLLHL, HHH, LHHHL, LHHHHH, LLLHHL, LLHHHL, ...

b. Strings not in L_{UTP}

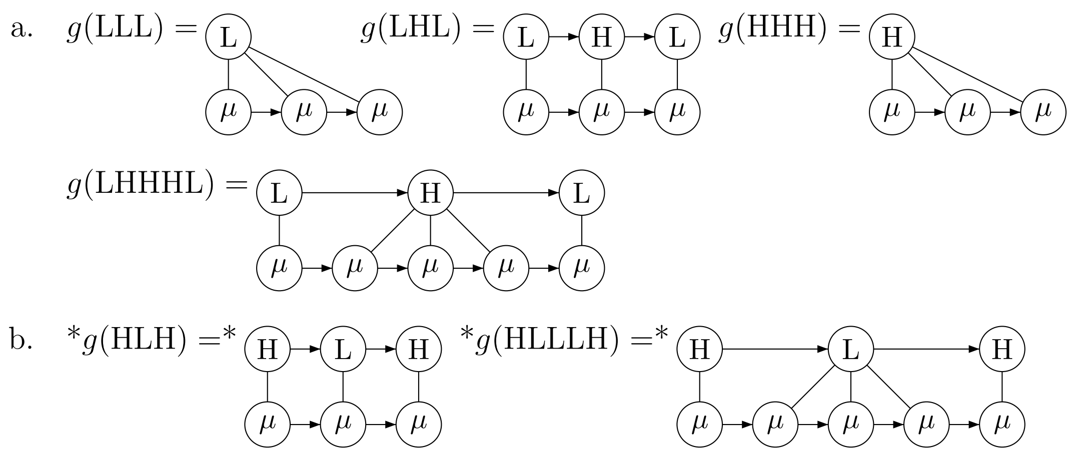
HLH, HLLH, HLHH, HLLLH, HHHHHLH, HLLLLLLH, ...

Recall that this pattern cannot be described by statements in either $\mathfrak{L}^{NL}$ or $\mathfrak{L}^{NL}_T$. The reason is that banned substructure constraints over strings cannot enforce the requirement that *at most one* plateau of Hs exists in each string. However, this pattern can be captured very easily with a statement $\mathfrak{L}^{NL}_G$ restricting banning APGs with more than one H node.

As L_{UTP} consists of strings of Hs and Ls, we can consider the set $APG(\Gamma)$ of graphs over $\Gamma = \{\mathrm{H}, \mathrm{L}\}$.[1] Example graphs corresponding to some of the strings in L_{UTP} from (7.50a) are given below in (6.6a), and examples of graphs corresponding to strings *not* in L_{UTP} are given in (6.6b).

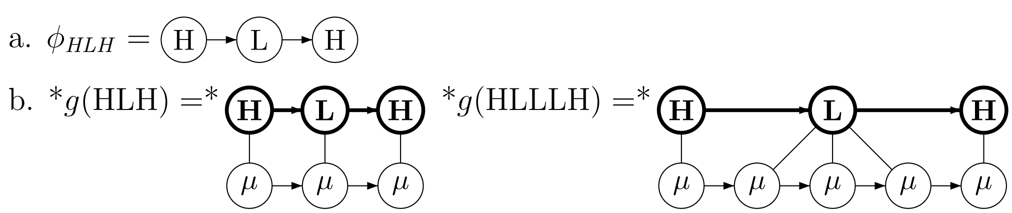
[1] Thus, $APG(\Gamma)$ will not contain any contours. Note that this effect was achieved in Chapter 5, §5.3.5 for N. Karanga Shona by including the negative literal $\neg\phi_{Cont}$, which matched any contour. These methods are essentially equivalent.

(6.6)



The graphs for the strings *not* in the pattern represent two (or more) H tones which separated by a 'trough' of L toned TBUs—in other words, two Hs which have failed to form a plateau. In the APGs, any such graph is going to contain a sequence HLH of melody tier nodes. This set of graphs can thus be isolated with the literal ϕ_{HLH}, which the reader can recall was used in Mende to ban HLH melodies and melodies with more than three tones. This literal is repeated in (6.12a) below from (5.21) from Chapter 5, §5.3.1 (p. 160). That it is a subgraph of the graphs in (6.6b) is exemplified in (6.7b).

(6.7)



Note that, in terms of strings, ϕ_{HLH} can pick out two Hs separated by any number of L-toned TBUs. This is because, as the concatenation operation was defined to obey the OCP, adjacent L-toned TBUs will be associated to the same L node in the corresponding APG. Thus, any string in which two Hs are separated by L-toned TBUs will have in its corresponding APG two H nodes on the melody tier separated by a single L node.

Thus, taking the negative of ϕ_{HLH} bans any such 'trough' of H tones separated by L tones, and so the UTP pattern can be described by this single negative literal:

(6.8) $L_{UTP} = L(\neg\phi_{HLH})$

Note that this is an explicit implementation of the *Trough constraint in tonal Optimality Theory, discussed in Chapter 2 (p. 66), militating against L tones intervening between a H tone. Through this single constraint, $\mathfrak{L}^{NL}_{G}$ is able to describe the surface UTP pattern, which was beyond the power of the banned substructure logics over strings $\mathfrak{L}^{NL}$ and $\mathfrak{L}^{NL}_{T}$. Crucially, this depended on the OCP ensuring that adjacent L-toned TBUs in the string were represented in the graph as timing tier nodes associated to the same L node. This will be discussed further in §6.2.4.

6.2.2 Hirosaki Japanese

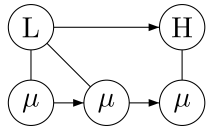
Recall from §3.6.1 that in Hirosaki Japanese (Haraguchi, 1977) words, at most one H or F can appear in a word, and F may only appear word-finally. Each word must contain either a H or F, and cannot contain both. This full pattern is given as L_{HJ} in (3.22), repeated below in (6.9).

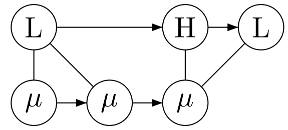
(6.9)

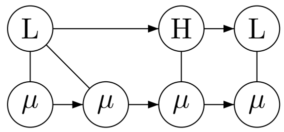
In L_{HJ}	Not in L_{HJ}
H, F, LH, LF, HL,	L, LL, HH, HF,
LLH, LLF, LHL, HLL,	LLL, FLL, LFL, HLF,
LLLH, LLLF, LLHL, LHLL,	LLLL, LLFL, LFLL,
HLLL, ...	FLLL, HLLF, ...

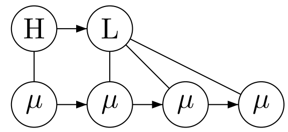
As discussed in §3.6.1, L_{HJ} is not describable by statements in neither $\mathfrak{L}^{NL}$ nor $\mathfrak{L}^{NL}_{T}$. Intuitively, this is because a H can appear anywhere in the word, and so the restriction that there must be exactly one H or F cannot be implemented by referring to local substructures.

However, interpreting these strings using APGs, these restrictions on H can be handled by banned substructure constraints on the melody tier. The set of APGs corresponding to the srings in L_{HJ} is the set of APGs with exactly one H node which does not associate to multiple TBUs. Falling contours are restricted to word-final TBUs. Some examples are given below.

(6.10) $g(\mathrm{LLH}) =$ 

$g(\mathrm{LLF}) =$ 

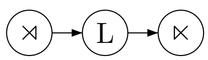
$g(\mathrm{LLHL}) =$ 

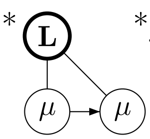
$g(\mathrm{HLLL}) =$ 

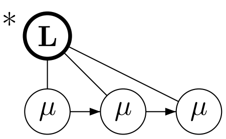
As rising tones do not appear, consider $\Gamma = \{\mathrm{H}, \mathrm{L}, \mathrm{F}\}$. The set of APGs for Hirosaki Japanese is a subset of $APG(\Gamma)$ defined by a statement in $\mathfrak{L}^{NL}_{G}$ made up of the following banned subgraph constraints.

First, we must distinguish words with no H tones or more than one H tone. For the former, the APG equivalent of any string of L tones will have the melody tier specified in $\phi_{\rtimes L \ltimes}$ in (6.11a). Again, this is because concatenation will merge consecutive L nodes in the primitives into a single L node in the resulting APG (some examples are given in (6.11b)).

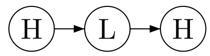
(6.11)

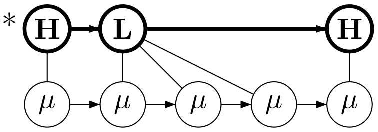
a. $\phi_{\rtimes \mathrm{L} \ltimes} =$ 

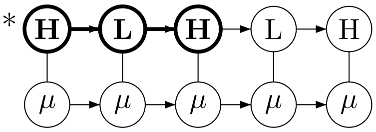
b. $^*g(\mathrm{LL}) = {}^*$ 

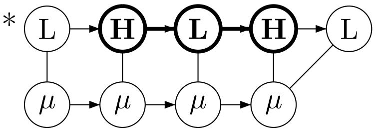
$^*g(\mathrm{LLL}) = {}^*$ 

Similarly, for any string with more than one H toned TBU, such as HLLLH, its corresponding graph will contain the subgraph specified in the literal ϕ_{HLH}, just discussed in the previous section. Examples of how ϕ_{HLH} can be used to ban strings with multiple, nonconsecutive Hs are given in (6.12b). (APGs corresponding to strings with consecutive Hs will result in a multiply associated H node, which will be dealt with momentarily.)

(6.12) a. ϕ_{HLH} = 

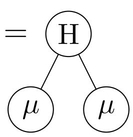
b. *g(HLLLH) = * 

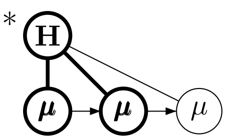
*g(HLHLH) = * 

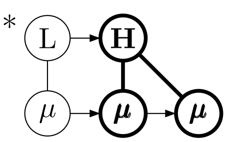
*g(LHLF) = * 

Note, importantly, that this constraint also captures the fact that H and F TBUs cannot appear in the same string. For example, *g(LHLF) contains the subgraph in ϕ_{HLH}, because an F string corresponds to a timing tier node associated to a H and a L tone. Thus, the melody for the set of APGs in the Hirosaki Japanese pattern can be specified by the $\mathfrak{L}^{NL}_{G}$ statement $\neg\phi_{\rtimes L \ltimes} \wedge \neg\phi_{HLH}$. However, we must also restrict the associations of the melody nodes to TBUs.

First, the H tone in Hirosaki Japanese does not spread at all. APGs in which an H node is multiply associated can be picked out with the following literal.

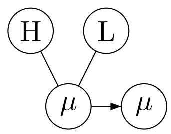
(6.13) a. ϕ_{H^2} = 

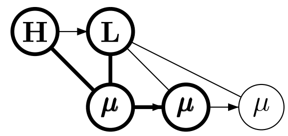
b. *g(HHH) = * 

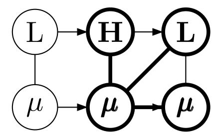
*g(LHH) = * 

Note that this literal is different from $\phi_{NF\text{-}H^2}$ and $\phi_{NI\text{-}H^2}$, which in Chapter 5 were used to ban initial multiply associated H tones and final multiply associated H tones, respectively. In particular, both were invoked in Kukuya to ban a multiply associated H tone *in the presence of another tone*—in Kukuya, multiple association of H was allowed, but only if H was the only tone in the melody tier. However, in Hirosaki Japanese, H *never* multiply associates, so the relevant subgraph is that of ϕ_{H^2}.

Finally, falling contours only occur on the final TBU in Hirosaki Japanese. This can be accomplished with $\phi_{NF\text{-}Cont}$, originally introduced in (5.22a) in Chapter 5, §5.3.1 (p. 160) in order to capture part of the 'directional' association pattern of Mende. This literal is repreated below in (6.14a).

(6.14) a. $\phi_{NF\text{-}Cont} =$ 

b. $^*g(\text{FLL}) = {}^*$ 

$^*g(\text{LFL}) = {}^*$ 

A conjunction of negative graph literals in $\mathfrak{L}^{NL}_{G}$ banning the subgraphs in the above primitives can thus describe the Hirosaki Japanese pattern:

(6.15) $L_{HJ} = L(\neg\phi_{\rtimes \mathrm{L} \ltimes} \wedge \neg\phi_{HLH} \wedge \neg\phi_{H^2} \wedge \neg\phi_{NF\text{-}Cont})$

Again, (6.15) captures the Hirosaki Japanese pattern with *local* constraints on the melodies and associations on APGs. This is a strikingly different result than from logical grammars over strings, which could not capture Hirosaki Japanese with local statements in either $\mathfrak{L}^{NL}$ or $\mathfrak{L}^{NL}_{T}$. There are two reasons for this. One, because APGs isolate the melody from its associated TBUs, the constraint that only one H can appear is reduced to banning a local substructure ($\neg\phi_{HLH}$). As mentioned in the discussion in the previous section regarding UTP, this is because intermediate L-toned TBUs are, through adherence to the OCP, associated to the same L-tone. Two, recall from the discussion in Chapter 3, §3.5.1 that the constraint against the cooccurrence of H and F TBUs and the constraint that F must be word-final could not both be captured in a single statement in either $\mathfrak{L}^{NL}$ or $\mathfrak{L}^{NL}_{T}$. However, with APGs, it is made explicit that H and F TBUs in the string both correspond to timing tier nodes associated to a H tone. Thus, the coocurrence constraint on H and F TBUs falls out from the constraint against two H tones (as seen in (6.12)). That falling contours must be restricted to final position is, as has already been seen in the previous chapter, also a local constraint over APGs.

6.2.3 Wan Japanese

Finally, $\mathfrak{L}^{NL}_G$ can capture patterns whose complexity derives from dependency on morphological information. In Chapter 2, we saw that Wan Japanese had a class of words, Type β, where words without a suffix are pronounced with a H^{n}LHL pattern (with the initial H^{n}L truncated in smaller words), and words with a suffix are pronounced with a H^{n}LHH-H^{m}L, where m and n are based on the length of the stem and suffix morphemes, respectively (here too the initial H^{n}L truncated in smaller stems, and the initial H^m truncated in smaller suffixes). The string pattern, as first given in Chapter 3 in (3.24) (3.25) (p. 104) is repeated below in (6.16), with '-' indicating a morpheme boundary.

(6.16) $L_{WJ} =$ (without suffix) HL, LHL, HLHL, HHLHL, HHHLHL, ...
(with suffix) HH-L, HH-HL, HH-HHL, HH-HHHL, ...,
LHH-L, LHH-HL, LHH-HHL, ...,
HLHH-L, HLHH-HL, HLHH-HHL, ...,
HHLHH-L, HHLHH-HL, HHLHH-HHL, ...
HHHLHH-L, HHHLHH-HL, HHHLHH-HHL, ...
...

L_{WJ} could not be described by statements in either $\mathfrak{L}^{NL}$ or $\mathfrak{L}^{NL}_T$. This is because the presence of the '-' boundary bears on the realization of the last three TBUs: if it is present, then the word must end in HHL$\ltimes$, -HL$\ltimes$, or -L$\ltimes$. If it is *not* present, then the word cannot end HHL$\ltimes$ (or any of the other above options). Because '-' is in principle not restricted on where it may appear in the string, there is thus a non-local dependency between it and the last three TBUs in the word.

However, L_{WJ} can be captured in $\mathfrak{L}^{NL}_G$, although we must first address the question of what the correct autosegmental representation is for the strings containing suffixes. Recall from Chapter 2 (p. 44) that there are two possibilities: as in (6.17a), that a HLHL melody associates such that the second H associates to the final mora in the stem and all but the final mora in the suffixes; or, as in (6.17b), the stem takes its phrase-medial HLH melody and association paradigm, whereas the suffix domain takes a separate, HL melody, with a '-' on the tonal tier marking the boundary between the two melodies.

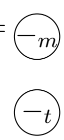
(6.17) a. $\mathrm{H}^n\mathrm{LHH}\text{-}\mathrm{H}^m\mathrm{L}$ =

b. $\mathrm{H}^n\mathrm{LHH}\text{-}\mathrm{H}^m\mathrm{L}$ =

While, as discussed in that chapter, previous analyses have assumed (6.17a), the following will assume (6.17b). The difference between the two is not trivial with respect to describability in $\mathfrak{L}^{NL}_{G}$, and this will be discussed further in §6.3.3.

Thus, let us consider $\Gamma = \{\text{-}, \mathrm{H}, \mathrm{L}\}$, where g(-) is as in (6.18) and the labelling alphabet for the graphs includes labels on each tier for this morpheme boundary; i.e. $\Sigma = \{\mathrm{H}, \mathrm{L}, \mu, -_t, -_m\}$.

(6.18) g(-) =

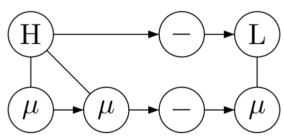


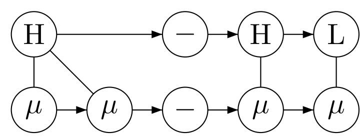
The reader will recall this is exactly like the graph for $g(\#)$ mentioned in Chapter 4, §4.5.2. Also as with the graphs in that section, the following will omit the subscripts from $-_t$ and $-_m$, as they will be clear from context.

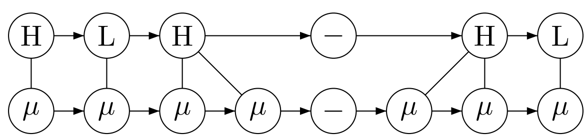
One further point must be made about '-' and thus g(-). In the available descriptions of Wan Japanese, only one '-' is allowed per string (as only the division between the stem and the following suffixes is relevant to the pattern). As we saw in the discussion in this chapter and Chapter 3 regarding Hirosaki Japanese, this kind of culminative restriction is not local over strings—that is, it cannot be described in $\mathfrak{L}^{NL}$, although it can be described in $\mathfrak{L}^{NL}_{T}$. However, there are two reasons we can shift our focus away from this restriction. One, this is fundamentally a *morphological* restriction, as it is a constraint on the structure in words that they must carry at most one boundary. We could thus assume that such a constraint is imposed by the morphological module of the grammar, and can ignore how it is implemented, as we are currently investigating the locality of the *phonological* grammar. Two, this cumulativity *can* be implemented with $\mathfrak{L}_G^{NL}$ and $APG(\Gamma)$, just as in Hirosaki Japanese. How this can be done for Wan Japanese will become clearer during the discussion of this pattern's particular constraints, specifically in Footnote 3.

Whether we assume the restriction on '-' is morphological or if we implement it in $\mathfrak{L}_G^{NL}$, we can ignore strings in Γ^* which have more than one '-' (and thus graphs in $APG(\Gamma)$ which include only one instance of g(-)) without affecting out conclusions about whether L_{WJ} can be described with local constraints in $\mathfrak{L}_G^{NL}$. Thus, for the sake of simplicity, the following assumes that strings/graphs will only include at most one instance of '-'/g(-).

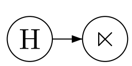
Sample graphs in $APG(\Gamma)$ corresponding to the strings in L_{WJ} which include a suffix are thus as follows:

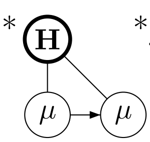
(6.19) g(HH-L) = 

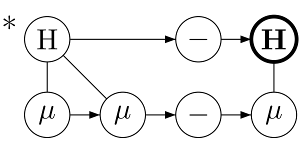
g(HH-HL) = 

g(HLHH-HHL) = 

The full graph set corresponding to the strings in L_{WJ} can be specified by banning the following substructures. To begin, there are a few clear restrictions on the melody tier. First, no words end in a H tone. Graphs with a final H tone all contain the subgraph in $\phi_{H\ltimes}$.

(6.20) a. $\phi_{H\ltimes}$ = 

b. *g(HH) = * 

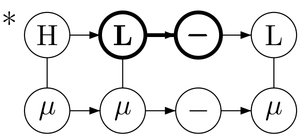
*g(HH-H) = * 

Second, the melody tier is maximally HLHL (ignoring, for the moment, the morpheme boundary on the melody tier). Any longer melodies will contain the subgraph in (6.21).

(6.21) $\phi_{\text{LHLH}} = \text{L} \rightarrow \text{H} \rightarrow \text{L} \rightarrow \text{H}$

Negating $\phi_{H\ltimes}$ and ϕ_{LHLH} thus restricts us to graphs containing the attested melodies HL, LHL, and HLHL in graphs not containing the morpheme boundary.[2] (This is much like the restrictions on possible melodies for Mende, Hausa, and Kukuya in Chapter 5.) For graphs including the morpheme boundary, a few more restrictions are nessary. First, note in (6.16) that a stem never ends in a L tone when suffixed. This can be captured by banning the following structure:

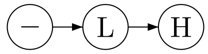
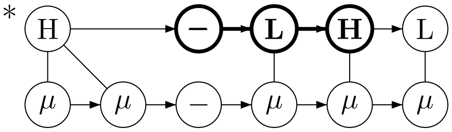
(6.22) a. $\phi_{L\text{-}} = \text{L} \rightarrow -$ b. $*g$(HL-L) = *

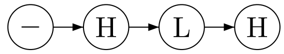


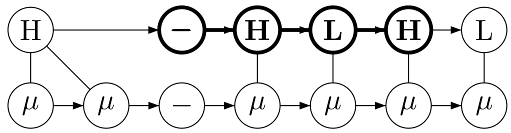
Banning the subgraph in $\phi_{L\text{-}}$, along with that in ϕ_{LHLH}, correctly restricts melodies for suffixed stems to those attested. However, it is still necessary to restrict the melodies on the suffixes. Note that the suffixes exhibit either a L or a HL melody. This can be accomplished using the following literals, whose subgraphs represent melodies longer than L or HL:

[2] There are no data on Wan words smaller than two morae (Breteler, 2013; Kubozono, 2011b; Uwano, 2012), so restrictions on melodies for monomoraic words will not be considered here. If we wish to ban monomoraic words, we can simply ban the subgraph in $\phi_{\rtimes\mu\ltimes}$ below:

(i) $\phi_{\rtimes\mu\ltimes} = \rtimes \rightarrow \mu \rightarrow \ltimes$

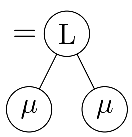
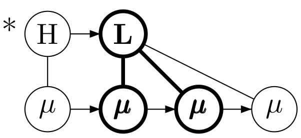
(6.23) a. $\phi_{\text{-}LH}$ =  b. $*g$(HH-LHL) = * 

c. $\phi_{\text{-}HLH}$ = 

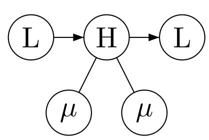
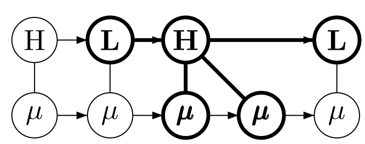
d. $*g$(HH-HLHL) = * 

This concludes the subgraph literals needed to restrict ourselves to graphs with the correct melody tiers.[3]

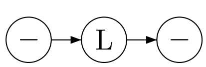
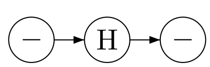
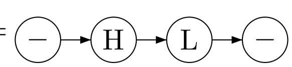
A few additional literals are needed to constrain the *associations* of these melody units to TBUs. Perhaps the most straightforward is the fact that L tones do not spread in Wan Japanese. This can be captured by ϕ_{L^2}, given below, whose subgraph picks out graphs with multiply associated L nodes.

(6.24) a. ϕ_{L^2} =  b. $*g$(HLLL) = * 

A second constraint is on the second H in case there is no suffix, which cannot spread. This can be accomplished iwth the literal ϕ_{LH^2L} below. This is similar to the literal $\phi_{NI\text{-}H^2}$ from Chapter 5's analysis of Hausa. However, we need to specify that there is an L tone on either side of the H, as, in cases where there is a suffix, both Hs may spread.

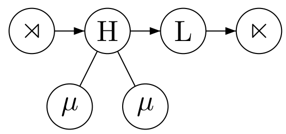
(6.25) a. ϕ_{LH^2L} =  b. g(HLHHL) = 

[3] To implement cumulativity of '-' using graph constraints for this pattern, it is only necessary to consider the following additional literals:

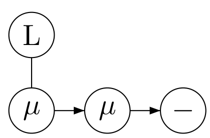
$\phi_{\text{-}L\text{-}}$ =  , $\phi_{\text{-}H\text{-}}$ =  , $\phi_{\text{-}HL\text{-}}$ = 

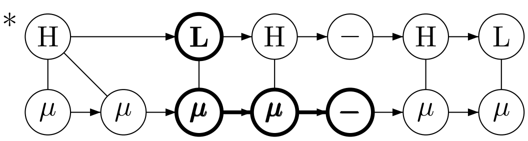
This does not restrict the second H in case of a suffix; note that ϕ_{LH^2L} is not a subgraph of g(HLHH-HHL) in (6.19).

One more similar constraint is necessary, as there is only one word with an HL melody—HL. In order to keep this H from spreading, we must also ban the following substructure.

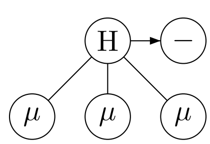
(6.26) $\phi_{\rtimes H^2 L \ltimes} =$ 

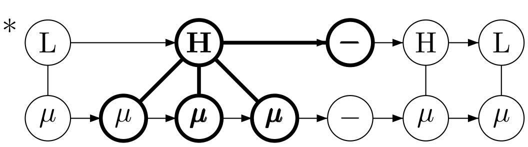
Finally, note that preceding a suffix, a H tone must spread to exactly two TBUs, if they are present. We have already discussed $\phi_{L\text{-}}$, which bans an L tone from appearing immediately before the morpheme boundary. However, it may not also appear two TBUs before this boundary as well; for example, HLHH-HL is a valid string in L_{WJ}, but *HHLH-HL is not. The corresponding graph for such a string will contain the subgraph denoted by $\phi_{L2\text{-}}$ in (6.27) below:

(6.27) a. $\phi_{L2\text{-}} =$ 

b. *g(HHLH-HL) = * 

As L cannot occupy these spaces, a H tone must 'fill' them in, ensuring that there are at least two H TBUs before the morpheme boundary. However, we must also ensure that there are not *more* than two H TBUs preceding the morpheme boundary. This can be achieved by banning the following literal:

(6.28) a. $\phi_{H^3\text{-}} =$ 

b. *g(LHHH-HL) = * 

The string pattern L_{WJ} can thus be described thusly:

(6.29) $L_{WJ} =$

$$L(\neg\phi_{H\ltimes} \wedge \neg\phi_{\mathrm{LHLH}} \wedge \neg\phi_{L\text{-}} \wedge \neg\phi_{\text{-}LH} \wedge \neg\phi_{\text{-}HLH} \wedge \neg\phi_{LH^2L} \wedge \neg\phi_{\rtimes H^2 L\ltimes} \wedge \neg\phi_{L2\text{-}} \wedge \neg\phi_{H^3\text{-}})$$

Thus, L_{WJ}, our final example of a tone pattern which could not be described in $\mathfrak{L}^{NL}$ or $\mathfrak{L}^{NL}_T$, is describable by the above statement in $\mathfrak{L}^{NL}_G$. Again, the complexity of L_{WJ} derives from its long-distance interaction between the morpheme boundary '-' and the tone pattern. However, $\mathfrak{L}^{NL}_G$ was able to capture this by encoding this morphological information on the melody tier. This choice shall be discussed further in §6.3.2 below.

6.2.4 Interim conclusion: subgraph grammars and long-distance phenomena

This section has thus shown how L_{UTP}, L_{HJ}, and L_{WJ} are all describable by statements in $\mathfrak{L}^{NL}_G$. This makes $\mathfrak{L}^{NL}_G$ more expressive than $\mathfrak{L}^{NL}$ and $\mathfrak{L}^{NL}_T$, and, like the derivational theories and OT, sufficient to describe these patterns. As discussed in detail in the previous chapter, these analyses are like OT in that they directly capture the surface generalizations—in fact, in $\neg\phi_{HLH}$ used in the analyses for L_{UTP} and L_{HJ}, we saw a constraint identical to *Trough used in tonal OT—but, crucially, they do so in a *local* way. As shall be discussed momentarily, this means that $\mathfrak{L}^{NL}_G$ is also *restrictive* in a well-defined way. However, it is important to first review what exactly it is about the representational assumptions and $\mathfrak{L}^{NL}_G$ which allow them to describe these patterns, as opposed to the local string grammars of $\mathfrak{L}^{NL}$ and $\mathfrak{L}^{NL}_T$. This is the focus of the next section.

6.3 Representational Assumptions and Graph Locality

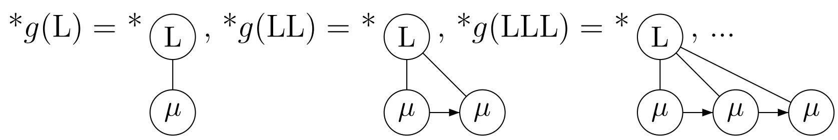
There are two representational assumptions that made the local analyses in the previous section possible. One is that the OCP holds at the surface level, and the other is that, for Wan Japanese, morphological information needs to be included on the tonal tier. The purpose of this section is to discuss these assumptions in detail and show how they are connected to locality.

6.3.1 The OCP and locality

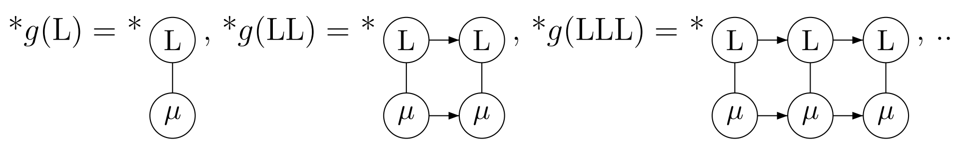
Let us begin with the OCP. Consider the L_{HJ} pattern. The core of this pattern is a restriction that there must be exactly one H or F TBU in a string. As discussed in Chapter 3, this is not describable by $\mathfrak{L}^{NL}$, as there is no finite set of substructures we can ban to remove the set of strings of only L-toned TBUs:

(6.30) *L, *LL, *LLL, *LLLL, *LLLLL, ...

However, we saw in this chapter that, given g and graph concatenation defined in Chapter 4, the APGs corresponding to this set of strings all have the same melody:

(6.31) 

As shown above, this set of strings can be banned by banning this particular melody. Of course, this is dependent on its adherence to the OCP. If concatenation *did not* merge melody end nodes with like labels, and instead 'bridged' them, the corresponding graphs might look like this:

(6.32) 

Banning this set of graphs with a finite set of banned subgraphs would thus fall to the same problems as attempting to ban the set of strings in (6.30) with a finite set of substrings—it is impossible, with a finite number of constraints, to ban a tier of potentially unboundedly many L nodes. *Thus, adhering to the the OCP allows the melody tier to express long-distance constraints locally.* This is similar conceptually to Odden (1994)'s locality condition, but importantly, *it does not require underspecification.*

We saw that this principle also applies to dependencies between units which correspond to long-distance dependencies in strings. For example, it was discussed both in this chapter and in Chapter 3 how the L_{WJ} pattern has a long-distance dependency between '-' and the tones on the final TBUs. All TBUs following the '-' must be of the pattern H^{n}L, but if no '-' is present the melody must be H^{n}LHL (if there are 3 or more TBUs). This cannot be captured by local constraints over strings because no set of substrings of a finite size can ban all strings in which '-' is not followed by a H^{n}L melody. Consider the substring -HHHLH. Banning this substring will correctly ban strings ending with this exact pattern, but not strings with more than three Hs following the '-':

(6.33) a.¬ -HHHLHL b. ✓ *HH-**HHHLH**L

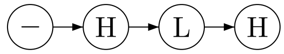
✖ *HH-HHHHLHL

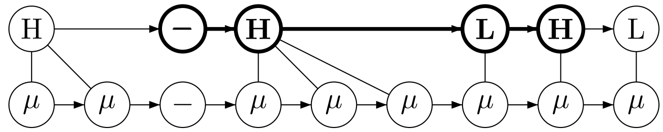
In the above example, *HH-HHHLHL is correctly banned, but *HH-HHHHLHL, because it has too many medial Hs for the substring in the constraint in (6.33a) to match. In general, consider banning any substring -H^{k}LH, where in the middle there is some arbitrary number k of Hs (in the previous example $k = 3$). This will not match a string with $k + 1$ medial Hs:

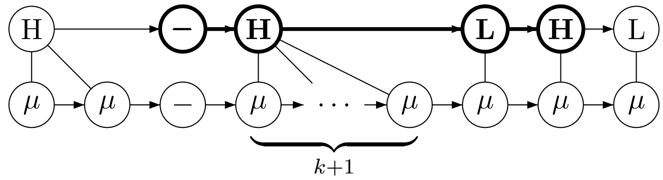
(6.34) a.¬ -H^{k}LH b. ✓ *HH-**H$^{\boldsymbol{k}}$LH**L

✖ *HH-H^{k+1}LHL

However, in APGs, these medial Hs are all represented on the melody tier by a single H node. Thus, $\phi_{\text{-}HLH}$ from (6.23) will match the corresponding graph to any such string. This is illustrated below with the corresponding graphs to *HH-HHHLHL from (6.33b) and *HH-H^{k+1}LHL from (6.34b).

(6.35) a. $\phi_{\text{-}HLH}$ = 

b. *g(HH-HHHLHL) = * 

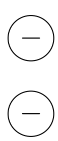
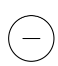
*g(HH-H^{k+1}LHL) = * 

Thus, by referring to the information on the melody tier, local constraints over APGs can capture what correlate to long-distance dependencies in the string pattern L_{WJ}.

This relies on two assumptions, however. One, as already mentioned, is that the OCP is adhered to, collapsing the stretch of medial H TBUs to a single H node on the melody tier. The second assumption is that the morphological information is present on the melody tier—which is crucial for capturing the long-distance dependency. Let us now turn to this second point.

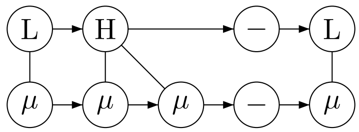
6.3.2 Morphological information and locality

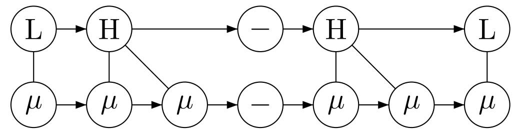
In the APG analysis of L_{WJ} in §6.2.3, the APG primitive corresponding to the morpheme boundary '-' had boundary nodes on both the timing tier and the melody tier, as repeated below in (6.36a). As discussed above, explicitly including a '-' node on the melody tier was what allowed the long-distance nature of the pattern to be captured with local constraints over APGs. The following will show this in more detail, by interpreting '-' as only appearing on the timing tier, as in (6.36a).

(6.36) a. g(-) =  b. g(-) = 

First, graphs corresponding to strings in L_{WJ} with the value of g(-) as in §6.2.3 are as in the examples in (6.37).

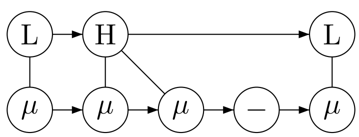
(6.37) a. g(LHH-L) = b. g(LHH-HHL)

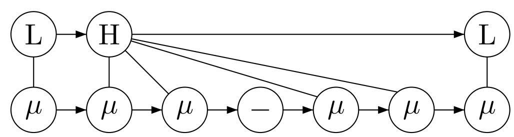




This can be interpreted as knowledge that, in terms of the melody, the span of Hs straddles a boundary. However, it can be criticized on the grounds that it obscures the fact that Wan Japanese in general has a maximal HLHL melody, regardless of whether or not there is a morpheme boundary. This can be made explicit by instead using the value of g(-) given in (6.36b). The graphs in (6.37) would instead look as below in (6.38).

(6.38) a. g(LHH-L) = b. g(LHH-HHL)

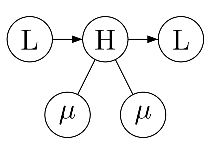




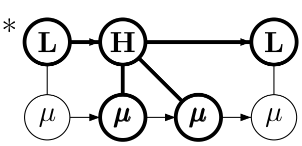
Here, there is only one H node, as the H nodes in consecutive g(H) primitives merging 'over' the timing tier-only g(-) (a similar effect was briefly seen in (4.46) in Chapter 4, §4.5.4, p. 145).

However, the absence of any morphological information on the melody tier makes it impossible to distinguish with local constraints the association behavior of H nodes when the morpheme boundary is present and that of H nodes when the boundary is not present. Recall that when no morpheme boundary is present, a noninitial H is not allowed to spread; this was enforced by ϕ_{LH^2L}, as repeated below from (6.25).

(6.39) a.ϕ_{LH^2L} =

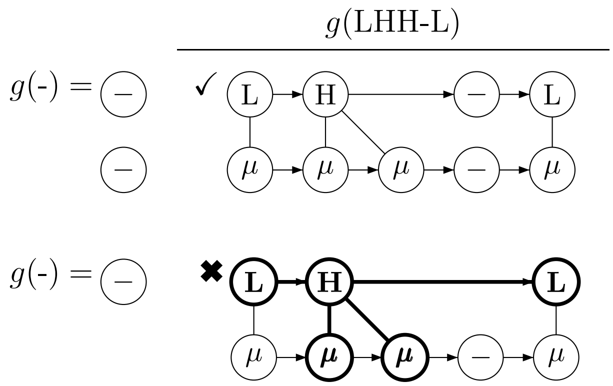


b. *g(LHHL) = *

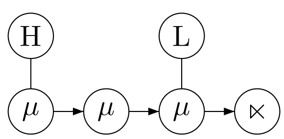
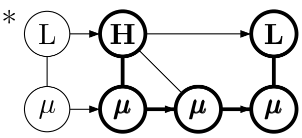


However, without the morpheme boundary appearing on the melody tier, this means that g(LHH-L) also matches the subgraph in ϕ_{LH^2L}. This can be seen in the following, which contrasts g(LHH-L) given the two interpretations of g(-) given in (6.36).

(6.40)

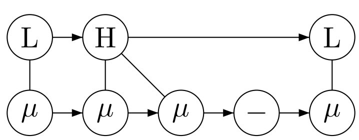


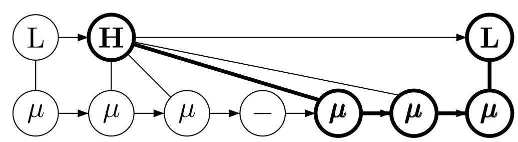
As seen in the lower graph of (6.40), $\neg\phi_{LH^2L}$ would incorrectly ban g(LHH-L). We may try to distinguish g(LHH-L) from *g(LHHL) by instead specifying that H cannot be associated two morae before a final L tone:

(6.41) a. $\phi_{H2L\ltimes}$ =  b. *g(LHHL) = * 

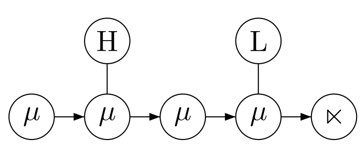
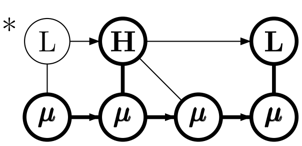
This correctly distinguishes *g(LHHL) from g(LHH-L), as the latter explicitly includes a morpheme boundary node in between the final and penultimate morae. Thus, the subgraph in (6.41a) does not match g(LHH-L, as shown in (6.42a).

(6.42) a. ✓g(LHH-L) b. ✖ g(LHH-HHL)



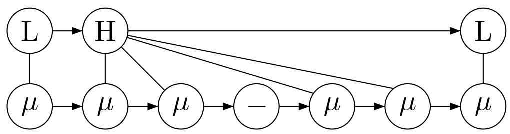


However, the graph g(LHH-HHL), which corresponds to a string in L_{WJ}, incorrectly matches this subgraph. We could try to then distinguish this graph from *g(LHHL) by including more nodes from *g(LHHL) in the constraint:

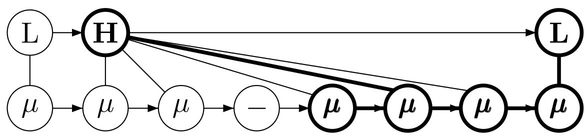
(6.43) a. $\phi_{1H2L\ltimes}$ =  b. *g(LHHL) = * 

This correctly distinguishes $*g$(LHHL) from g(LHH-HHL), as shown below in (6.44a). However, this again incorrectly matches the graph of another string in L_{WJ}, g(LHH-HHHL), as shown in (6.44a).

(6.44) a. ✓ g(LHH-HHL)



b. ✖ g(LHH-HHHL)



It is possible to try and further distinguish these graphs by including more nodes from $*g$(LHHL) in the constraint, but such a constraint would not be able to ban a similar string with a longer stretch of Hs, such as $*g$(LHHHL). We can thus conclude that there is no finite set of subgraphs which is sufficiently general to ban all strings of the shape LH^{n}L while also including all strings of the shape LHH-H^{n}L. Thus, without g(-) including morphological information on the melody tier, it is impossible to capture L_{WJ} with a statement in $\mathfrak{L}_G^{NL}$.

Thus, to review, there are two possible ways to represent the Wan Japanese β pattern—one in which the morphological information is present on the tonal tier, and one in which it is not—and this difference is non-trivial in terms of locality, as the pattern can only be described in terms of $\mathfrak{L}_G^{NL}$ if the morphological information *is* included on the tonal tier. As discussed in Chapter 2, there is currently no empirical evidence to distinguish between the two. For example, no phonetic studies have been done to test whether or not the suffix tones constitutes a separate melody, and it is not clear if such a test is even possible (the reader is referred to p. 44 for the full discussion). However, the discussion in this chapter provides a *theory-internal* argument for including morphological information on the tier: if it is included, then Wan Japanese is, like all the other tone patterns mentioned in this dissertation, local over autosegmental representations. If not, then it is (as far as has been determined) unique in being non-local.

6.3.3 Interim conclusion: representation and locality

This section has explicitly described how describability in $\mathfrak{L}^{NL}_G$ can be dependent on the OCP and morphological information on the tier. This is another contribution of the study of tone patterns in $\mathfrak{L}^{NL}_G$, as it has shown how these two representational assumptions are intimately connected to locality.

6.4 Comparing Banned Subgraph Grammars and String Logics

We have thus so far seen how, given the representational assumptions discussed in the previous section, statements in $\mathfrak{L}^{NL}_G$ could be used to specify sets of strings which were shown in Chapter 3 to be beyond the power of $\mathfrak{L}^{NL}$ and $\mathfrak{L}^{NL}_T$. As these patterns are attested in natural language, this immediately makes $\mathfrak{L}^{NL}_G$ a better theory for tonal well-formedness, as it is *sufficient* to describe the attested range of patterns. However, in order to make a complete comparison to string logics, it is also important to show that $\mathfrak{L}^{NL}_G$ is also *restrictive* relative to more powerful string logics. This section accomplishes this by showing how $L(\phi_{IH,FH})$, the unattested string pattern introduced in Chapter 3 which is describable by the full propositional string logic $\mathfrak{L}^P$, is not describable by $\mathfrak{L}^{NL}_G$, and thus $\mathfrak{L}^{NL}_G$ compares favorably to $\mathfrak{L}^P$ as a theory of phonological well-formedness.

Recall from Chapter 3 that in the pattern $L(\phi_{IH,FH})$, if the first syllable in the string is H the final syllable must agree with it. This was describable with the string-based propositional statement $\phi_{IH,FH} = \rtimes H \rightarrow H \ltimes$ in $\mathfrak{L}^P$(see p. 90).

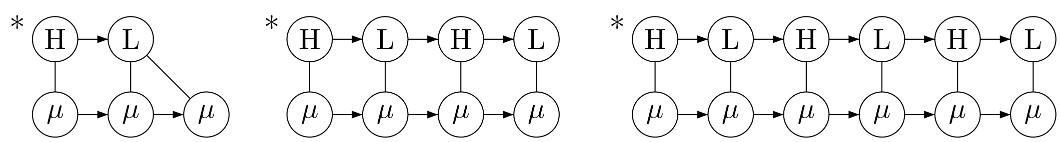
(6.45)

In $L(\phi_{IH,FH})$	Not in $L(\phi_{IH,FH})$
L, H, LL, HH, LLLL,	HL, HHL, HLLL, HLHL,
HLLH, HLHH,	HHHL, HLLLLLL, ...
HLLLLH,...	

In Chapter 3, §3.4.3 it was discussed how this language was not describable in $\mathfrak{L}^{NL}$. The intuition given in that section was that it is impossible, with local string constraints, to distinguish strings of the shape H^{n}LH, which belong to $L(\phi_{IH,FH})$, from strings of the shape H^{n}L, which do not. As the reader can probably guess, $\mathfrak{L}^{NL}_G$ constraints can easily distinguish these two types of strings (by banning graphs with a $\rtimes$HL$\ltimes$ melody). Instead, to show that $\mathfrak{L}^{NL}_G$ cannot describe $L(\phi_{IH,FH})$, we can focus on a different set of strings. This set of strings is of the shape HΓ^nL—the set of strings which begin with H, have n H or L symbols, and then end in an L. As any such string begins with an H and ends in an L, it does not belong to $L(\phi_{IH,FH})$. The following shows that constraints in $\mathfrak{L}^{NL}_G$ cannot isolate this set of strings from those which belong to $L(\phi_{IH,FH})$.[4]

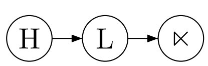
As contours are irrelevant, consider graphs in $APG(\Gamma)$ for $\Gamma = \{\mathrm{H}, \mathrm{L}\}$. Example graphs of *H$\Gamma^n$L strings are given below in (6.46).

(6.46) $^*g$(HLL) = $\quad$ $^*g$(HLHL) = $\quad$ $^*g$(HLHLHL) =



We cannot ban this set of graphs with a finite set of constraints in $\mathfrak{L}^{NL}_G$ without also banning some graphs corresponding to strings in $L(\phi_{IH,FH})$. For example, one subgraph shared by all of the graphs in (6.46) is that denoted by $\phi_{HL\ltimes}$ below.

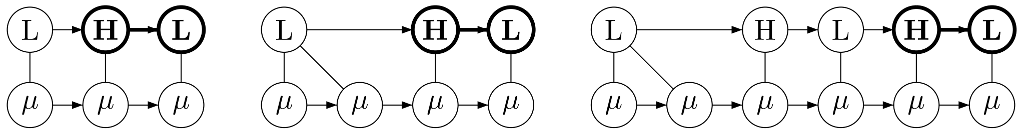
(6.47) $\phi_{HL\ltimes}$ =



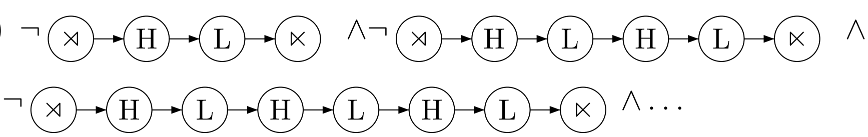
However, this subgraph is also shared by graphs corresponding to strings that belong in $L(\phi_{IH,FH})$—namely, strings that do not *start* with H (and thus do not have to end with H), but end in an HL melody:

[4] In Chapter 3, Footnote 3 (p. 93) mentions suffix substitution closure (Rogers and Pullum, 2011), an abstract characterization of $\mathfrak{L}^{NL}$ patterns over strings that allows one to prove whether or not a pattern is describable in $\mathfrak{L}^{NL}$. Currently, there is no such characterization for the graph patterns describable in $\mathfrak{L}^{NL}_G$. Defining suffix substitution closure for graphs (or perhaps just autosegmental graphs) is an interesting problem that will be left for future work, but it can draw on the ideas in the argument that follows in the main text.

(6.48) $g(\text{LHL}) =$ $\quad g(\text{LLHL}) =$ $\quad g(\text{LLHLHL}) =$



Trying to ban other subgraphs shared by all of the graphs in (6.46) runs into the same problem. For example, we cannot ban an initial H tone, because there are of course strings in $L(\phi_{IH,FH})$ which contain an initial H tone. The only possible way is to ban all of the melodies which begin with a H tone and end with a L tone. However, as indicated below, such a statement would need to be infinitely long, as there are in principle infinitely many melodies which start H and end L:

(6.49)


Again, as only finite statements belong to $\mathfrak{L}_G^{NL}$, the statement in (6.49) is not a valid statement in $\mathfrak{L}_G^{NL}$.

Intuitively, the reason why $\mathfrak{L}_G^{NL}$ cannot describe this pattern is because $L(\phi_{IH,HF})$ is a long-distance pattern *over the melody tier*. That is, any number of melody nodes may come between an initial H node and the final node, so there is no local way of requiring that the second node also be H. In contrast, $\mathfrak{L}^P$ can describe this pattern because it allows statements which require structures. Thus, while $\mathfrak{L}_G^{NL}$ is powerful enough to capture attested patterns which are beyond the range of $\mathfrak{L}^{NL}$ and $\mathfrak{L}_T^{NL}$, it is not able to mimic statements in $\mathfrak{L}^P$ which *require* structures.

Thus, $\mathfrak{L}_G^{NL}$ is powerful enough to describe attested patterns outside of the range of banned substructure constraints over strings, but not so powerful that it overgenerates unattested patterns predicted to exist by the more powerful string logic $\mathfrak{L}^P$. Thus, as a theory of tonal well-formedness, $\mathfrak{L}_G^{NL}$ is superior to these string logics, as it forms a tighter fit to the attested typology. As this is sufficient for the goals of this dissertation, a full exploration of the formal relationships between $\mathfrak{L}^{NL}_{G}$ and other kinds of string logics will be left for future research.

6.5 Conclusion

This chapter has synthesized the results of the past two chapters in order to approach the goal set out in Chapter 3, which was to apply concepts of phonological structure to the traditional string-based results of formal language theory in order to better describe patterns seen in tone languages. In this chapter, we have seen how the reanalysis of string symbols as graph primitives representing possible associations of the autosegments, and thus strings as concatenation of these primitives, allows us to describe long-distance tone patterns in Hirosaki Japanese, Wan Japanese, and unbounded tone plateauing, with local constraints. This was shown to be possible because the graph structures reveal relationships between symbols in a traditional string. As such, we have seen that grammars based on $\mathfrak{L}^{NL}_{G}$ match the attested tonal typology better than $\mathfrak{L}^{NL}$ or $\mathfrak{L}^{NL}_{T}$, but also do not overgenerate like $\mathfrak{L}^{P}$.

We have now seen how the entire range of tonal patterns originally discussed in Chapter 2 are fundamentally *local* in a well-defined way. As discussed in detail in the previous chapter, this local nature of the patterns is missed by derivational and optimization-based theories of tonal phonology. One potential criticism of $\mathfrak{L}^{NL}_{G}$ as a theory of tonal well-formedness is that it focuses on the surface structures, and thus says nothing about the transformations from underlying form to surface form from which they are derived. This criticism, however, is invalid. Regardless of how one chooses to explain the transformations, it remains a fact that the surface patterns are local, and thus *a strong theory of tonal transformations should focus on this fact.* The following chapter sketches out one way of developing such a theory.

Chapter 7

PHONOLOGICAL TRANSFORMATIONS AS GRAPH SETS

Building on the concepts of the preceding chapters, this chapter presents a method for representing phonological transformations as graphs and specifying these transformations through banned subgraph constraints in $\mathfrak{L}_G^{NL}$. This demonstrates that the surface locality of tonal generalizations established in the preceding two chapters can be directly incorporated into a theory of phonological transformations.

The main idea is to apply these constraints to *correspondence graphs*, which explicitly include edges representing correspondence between units in the input and units in the output. By specifying sets of these graphs with banned subgraph grammars, we can specify input/output relations corresponding to phonological transformations, be it over strings or over autosegmental structures. There are several main advantages to viewing transductions this way. One, concatenation provides for a restrictive theory of correspondence. Two, by applying the idea of locality established throughout this dissertation directly to phonological transformations, we can begin to develop a new way of specifying locality in phonological transformations. Most importantly, it allows the subgraph constraints already defined in the preceding chapters to specify autosegmental transformations, not unlike the role Markedness constraints play in Optimality Theory. However, it does this in a way that ensures the locality of the surface generalizations.

This method of specifying transformations is demonstrated through analyses of the patterns in Hirosaki Japanese, unbounded tone plateauing, Hausa, and Mende as transformations from underlying to surface autosegmental structures. The surface well-formedness constraints established in the previous two chapters for these patterns form the core of these analyses, thus preserving the OT insight that Markedness constraints drive phonological transformation. However, by applying the framework of the previous chapters, we also gain the insight that these transformations are fundamentally *local*. It should be stressed that this is an exploratory chapter, and there are some issues that deserve further work—one example is the particular representations necessary to capture the mappings for Hausa and Mende. However, it is still a not insubstantial result that these transformations can be shown to be local in the sense advocated for by this dissertation.

This chapter is structured as follows. §7.1 outlines shows how phonological transformations are *relations* between sets of input objects and output objects. §7.2 introduces the notion of specifying these relations through graphs and subgraph constraints. §7.3 then shows how relations between sets of APGs can also be specified using these constraints, using the example of Hirosaki Japanese. §7.4 further explores how these can be used to specify the processes in UTP, Mende and Hausa. Finally, §7.5 concludes, highlighting issues that require further investigation.

7.1 Phonological Transformations as Relations

7.1.1 Phonological formalisms specify relations

Any theory of phonological transformations pairs a set of input structures with a set of output structures. This is true for both rule-based theories based on the formalism from Chomsky and Halle (1968) or optimization-based theories, both parallel (Prince and Smolensky, 1993, 2004) and serial (McCarthy, 2000, 2010b). For example, consider the following simple phonological generalization.

(7.1) Obstruents are voiced intervocalically. Ex., /apa/ → [aba]

In a rule-based framework, this generalization might be captured as follows:

(7.2) [+sonorant] → [+voice] / V __ V

This rule singles out all intervocalic obstruents and ensures that they are output as [+voice].

In an OT framework, this generalization would be captured with a partial ranking of the following constraints:

(7.3) a. *VTV: Penalize sequences of a V, a voiceless obstruent, and another V. Assign one violation mark for every V[−sonorant, +voice]V sequence in the output.

b. MAX: Do not delete segments. Assign one violation for every segment in the input without a corresponding segment in the output.

c. IDENT(voice): Do not change voicing features. Assign one violation for every segment in the input whose corresponding output differs in its value for [±voice]

The constraint *VTV is a MARKEDNESS constraint militating against intervocalic voiced obstruents. By ranking the faithfulness constraint IDENT(voice), which penalizes changes in voicing, below *VTV, we can allow changes in the voicing to ensure that no outputs have VTV sequences. Crucially, IDENT(voice) must also be ranked below other faithfulness constraints, such as MAX, to ensure that a change in voicing is preferred over other repairs, such as deletion of the offending T. This is illustrated in the following tableau:

(7.4)

/apa/	*VTV	MAX	IDENT(voice)
apa	*!		
aa		*!	
☞ aba			*

Importantly, the generalization in (7.1) is specifically about intervocalic voicing, and is independent of other concerns, such as the length of words, or the well-formedness of vowel sequences, et cetera. This can be seen in the fact that both the rule and the OT tableau are agnostic to the particular inputs they are given. There is nothing about either the rule or the tableau that would reject an input like /aaa/—both would output an [aaa]. In rule-based theories, such inputs might be banned by morpheme structure constraints, and in OT, other constraints dealing with vowel sequence would likely change /aaa/ to something else. But, if we accept that phonological generalizations can exist as independent objects, we can ignore these additional constraints. (This issue was also touched on in Chapter 3, §3.2.)

We can then realize that the rule in (7.2) and the OT tableau in (7.4) represent the same map of inputs to outputs. For simplicity, let us only use /a/ to represent vowels, /b/ to represent voiced obstruents, and /p/ to represent voiceless obstruents. Both (7.2) and (7.4) admit the following set of input/output pairs. For reasons that will become clear in a moment, let us call this set R_{voice}.

(7.5) $R_{voice} = \{ \text{(pa, pa), (aaa, aaa), (apa, aba), (aba, aba), (appa, appa),}$
$\text{(aapaaapa, aabaaaba)}, \ldots \}$

Because it represents the set of pairs R_{voice}, the intervocalic voicing generalization in (7.1) is a *relation*, a notion briefly discussed in Chapter 3, §3.1. The rule in (7.2) and the OT tableau in (7.4) are thus two ways of describing the same relation. The rest of this chapter is concerned with a way of describing relations based in graphs. The following sections thus review the idea of a relation, then show how they can be represented as graphs.

7.1.2 Formal definition of relations

Recall from Chapter 3 that for two sets S and T, the *Cartesian product* of S and T, written $S \times T$, is the set of all possible pairs (x, y) where $x \in S$ and $y \in T$. (S and T need not be distinct sets.) A relation is some subset $R \subseteq S \times T$ of the Cartesian product of S and T.

Let $\Sigma = \{\text{a, b, p}\}$ be our input alphabet and $\Delta = \{\text{a, b, p}\}$ be our output alphabet. (In this example, Σ and Δ are the same but in general it will be useful to distinguish them). The Cartesian product of the sets Σ^* and Δ^* of strings over each alphabet is thus the following set:

(7.6) $\Sigma^* \times \Delta^* = \{ \text{(a, a), (a, b), (a, }\lambda), \ldots, \text{(aaa, aaa)}, \ldots,$
$\text{(apa, apa), (apa, aba)}, \ldots, \text{(apapa, b)}, \ldots \}$

That is, $\Sigma^* \times \Delta^*$ is the set resulting from pairing every string in Σ^* with a string in Δ^*. Returning to the set of pairs R_{voice} in (7.5), note that any pair in R_{voice} is also a pair in $\Sigma^* \times \Delta^*$ (but not vice versa). Thus $R \subset \Sigma^* \times \Delta^*$, and so R_{voice} is a relation between strings in Σ^* and Δ^*.

While they do it in different ways, the rule in (7.2) or the tableau (7.4) *specify* R_{voice} by singling out a specific subset of the pairs in $\Sigma^* \times \Delta^*$. More generally, this is the purpose of a grammar specifying a relation between sets S and T: to pick out a subset of the pairs in $S \times T$.

Like languages, relations also can be grouped into classes based on their computational properties. For example, the *regular relations* are those string relations specifiable by finite-state automata (Hopcroft et al., 2006). In terms of phonology, Chandlee (2014) and Chandlee and Heinz (to appear) define a subset of the regular relations called the *Strictly Local functions*, which extend the idea of locality over strings and formal langauges discussed in Chapter 3 to relations. They argue how they provide a strong computational characterization of local phonological processes as relations, and related work has shown how this characterization leads to an understanding of how they can be learned (Chandlee, 2014; Chandlee and Jardine, 2014; Chandlee et al., 2014; Jardine et al., 2014).

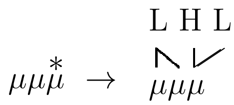
However, the regular relations and Strictly Local functions are defined in terms of finite-state automata over strings.[1] The focus of this dissertation has been on *graphs*, and it has shown that, at least for tone, graph-like structures are important for understanding phonological patterns. Of course, the idea of a relation is not restricted to strings, as in $S \times T$ the sets S and T can be comprised of any kind of object. Thus, autosegmental processes can be viewed as relations between graphs, and the aim of this chapter is to propose a strong, *local* characterization of phonological graph relations.

[1] Technically, these definitions are for *single-tape* automata. There exists work on implementing autosegmental phenomena with *multi*-tape finite state automata (Kay, 1987; Wiebe, 1992; Kornai, 1991, 1995). However, this work does not define any notion of locality for these machines, or how it would relate to locality over strings.

This is achieved not with automata (though that would likely be a fruitful avenue for future research) but with the logical grammars thus far established in the dissertation.

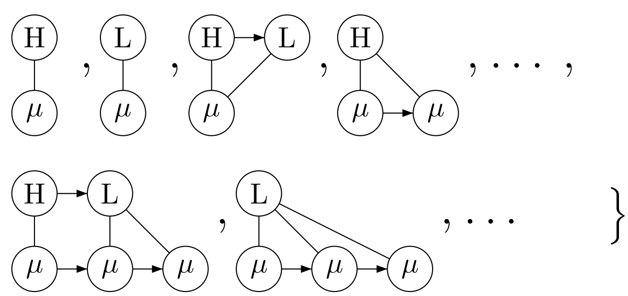
To give an example of an autosegmental process as a graph relation, it was discussed in Chapter 4 (p. 67), the surface pattern of Hirosaki Japanese was analyzed by Haraguchi (1977) as the realization of an underlying accented mora (or lack thereof). For example, an LLF string is the realization of an underlying final accented mora:

(7.7)



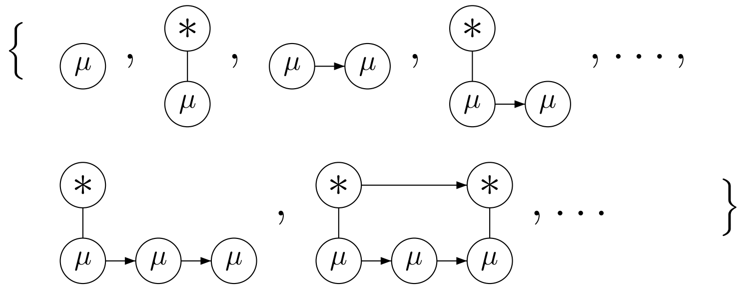
Transformations such as the mapping of tones to mora in Hirosaki Japanese are relations between autosegmental structures, and thus can be seen as subsets of the Cartesian product of two sets of graphs. As discussed in Chapter 6, the graphs representing the surface pattern of Hirosaki Japanese are a subset of $APG(\Gamma)$ where the set $\Gamma = \{\mathrm{H}, \mathrm{L}, \mathrm{F}\}$ of APG graph primitives is defined as in Chapter 6. To distinguish this from the set of input graphs $APG(\{\mu, *\})$ discussed below, let us refer to this set here as $APG(\{\mathrm{H}, \mathrm{L}, \mathrm{F}\})$. Again, any APG that can be constructed through the concatenation of graph primitives $g(\mathrm{H})$, $g(\mathrm{L})$, and $g(\mathrm{F})$ belongs to this set:

(7.8) $APG(\{\mathrm{H}, \mathrm{L}, \mathrm{F}\}) = \{$



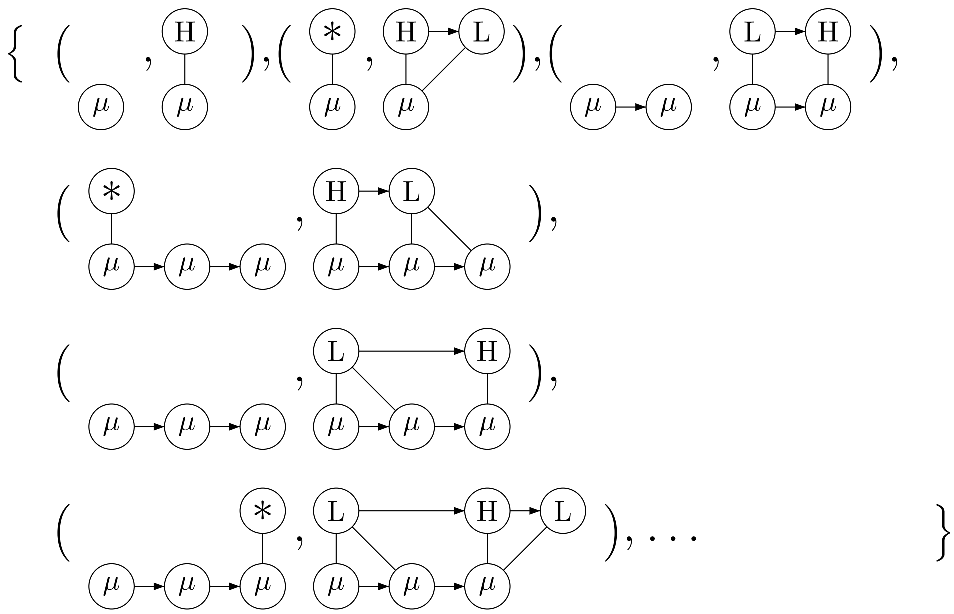
For representing the inputs, take the following set $APG(\{\mu, *\})$ of autosegmental graphs made up of unspecified morae and morae marked with a star. How this set can be generated will be discussed below in §7.3.

(7.9) $APG(\{\mu, *\}) =$



In Hirosaki Japanese, underlying forms have at most one accented mora, which is mapped to a single H tone, or if it is final a falling tone. Underlying forms with no starred morae get a final H. This mapping corresponds to the following subset R_{HJ} of $APG(\{\mu, *\}) \times APG(\{\mathrm{H}, \mathrm{L}, \mathrm{F}\})$:

(7.10) $R_{HJ} =$



Thus, in order to talk about autosegmental transformations, we need some way of specifying graph relations. This can be done using graphs representing input/output relations, as the remainder of the chapter will show. However, first, as strings are simpler models, this section first shows how string relations can be specified using graphs. §7.3 will then extend this idea to graphs.

7.2 String Relations as Graphs

This section outlines how to define relations using graphs, drawing inspiration from Potts and Pullum (2002), who show how input/output pairs of phonological structures can be represented as graph-like relational structures matching input units to their corresponding outputs. However, unlike Potts and Pullum (2002), this section defines a restrictive notion of correspondence using Chapter 4's notion of graph concatenation.

Also, whereas Potts and Pullum (2002)'s grammars were OT-based grammars evaluated through modal logic, this section shows how the idea of banned subgraph constraints developed in the preceding chapters of this dissertation can be fruitfully applied to specify phonological relations, both of strings and of autosegmental structures, in a local manner.

This section first introduces this notion using strings, and then the following section extends it to autosegmental structures.

7.2.1 Correspondence graphs

An important concept of studying relations as graphs comes from the idea in Optimality Theory that units in the input *correspond* to units in the output (Kager, 1999; McCarthy and Prince, 1995). For example, consider the following input/output pairs from the intervocalic voicing relation R_{voice} in (7.5):

(7.11) a. pa, pa

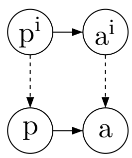
b. apa, aba

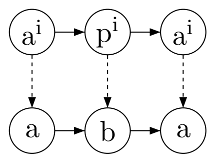
When we think about R_{voice} as representing a phonological transformation from an input underlying form to an output surface form, there is a sense in which, for example in (7.11b), the first /a/ corresponds to the first [a], the /p/ corresponds to the [b], and the second /a/ corresponds to the second [a]. The correspondences for (7.11) are schematized in (7.12).

(7.12) a.
$$\begin{tikzcd} \text{p} \arrow[d, dashed] & \text{a} \arrow[d, dashed] \\ \text{p} & \text{a} \end{tikzcd}$$
b.
$$\begin{tikzcd} \text{a} \arrow[d, dashed] & \text{p} \arrow[d, dashed] & \text{a} \arrow[d, dashed] \\ \text{a} & \text{b} & \text{a} \end{tikzcd}$$

Such representations can be studied as graphs much like those studied throughout the dissertation so far, with an additional type of arc to represent the correspondence relations. Constraints can then be defined over these graphs, an idea which has precedent in Potts and Pullum (2002). However, Potts and Pullum (2002) do not explicitly define GEN, or the OT version of the set of possible correspondences (as is usual in OT analyses). The following shows how a restrictive set of possible string-string correspondences can be explicitly defined with a concatenation operation like the one defined in Chapter 4.

Graphs representing the input/output pairs in (7.12) are depicted below in (7.13) with *precedence arcs* depicted as solid lines and *correspondence arcs* depicted as dotted lines. The labels on the input side have been given 'i' subscripts; this is to distinguish the two alphabets Σ and Δ, and will be explained further momentarily.

(7.13) a. 

b. 

To distinguish between arcs, we can add an *arc labeling function* ℓ_A to our definition of a graph. Thus, let us define *correspondence graphs* as graphs of the shape $\langle V, E, A, \ell, \ell_A \rangle$, where $\ell_A : A \to \{0, 1\}$ is a function labeling precedence arcs 0 and correspondence arcs 1.[2] As in the graph above, these labels will not be written on depictions of the graph but instead arcs labeled 0 (=precedence arcs) will be represented by solid arrows and arcs labeled 1 (=correspondence arcs) will be represented by dotted arrows.

Note that modulo the presence of correspondence arcs instead of association lines, the shape of the correspondence graph in (7.13) is very much like that of an APG. The nodes form two separate 'tiers', one whose labels are drawn from the alphabet Σ (marked here with the 'i' subscripts) and one whose labels Δ. Parallel to the tier

[2] The undirected edges E will come into play when we consider relational APGs.

alphabets of APGs, the precedence arcs in (7.13) only connect nodes of the same alphabet.

As such, we can straightforwardly build a graph like (7.13) through concatenation of graph primitives. Consider the following set Γ of graph primitives in (7.14). For the symbols in Γ, each symbol represents the input and its subscript the output.

(7.14) $\Gamma = \{\mathrm{a_a}, \mathrm{p_b}, \mathrm{p_p}, \mathrm{b_b}\}$, $g(\mathrm{a_a}) = \begin{tikzcd} \mathrm{a^i} \arrow[d, dotted] \\ \mathrm{a} \end{tikzcd}$, $g(\mathrm{p_p}) = \begin{tikzcd} \mathrm{p^i} \arrow[d, dotted] \\ \mathrm{p} \end{tikzcd}$, $g(\mathrm{p_b}) = \begin{tikzcd} \mathrm{p^i} \arrow[d, dotted] \\ \mathrm{b} \end{tikzcd}$, $g(\mathrm{b_b}) = \begin{tikzcd} \mathrm{b^i} \arrow[d, dotted] \\ \mathrm{b} \end{tikzcd}$

We can concatenate these primitives together with an operation almost identical to that defined in Chapter 4, treating $\{\Sigma, \Delta\}$ as a tier partition on the labeling alphabet $\Sigma \cup \Delta$. However, we of course do not want to 'merge' any nodes in these kinds of graph primitives. We can then consider a slightly modified graph concatenation operation $\odot$, which only draws precedence arcs between ends on each tier.[3] If we then use $\odot$ to extend the g function to strings in Γ^* as this was done in Chapter 4 with $\circ$, the graphs from (7.13) can be seen as the values for $g(\mathrm{p_p a_a})$ and $g(\mathrm{a_a p_b a_a})$, respectively:

(7.15) a. $g(\mathrm{p_p a_a}) = g(\mathrm{p_p}) \odot g(\mathrm{a_a}) = \begin{tikzcd} \mathrm{p^i} \arrow[d, dotted] \\ \mathrm{p} \end{tikzcd} \odot \begin{tikzcd} \mathrm{a^i} \arrow[d, dotted] \\ \mathrm{a} \end{tikzcd} = \begin{tikzcd} \mathrm{p^i} \arrow[r] \arrow[d, dotted] & \mathrm{a^i} \arrow[d, dotted] \\ \mathrm{p} \arrow[r] & \mathrm{a} \end{tikzcd}$

b. $g(\mathrm{a_a p_b a_a}) = (g(\mathrm{a_a}) \odot g(\mathrm{p_b})) \odot g(\mathrm{a_a}) =$

$$\left(\left(\begin{tikzcd} \mathrm{a^i} \arrow[d, dotted] \\ \mathrm{a} \end{tikzcd} \odot \begin{tikzcd} \mathrm{p^i} \arrow[d, dotted] \\ \mathrm{b} \end{tikzcd} \right) \odot \begin{tikzcd} \mathrm{a^i} \arrow[d, dotted] \\ \mathrm{a} \end{tikzcd} \right) = \begin{tikzcd} \mathrm{a^i} \arrow[r] \arrow[d, dotted] & \mathrm{p^i} \arrow[r] \arrow[d, dotted] & \mathrm{a^i} \arrow[d, dotted] \\ \mathrm{a} \arrow[r] & \mathrm{b} \arrow[r] & \mathrm{a} \end{tikzcd}$$

Let $CG(\Gamma) = \{g(w) | w \in \Gamma^*\}$; that is, the set of all correspondence graphs built from strings in Γ^*. The graphs in (7.13) are of course members of $CG(\Gamma)$; some other examples are given below.

[3] This small change to the concatenation operation, as with the other modifications to concatenation dicussed throughout this chapter, has no effect on its associativity.

(7.16) $g(\mathrm{a_a a_a}) = \begin{tikzcd} \mathrm{a^i} \arrow[r] \arrow[d, dashed] & \mathrm{a^i} \arrow[d, dashed] \\ \mathrm{a} \arrow[r] & \mathrm{a} \end{tikzcd}$ $g(\mathrm{p_b a_a}) = \begin{tikzcd} \mathrm{p^i} \arrow[r] \arrow[d, dashed] & \mathrm{a^i} \arrow[d, dashed] \\ \mathrm{b} \arrow[r] & \mathrm{a} \end{tikzcd}$ $g(\mathrm{a_a p_p a_a}) = \begin{tikzcd} \mathrm{a^i} \arrow[r] \arrow[d, dashed] & \mathrm{p^i} \arrow[r] \arrow[d, dashed] & \mathrm{a^i} \arrow[d, dashed] \\ \mathrm{a} \arrow[r] & \mathrm{p} \arrow[r] & \mathrm{a} \end{tikzcd}$

$g(\mathrm{a_a p_p p_p a_a}) = \begin{tikzcd} \mathrm{a^i} \arrow[r] \arrow[d, dashed] & \mathrm{p^i} \arrow[r] \arrow[d, dashed] & \mathrm{p^i} \arrow[r] \arrow[d, dashed] & \mathrm{a^i} \arrow[d, dashed] \\ \mathrm{a} \arrow[r] & \mathrm{p} \arrow[r] & \mathrm{p} \arrow[r] & \mathrm{a} \end{tikzcd}$ $g(\mathrm{a_a p_b p_p a_a}) = \begin{tikzcd} \mathrm{a^i} \arrow[r] \arrow[d, dashed] & \mathrm{p^i} \arrow[r] \arrow[d, dashed] & \mathrm{p^i} \arrow[r] \arrow[d, dashed] & \mathrm{a^i} \arrow[d, dashed] \\ \mathrm{a} \arrow[r] & \mathrm{b} \arrow[r] & \mathrm{p} \arrow[r] & \mathrm{a} \end{tikzcd}$

The set $CG(\Gamma)$ thus represents a subset of $\Sigma^* \times \Delta^*$ where string length is preserved (that is, for each pair (w, v) of strings, w and v are of the same length) and where each 'p' in the input can correspond to either a 'p' or 'b' in the output. To make this explicit, we can define the string relation represented by a set of correspondence graphs. Let *input*(G) and *output*(G) denote the string in Σ^* and the string in Δ^*, respectively, represented by the labeled nodes in a correspondence graph G.[4] For example, *input*$(g(\mathrm{a_a p_b a_a}))$ = apa, and *output*$(g(\mathrm{a_a p_b a_a}))$ = aba. The relation represented by a set CG of correspondence graphs is thus the set of pairs from each $G \in CG$:

Definition 25 (String relation of a set of correspondence graphs) *The string relation represented by a set CG of correspondence graphs is*

$$R(CG) = \{(w, v) | \exists G \in CG \text{ s.t. } \mathit{input}(G) = w \text{ and } \mathit{output}(G) = v\}$$

Thus, the relation represented by $CG(\Gamma)$ above is as follows:

(7.17) $$R(CG(\Gamma)) = \{ \ (\mathrm{aa}, \mathrm{aa}), (\mathrm{pa}, \mathrm{pa}), (\mathrm{pa}, \mathrm{ba}), (\mathrm{apa}, \mathrm{apa}), (\mathrm{apa}, \mathrm{aba}), (\mathrm{appa}, \mathrm{appa}), (\mathrm{appa}, \mathrm{abpa}), \ldots \ \}$$

---

[4] Explicitly, *input*(G) is the string obtained by concatenating the labels on the nodes of the Σ tier of the graph in order of their arcs:

$$\begin{aligned} \mathit{input}(G) = \ & \sigma_0\sigma_1 \ldots \sigma_n \in \Sigma^* \text{ such that} \\ & \quad \exists v_0, v_1, ..., v_n \in V \text{ such that } \forall v_i, \ell(v_i) = \sigma_i \text{ and} \\ & \qquad \forall v_j, v_{j+1}, (v_j, v_{j+1}) \in A \end{aligned}$$

The definition for *output*(G) is nearly identical, except the resulting string is $\delta_0\delta_1 ... \delta_n \in \Delta^*$ extracted from nodes in G whose labels are in Δ.

For any relation defined using a set of correspondence graphs built out of a finite set of graph primitives primitives, the string pairs in that relation will be *restricted* based on the input/output correspondences defined by the graph primitives. For example, a pair like (apaab, baapa), which is a member of $\Sigma^* \times \Delta^*$, is *not* a member of $R(CG(\Gamma))$, because it is impossible to define, with concatenation of graph primitives, a correspondence such that the output is the reverse of the input. (This is of course possible in GEN under correspondence; for an example, see Potts and Pullum (2002), p. 374, fn. 11.) As such pairs are unlikely to belong to any *phonological* transformation, this is a welcome result in trying to define relations which describe such transformations.[5]

However, $R(CG(\Gamma))$ is still not equal to the intervocalic voicing relation R_{voice}. In a way, $CG(\Gamma)$ is like OT's GEN: it simply lists the set of possible input/output correspondences.[6] The relation $R(CG(\Gamma))$ contains pairs in which intervocalic 'p's correspond to a 'p' instead of the voiced 'b', such as (apa, apa), as well as those in which non-intervocalic 'p's in the input corresponding to voiced 'b's in the output, such as (pa, ba). These pairs do not belong to R_{voice}, so we need to be able to restrict relations further in order to describe R_{voice}. The following shows how this can be done by describing a subset of $CG(\Gamma)$ with banned subgraph constraints, in the exact same way previous chapters restricted autosegmental graphs with such constraints. This is a somewhat similar approach to Koskenniemi (1983)'s two-level constraints for phonology and morphology, which also can specify phonological processes through constraints on relations, but distinct in that Koskenniemi's framework was based in automata theory. By basing the constraints in graph theory, the methodology advanced

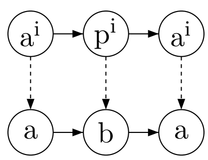
[5] Metathesis is attested, although all synchronic cases appear to be *bounded* (Chandlee et al., 2012). Bounded metathesis can be added to the set of correspondence graphs with primitives representing two (or more) input nodes corresponding to output nodes in a different linear order. A full analysis of metathesis in this framework will be left to future work.

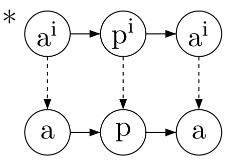
[6] Of course, it is a very simple GEN. We can consider more correspondences by enriching Γ, a point which will be returned to later.

in the preceding chapters of this dissertation can be extended to both strings and autosegmental representations, as shall be shown momentarily.

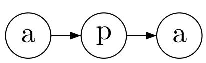
7.2.2 Defining string relations with banned subgraph grammars

The graph set $CG(\Gamma)$ contains all of the string correspondences in (7.5), but as noted above, contains some string correspondences not in R_{voice}. The two graphs below illustrates this. The graph (7.18a) represents the voicing of an intervocalic 'p'. The graph in (7.18b) is similar, but the intervocalic 'p' surfaces as voiceless. Following the notation used throughout the dissertation, this graph is marked with a star to indicate that we wish to exclude it.

(7.18) a. $g(\mathrm{a_a p_b a_a}) =$ 

b. $*g(\mathrm{a_a p_p a_a}) = *$ 

As $CG(\Gamma)$ is a graph set, there exists a language $\mathfrak{L}_G^{NL}$ of banned subgraph constraints defined exactly as in Chapter 5.[7] For example, the following is a valid subgraph literal in $\mathfrak{L}_G^{NL}$ as defined over $CG(\Gamma)$.

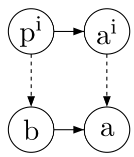
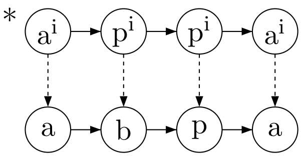
(7.19) $\phi_{apa} =$ 

This subgraph represents an 'apa' sequence in the output (note the lack of 'i' subscripts, indicating that this is a constraint on the output side of the graph). The negation of this literal, $\neg\phi_{apa}$, is thus only satisfied by graphs which do not contain an 'apa' sequence in the output. This thus distinguishes (7.18a), which satisfies $\neg\phi_{apa}$, from (7.18b), which contains ϕ_{apa} as a subgraph and thus does not satisfy $\neg\phi_{apa}$. In general, for all the graphs in $CG(\Gamma)$ with an underlying '$\mathrm{a^i p^i a^i}$' sequence which satisfy $\neg\phi_{apa}$, this sequence must correspond to an 'aba' sequence in the output. The constraint $\neg\phi_{apa}$ thus functions exactly as the MARKEDNESS constraint *VTV from

[7] Technically, there is one difference, which is that for correspondence graphs, the notion of subgraph needs to be updated to respect labeling of the arcs.

(7.3a), as it motivates the change from an underlying intervocalic /p/ to a surface [b] by penalizing any ‘apa’ sequence in an output.[8]

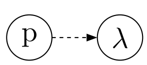
However, $\neg\phi_{apa}$ is not the only constraint we need to consider in order to fully capture the intervocalic voicing relation. The reason for this is that, as $g(\mathrm{p_b})$ belongs to our set of graph primitives, $CG(\Gamma)$ includes graphs in which an input ‘p’ changes to an output ‘b’ when it is *not* intervocalic, such as in the graphs below.

(7.20) a. $*g(\mathrm{p_b a_a}) = *$  b. $*g(\mathrm{a_a p_b p_p a_a}) = *$ 

In order to exclude graphs such as these, it is necessary to also specify when a ‘p’ in the input does *not* surface as a ‘b’ in the output. Any correspondence graph in $CG(\Gamma)$ in which a ‘p’ corresponds with a *non*-intervocalic output ‘b’ will contain one of the subgraphs in (7.21a). Examples are given in (7.21b), with the particular subgraph given in parentheses.

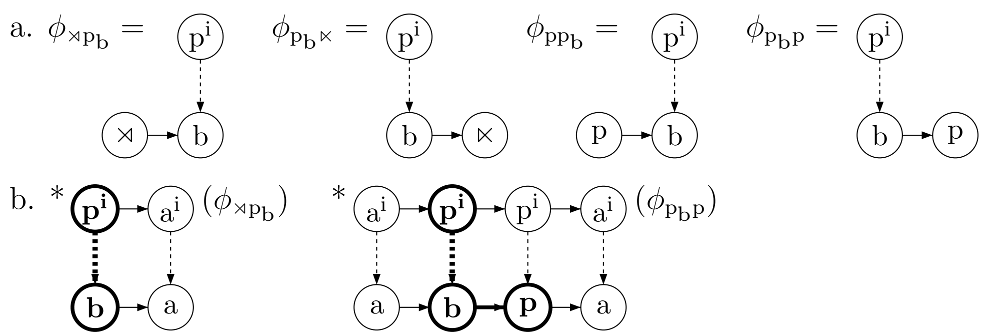
---

[8] In this simple example, the change from ‘p’ to ‘b’ is the only one available in order to satisfy ϕ_{apa}, as dictated by the primitives in Γ (for example, there is no primitive relating an input ‘p’ to an output ‘a’). Other repairs may be considered by enriching Γ to include other correspondences, such as this one from a ‘p’ to a special label representing the empty string, λ:

(i) 

This primitive can be interpreted as deleting the input ‘p’. (Some changes in the definition of *output*(G) would be required to correctly interpret λ as an output symbol.) This repair can be avoided in one of two ways. One, as in the main text, we can simply exclude such a primitive from Γ. Two, we could include (i) in the primitives, but then in the grammar for the intervocalic voicing pattern include a statement banning (i) as a substructure. This point is returned to at the end of this section, but the point to remember is that by abstracting away from this possible repair, the discussion in the main text is not ignoring any possibilities that cannot be described by banned subgraph constraints.

(7.21)



Negating the subgraph literals in (7.21a) will thus exclude any graphs in which an input ‘p’ corresponds to a non-intervocalic output ‘b’. This specification of environments in which a change does *not* occur is quite unlike rewrite-rules or OT constraints. However, this may look, as there may be other environments in which a ‘p’ should correspond to an output ‘b’ (such as if there was also post-nasal voicing). The reader may also have noticed that the number of constraints necessary will likely increase with any additions to the alphabet of graph primitives. However, this issue can be avoided with other representational schemes based on feature systems or, as will be shown in a moment, autosegments.

The correct set of graph correspondences for intervocalic voicing can thus be specified with the following statement $\phi_{\mathrm{ap_ba}}$ in $\mathfrak{L}_G^{NL}$:

(7.22) $\phi_{\mathrm{ap_ba}} = \neg\phi_{apa} \wedge \neg\phi_{\rtimes \mathrm{p_b}} \wedge \neg\phi_{\mathrm{p_b}\ltimes} \wedge \neg\phi_{\mathrm{pp_b}} \wedge \neg\phi_{\mathrm{p_bp}}$

Let $CG(\phi_{\mathrm{ap_ba}})$ be the set of graphs in $CG(\Gamma)$ which satisfy $\phi_{\mathrm{ap_ba}}$. The reader can confirm that $R(CG(\phi_{\mathrm{ap_ba}})) = R_{voice}$; that is, $\phi_{\mathrm{ap_ba}}$ specifies the string-to-string relation representing the intervocalic voicing process.

We have seen how the idea of local constraints over graphs can be extended to describe phonological relations. $CG(\Gamma)$ serves as a kind of GEN, providing the set of possible correspondences. A local grammar over these correspondences can then specify a particular relation. The Γ used in this example was quite simple, however as seen in Footnote 8, individual graph primitives can be banned with subgraph constraints, and so the presence or absence of particular primitives has no effect on the locality of a process. However, the nature of the alphabet Γ is an interesting question. Interpreted phonologically, this question asks what the set of possible correspondences in human phonology are. This question will be considered again in the following section.

Finally, given that decisions over well-formedness for statements in $\mathfrak{L}_G^{NL}$ is made *locally*, it is likely there is some relationship between the string relations which can be defined this way and those defined in Chandlee (2014), inter alia. From that work, we also know there are phonological processes which are *not* local over strings, and these are likely not describable by the $\mathfrak{L}_G^{NL}$ constraints described here. However, the purpose of this section is not to develop a full theory of phonological string relations over graphs, and so an investigation of these issues shall be left to future work. However, the following section will show that, when extended to APGs, the idea of banned subgraph constraints over correspondence graphs *can* capture the long-distance dependencies described in Chapter 6 with banned APG subgraph constraints.

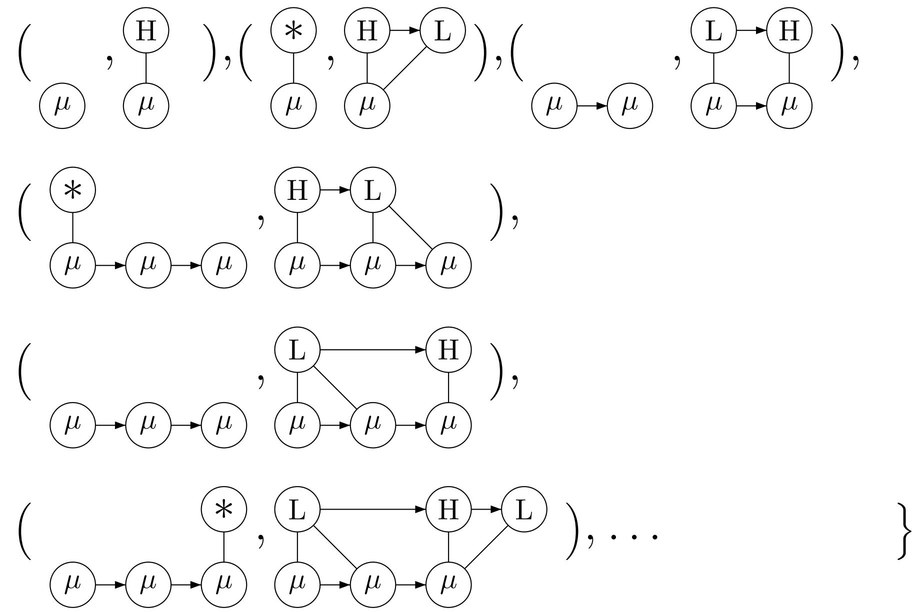
7.3 Defining Graph Relations with Banned Subgraph Grammars

We can now turn to the main focus of this chapter, which is beginning a theory of autosegmental transformations with banned subgraph constraints over correspondence graphs. That is the purpose of this section, which illustrates the concepts using the transformation of Hirosaki Japanese discussed earlier in the section.

7.3.1 Correspondence graphs for graph relations

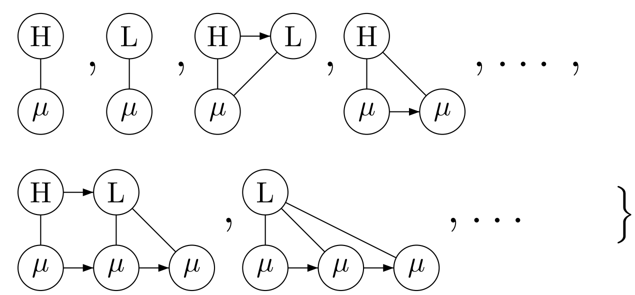
As an example, recall the Hirosaki Japanese relation R_{HJ}, in which words with no underlying accent surface with a final H tone and words with an underlying accent surface with a H tone on the accented mora (or a F if it is the final mora). Originally given in (7.10), R_{HJ} is repeated below in (7.23).

(7.23) $R_{HJ} = \{$

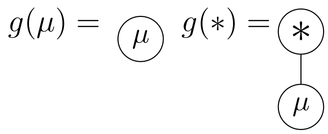


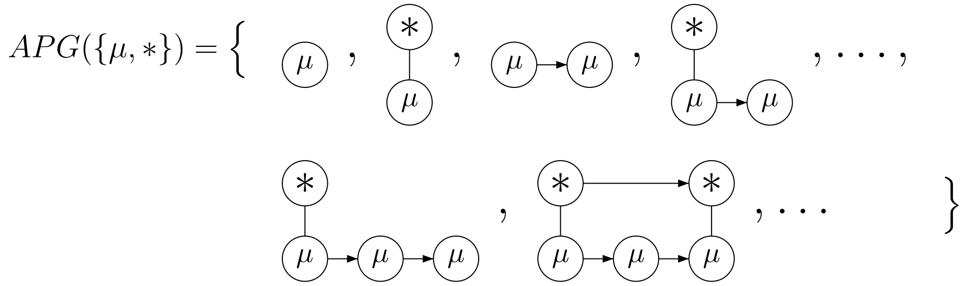
The relation R_{HJ} is a subset of the Cartesian product of two graph sets, $APG(\{\mu, *\}) \times APG(\{\mathrm{H}, \mathrm{L}, \mathrm{F}\})$. As noted above, $APG(\{\mathrm{H}, \mathrm{L}, \mathrm{F}\})$ is simply the set of APGs used in Chapter 6 to describe the surface pattern of Hirosaki Japanese. It was constructed using the graph primitives $\{\mathrm{H}, \mathrm{L}, \mathrm{F}\}$ as introduced in Chapter 6 through the concatenation operation defined in Chapter 4. This set is repeated below in (7.24).

(7.24) $APG(\{\mathrm{H}, \mathrm{L}, \mathrm{F}\}) = \{$



The set of graphs $APG(\{\mu, *\})$ is a bit different, as it is meant to represent underlying forms. As we shall see in this and the following section, we do not always want graph sets representing underlying forms to respect the OCP, and it will be useful to treat $APG(\{\mu, *\})$ this way too. Let $\{\mu, *\}$ be a set of APG primitives representing an unspecified mora and an accented mora, as defined below in (7.25). We can then define $APG(\{\mu, *\})$ as the set of graphs built out of concatenating $\{\mu, *\}$ with $\odot$, which again is the concatenation operation that only 'bridges' nodes by drawing arcs between them. This thus obtains the set of graphs below in (7.52), repeated from (7.9).

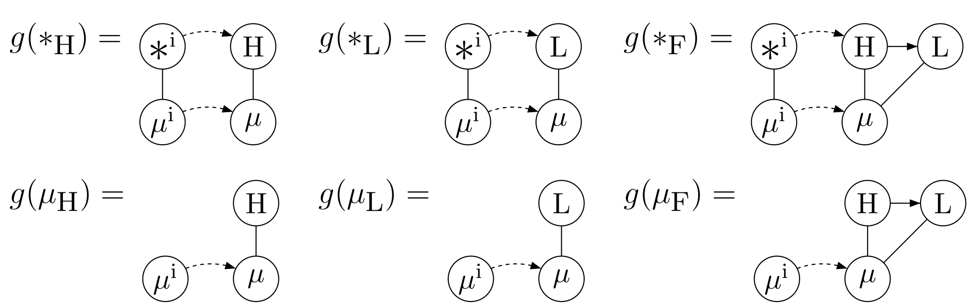
(7.25) 

(7.26) 

Again, our target relation R_{HJ} is a subset of $APG(\{\mu, *\}) \times APG(\{\mathrm{H}, \mathrm{L}, \mathrm{F}\})$, the product of the sets in (7.24) and (7.52). However, as $APG(\{\mu, *\}) \times APG(\{\mathrm{H}, \mathrm{L}, \mathrm{F}\})$ pairs *every* graph from $APG(\{\mu, *\})$ with *every* graph from $APG(\{\mu, *\})$, describing R_{HJ} requires excluding graph pairs from $APG(\{\mu, *\}) \times APG(\{\mathrm{H}, \mathrm{L}, \mathrm{F}\})$.

As above with string relations, the first step in doing this is to set up correspondence graph primitives which relate the primitives $\{\mu, *\}$ with primitives in $\{\mathrm{H}, \mathrm{L}, \mathrm{F}\}$. We can then use these as a set of correspondence graphs which represent pairs of graphs in $APG(\{\mu, *\}) \times APG(\{\mathrm{H}, \mathrm{L}, \mathrm{F}\})$. Importantly, we do not simply want to recapitulate string-based correspondence graphs as in the previous section. Patterns such as Hirosaki Japanese were shown in Chapters 3 and 6 to require graph structures to be described locally. Thus, in order to take advantage of this, it is necessary that the correspondence graph primitives preserve this graph structure.

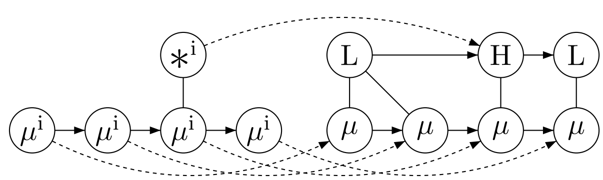
Given the graph primitives $\{\mu, *\}$ and $\{\mathrm{H}, \mathrm{L}, \mathrm{F}\}$, we can enumerate the possible correspondences as in the alphabet of correspondence graph primitives $\Gamma = \{*_\mathrm{H}, *_\mathrm{L}, *_\mathrm{F}, \mu_\mathrm{H}, \mu_\mathrm{L}, \mu_\mathrm{F}\}$ as defined below.

(7.27) 

For example, $g(*_{\mathrm{H}})$ represents an accented mora in the input corresponding to a H-toned mora in the output, whereas $g(\mu_{\mathrm{H}})$ represents an unaccented mora in the input corresponding to a H-toned mora in the output. Again, the labels on the input side of the graph have been marked with a subscript ‘i’.

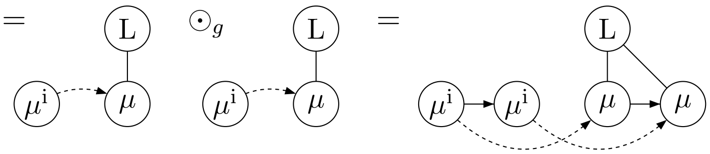
We can construct correspondence graphs out of these primitives, however it requires a slight modification of our $\odot$ operation. The graphs in $APG(\{\mu, *\})$ are built using $\odot$, which has no merging, but the graphs in $APG(\{\mathrm{H}, \mathrm{L}, \mathrm{F}\})$ are built using $\circ$, which merges nodes with like labels on the melody tier. The set of correspndence graphs thus requires a slight modification of $\odot$ such that it respects the differing concatenation paradigms for each side of the correspondence graph primitives. This is not technically difficult, however—we simply treat the nodes labeled with $\{\mathrm{H}, \mathrm{L}\}$ on the output side as the only true melody tier nodes, allowing them to be merged while the nodes on all other tiers have arcs drawn between them. Following the preceding definition of correspondence graphs for strings, let Σ refer to the label alphabet for the input side of a correspondence graph (here, corresponding to a graph in $APG(\{\mu, *\})$) and Δ be the label alphabet for the output side of a correspondence graph (i.e., corresponding to a graph in $APG(\{\mathrm{H}, \mathrm{L}, \mathrm{F}\})$). Then let $\Sigma_m = \{*^{\mathrm{i}}\}$ and $\Sigma_t = \{\mu^{\mathrm{i}}\}$ be the tier partitions on Σ for the input side of the graph, and $\Delta_m = \{\mathrm{H}, \mathrm{L}\}$ and $\Delta_t = \{\mu\}$ be the tier partitions for the output. Then define $\odot_g$ as a concatenation operation which merges end nodes with like labels in Δ_m but only draws arcs between end nodes on all of the other tiers. Let $CG(\Gamma)$ then refer to the set of correspondence graphs built by concatenating the graph primitives in Γ as defined in (7.27). An example graph in $CG(\Gamma)$, $g(\mu_{\mathrm{L}}\mu_{\mathrm{L}}*_{\mathrm{H}}\mu_{\mathrm{L}})$ is given below.

(7.28) $g(\mu_{\mathrm{L}}\mu_{\mathrm{L}}*_{\mathrm{H}}\mu_{\mathrm{L}}) =$



As these graphs can be visually complex, let us sequentially look at the concatenation of the primitives in $g(\mu_{\mathrm{L}}\mu_{\mathrm{L}}*_{\mathrm{H}}\mu_{\mathrm{L}})$. First, let us look at the concatenation of the first two primitives, $\mu_{\mathrm{L}}\mu_{\mathrm{L}}$. The graph $g(\mu_{\mathrm{L}}\mu_{\mathrm{L}})$ is given below.

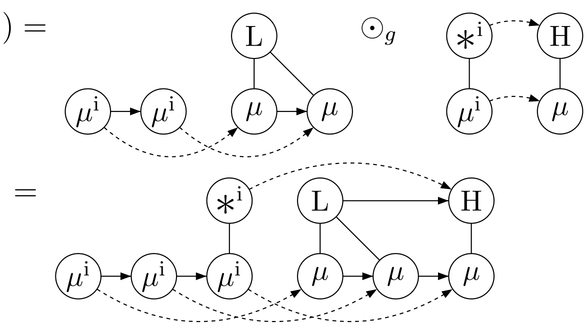
(7.29) $g(\mu_\text{L}\mu_\text{L}) =$



The concatenation above can be broken down as follows. Concatenation draws a precedence arc between the μ^i node in the first $g(\mu_\text{L})$ graph with the μ^i node in the second. Similarly, it draws an arc between the two μ nodes. (Remember that no new *correspondence* arcs are drawn, but they may be depicted differently for visual clarity.) Because the two L nodes are end nodes whose labels are in Δ_m, concatenation merges them. The result is a graph in which two unspecified mora nodes in the input correspond to two nodes associated with an L node in the output.

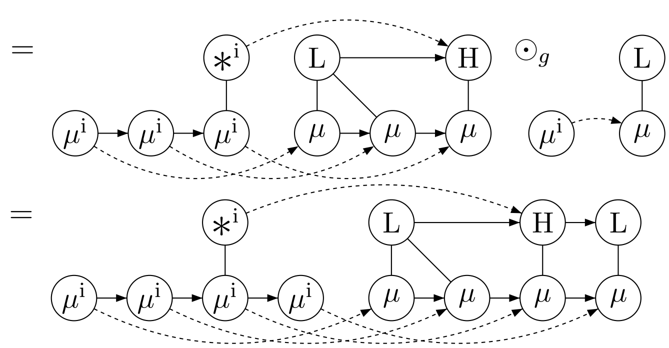
Let us add to this $g(*_\text{H})$.

(7.30) $g(\mu_\text{L}\mu_\text{L}) \odot_g g(*_\text{H}) =$



Here, concatenation draws a precedence arc from the last μ^i node of $g(\mu_\text{L}\mu_\text{L})$ to the first μ^i node of $g(*_\text{H})$. Likewise for the μ nodes on the output side of the graph. Similarly, the H is added to the end of the Δ_m tier on the output side of the graph. Finally, adding one further $g(\mu_\text{L})$ completes the graph in (7.28):

(7.31) $g(\mu_\text{L}\mu_\text{L}*_\text{H}) \odot_g g(\mu_\text{L}) =$

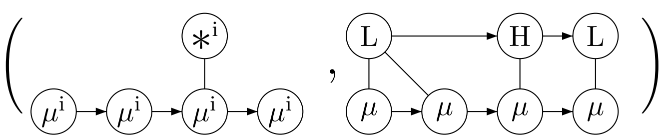


Like we saw with the string transformations in the previous section, this $CG(\Gamma)$ restricts us to graph pairs which can be built out of the primitives in Γ. Parallell to Definition 25, Definition 26 derives a graph relation from a set of correspondence graphs. In the following, $input(G)$ and $output(G)$ are functions returning the input and output side, respectively, of a correspondence graph G.[9]

Definition 26 (Graph relation of a set of correspondence graphs) *The graph relation represented by a set CG of correspondence graphs is*

$$R(CG) = \{(G_1, G_2) | \exists G \in CG \text{ s.t. } input(G) = G_1 \text{ and } output(G) = G_2\}$$

The valid transformations in Hirosaki Japanese, R_{HJ}, can then be seen as represented by a subset of the graphs in $CG(\Gamma)$. For example, in $g(\mu_{\mathrm{L}}\mu_{\mathrm{L}}*_{\mathrm{H}}\mu_{\mathrm{L}})$ above an underlying star is realized as a H tone, whereas the other, underspecified mora are associated to default L tones. Thus, the following pair is in $R(CG(\Gamma))$:

(7.32)


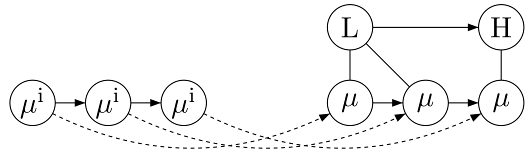
However, there are also some graph pairs in $CG(\Gamma)$ which do not represent valid transformations in Hirosaki Japanese. For reference, (7.33) gives more examples of graphs in $CG(\Gamma)$. Graphs in (7.33a) represent pairs that we will wish to keep to describe R_{HJ}, whereas graphs in (7.33b) are ones we will want to exclude.

[9] To be explicit, $input(G)$ is the *node-induced subgraph* of G for the set of nodes in G whose label is in Σ. A node-induced subgraph of G for a set X of nodes is the graph generated by taking all of the nodes in X and all of the edges and arcs in G whose endpoints are in X. Formally, for a correspondence graph $G = \langle V, E, A, \ell, \ell_A \rangle$,

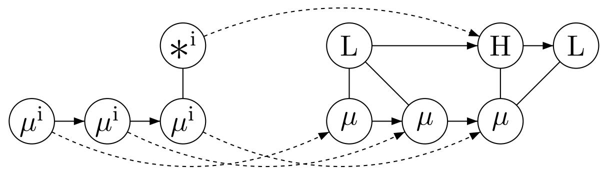
$$input(G) = G_i \text{ such that } \begin{array}{l} V_i = \{v \in G | \ell(v) \in \Sigma\} \\ E_i = \{\{v, w\} \in E | v, w \in V_i\} \\ A_i = \{(v, w) \in A | v, w \in A_i\} \\ \forall v \in V_i, \ell_i(v) = \ell(v) \end{array}$$

Note that G_i is not a correspondence graph since there is no ℓ_{A_i} defined. The function $output(G)$ is defined in an identical fashion, except with respect to Δ.

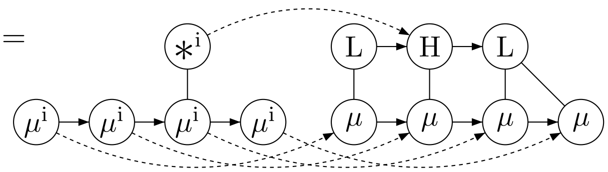
(7.33) a. $g(\mu_{\mathrm{L}}\mu_{\mathrm{L}}\mu_{\mathrm{H}})$=



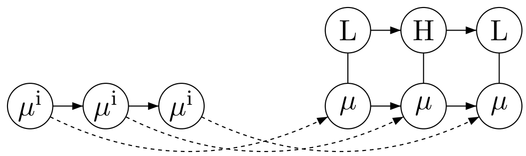
$g(\mu_{\mathrm{L}}\mu_{\mathrm{L}}{*}_{\mathrm{F}})$=



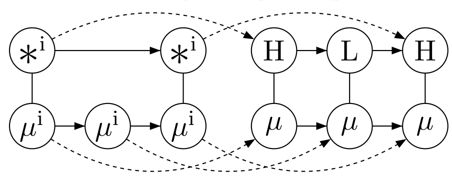
$g(\mu_{\mathrm{L}}{*}_{\mathrm{H}}\mu_{\mathrm{L}}\mu_{\mathrm{L}})$=



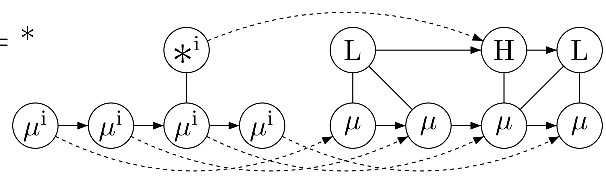
b. $^{*}g(\mu_{\mathrm{L}}\mu_{\mathrm{H}}\mu_{\mathrm{L}})$= *



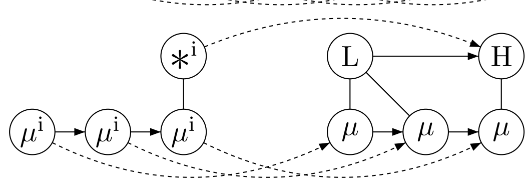
$^{*}g({*}_{\mathrm{H}}\mu_{\mathrm{L}}{*}_{\mathrm{H}})$ = *



$^{*}g(\mu_{\mathrm{L}}\mu_{\mathrm{L}}{*}_{\mathrm{F}}\mu_{\mathrm{L}})$ = *



$^{*}g(\mu_{\mathrm{L}}\mu_{\mathrm{L}}{*}_{\mathrm{H}})$= *

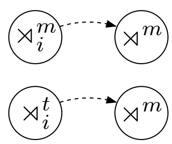


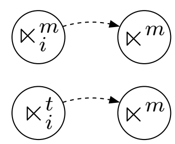
Now that we have defined a set of correspondence graphs, we can use constraints in $\mathfrak{L}_G^{NL}$ to define the language-specific patterning of Hirosaki Japanese.

7.3.2 Specifying graph relations with banned subgraph constraints

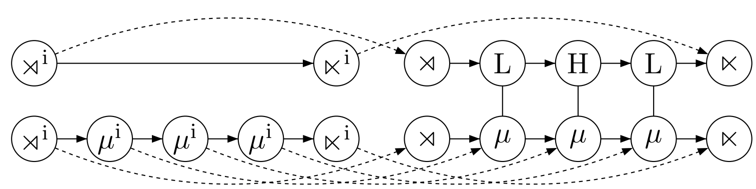
The following shows how a relation between APGs can be specified using banned subgraph constraints over a set of correspondence graphs. First, however, it is necessary to be explicit about the correspondence graph values of the word boundary symbols $\rtimes$ and $\ltimes$, as they will be used in some of the constraints throughout the remainder of this chapter. Recall that for a graph $g(w)$, the definition of its satisfaction of a statement in $\mathfrak{L}_G^{NL}$ is based on $g(\rtimes w \ltimes)$, and subgraph constraints may include portions of $g(\rtimes)$ and $g(\ltimes)$.

The following discussion will assume that for correspondence graphs representing relations between APGs, the graph primitives $g(\rtimes)$ and $g(\ltimes)$ are as below in (7.34). For now, the melody tier border node labels are marked with a superscript m, and the timing tier border node labels are marked with a superscript t. As they will be clear from the context, these marks will be omitted later on.

(7.34) $g(\rtimes) =$ 

$g(\ltimes) =$ 

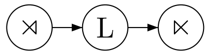
Note that there are correspondence arcs between the input and output word boundary symbols; this assumption will be crucial for some of the graph constraints used below. Also, as the input graphs are built out of primitives which do not necessarily contain melody tier nodes, it is important to also note the possibility of empty tiers, such as in the graph $g(\mu_{\mathrm{L}}\mu_{\mathrm{H}}\mu_{\mathrm{L}})$. When delimited by the word boundary symbols, such a graph will appear as in (7.35), with a precedence arc between the $\rtimes$ and $\ltimes$ nodes on the melody tier on the input side of the graph.

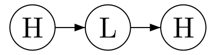
(7.35) $g(\rtimes\mu_{\mathrm{L}}\mu_{\mathrm{H}}\mu_{\mathrm{L}}\ltimes)=$ 

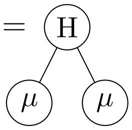
As discussed below, an input melody tier such as in (7.35) can identify input APGs in which no TBU nodes are associated to any melody tier node.

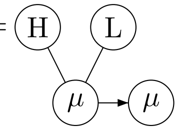
With the details of the tier boundaries made explicit, it is now possible to discuss the constraints which can ultimately specify the relation R_{HJ}. First, a great deal of the work can be done by the constraints on the surface graph pattern outlined in Chapter 6, §6.2.2. These constraints are repeated in (7.36), and their relevant subgraph literals are recapitulated below in (7.37).

(7.36) $\neg\phi_{\rtimes \mathrm{L} \ltimes} \wedge \neg\phi_{HLH} \wedge \neg\phi_{H^2} \wedge \neg\phi_{NF\text{-}Cont}$

(7.37) a. $\phi_{\rtimes L \ltimes} =$ 

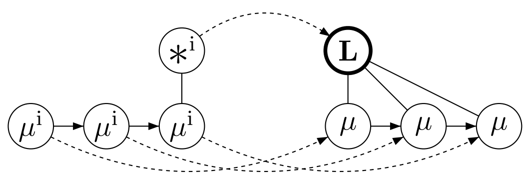
b. $\phi_{HLH} =$ 

c. $\phi_{H^2} =$ 

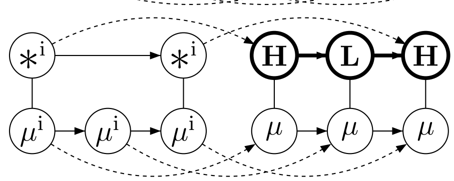
d. $\phi_{NF\text{-}Cont} =$ 

If we interpret the labels of the subgraphs in (7.37) as belonging to the alphabet in the output side of the graph (that is, $\Delta_m \cup \Delta_t$), then the constraints in (7.36) ban graphs in $CG(\Gamma)$. Examples of excluded graphs are given below in (7.38a) through (7.38d), corresponding to the banned subgraphs in (7.37a) through (7.37d). Specifically, $\neg\phi_{\rtimes L \ltimes}$ bans any correspondence graph without a H tone on the melody tier in the output graph (such as when an underlying star node corresponds to a surface L node, as in (7.38a) below); $\neg\phi_{HLH}$ bans any graph with more than one H node in the output (such as in the case of two star nodes corresponding to H nodes, as in (7.38b) below); $\neg\phi_{H^2}$ bans any graph with a multiply associated H node in the output (as in (7.38c) below); and $\neg\phi_{NF\text{-}Cont}$ bans any nonfinal contours (such as in (7.38d) below). As per usual, the examples below in (7.38) have the offending subgraphs highlighted in bold.

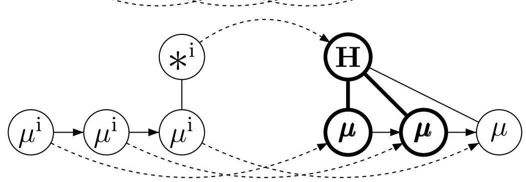
(7.38) a. $*g(\mu_\mathrm{L}\mu_\mathrm{L}*_\mathrm{L}) = *$



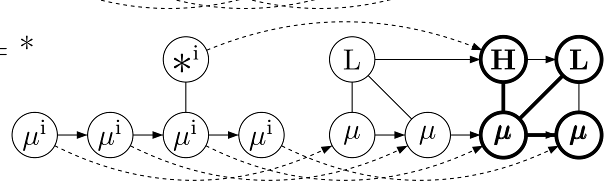
b. $*g(*_\mathrm{H}\mu_\mathrm{L}*_\mathrm{H}) = *$



c. $*g(\mu_\mathrm{H}\mu_\mathrm{H}*_\mathrm{H}) = *$

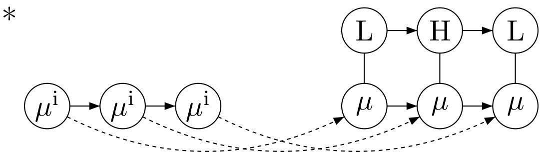


d. $*g(\mu_\mathrm{L}\mu_\mathrm{L}*_\mathrm{F}\mu_\mathrm{L}) = *$



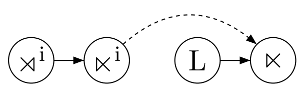
By specifying the valid set of output graphs, the constraints in (7.36) go a long way in specifying the subset of $CG(\Gamma)$ that represents the the Hirosaki Japanese APG relation R_{HJ}. However, there do need to be some constraints on correspondences. For example, recall that in Hirosaki Japanese unaccented words have a final H tone. In APG terms, this can be restated as saying that input graphs with an empty melody tier cannot correspond to a graph whose melody tier ends in L:

(7.39) $^*g(\mu_{\mathrm{L}}\mu_{\mathrm{H}}\mu_{\mathrm{L}}) = {}^*$

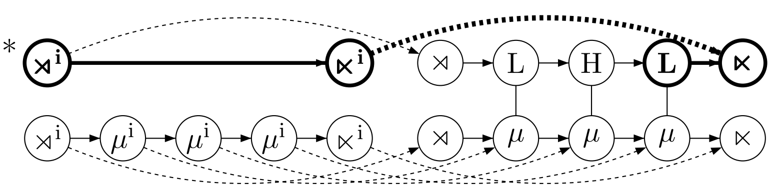


All such graphs can be specified with the subgraph in $\phi_{No^*\text{-}L\ltimes}$, which specifies an empty melody tier on the input and a L-final melody tier on the output. As an example, the offending graph in (7.39) is given with its boundary symbols explicitly added for clarity in (7.40b).

(7.40) a. $\phi_{No^*\text{-}L\ltimes} =$

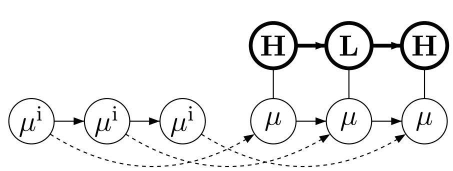


b. $^*g(\rtimes\mu_{\mathrm{L}}\mu_{\mathrm{H}}\mu_{\mathrm{L}}\ltimes) = {}^*$



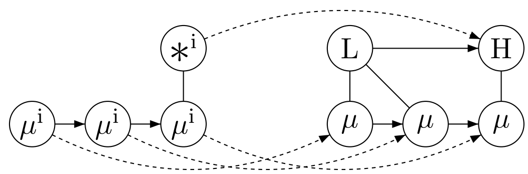
The reader can confirm that any correspondence graph whose input side has an empty melody tier (i.e., no star nodes) but whose final mora corresponds is associated to a H node on the surface will not contain the subgraph in $\phi_{No^*\text{-}L\ltimes}$, and so banning this substructure will help realize the final H pattern for unaccented words. Again, any other possible correspondences for unaccented words are taken care of by the surface constraints in (7.36). For example, a graph in which *two* (or more) unspecified mora nodes in the input correspond to two (or more) mora specified to H tones in the output may satisfy $\phi_{No^*\text{-}L\ltimes}$, but it will necessarily contain either the subgraphs in ϕ_{HLH} or ϕ_{H^2}:

(7.41) $^*g(\mu_{\mathrm{H}}\mu_{\mathrm{L}}\mu_{\mathrm{H}}) = {}^*$



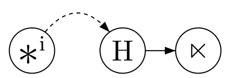
We have thus discussed all of the possible correspondences for input mora nodes not associated with star nodes, and can now turn to the correspondences for input mora which are associated to a star node. First, recall that in Hirosaki Japanese, a final accented mora always results in a F tone. In graph terms, this means that a star node cannot correspond to a final H node, as in the following graph:

(7.42) $^*g(\mu_{\mathrm{L}}\mu_{\mathrm{L}}{*}_{\mathrm{H}})= {}^*$

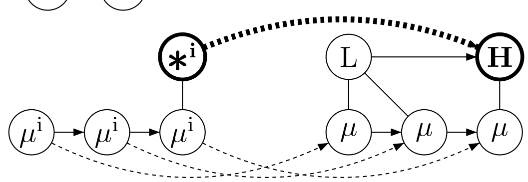


Such graphs can be singled out by the following literal representing a star node corresponding with a final H node:

(7.43) a. $\phi_{*\text{-}H\ltimes} = {}^*$

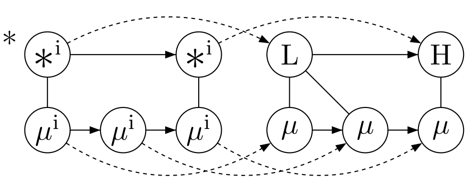


b. $^*g(\mu_{\mathrm{L}}\mu_{\mathrm{L}}{*}_{\mathrm{H}})= {}^*$



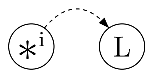
There is one final correspondence which has not yet been discussed. In graphs in $CG(\Gamma)$ which contain the primitive $g(*_{\mathrm{L}})$, a star node corresponds to an output L node, which can be interpreted as an accent being deleted. Such a graph is given below.

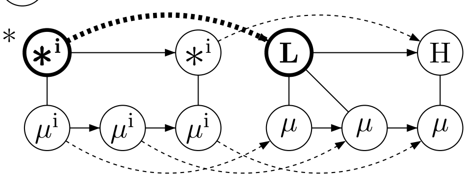
(7.44) $^*g(*_{\mathrm{L}}\mu_{\mathrm{L}}{*}_{\mathrm{H}}) = {}^*$



While accents do delete in compounds in Hirosaki Japanese (Haraguchi, 1977), this discussion is focused on the simple case of single words, and shall thus abstract away from such compound rules.[10] We thus want to exclude any cases in which an input star node corresponds to an output L node. The following graph literal specifies exactly the set of graphs containing such a correspondence, as exemplified in (7.45b).

[10] Such rules can likely be described by not banning $\phi_{*\text{-}L}$ below.

(7.45) a. $\phi_{*\text{-}L}$ = 

b. $*g(*_{\mathrm{L}}\mu_{\mathrm{L}}*_{\mathrm{H}})$ = * 

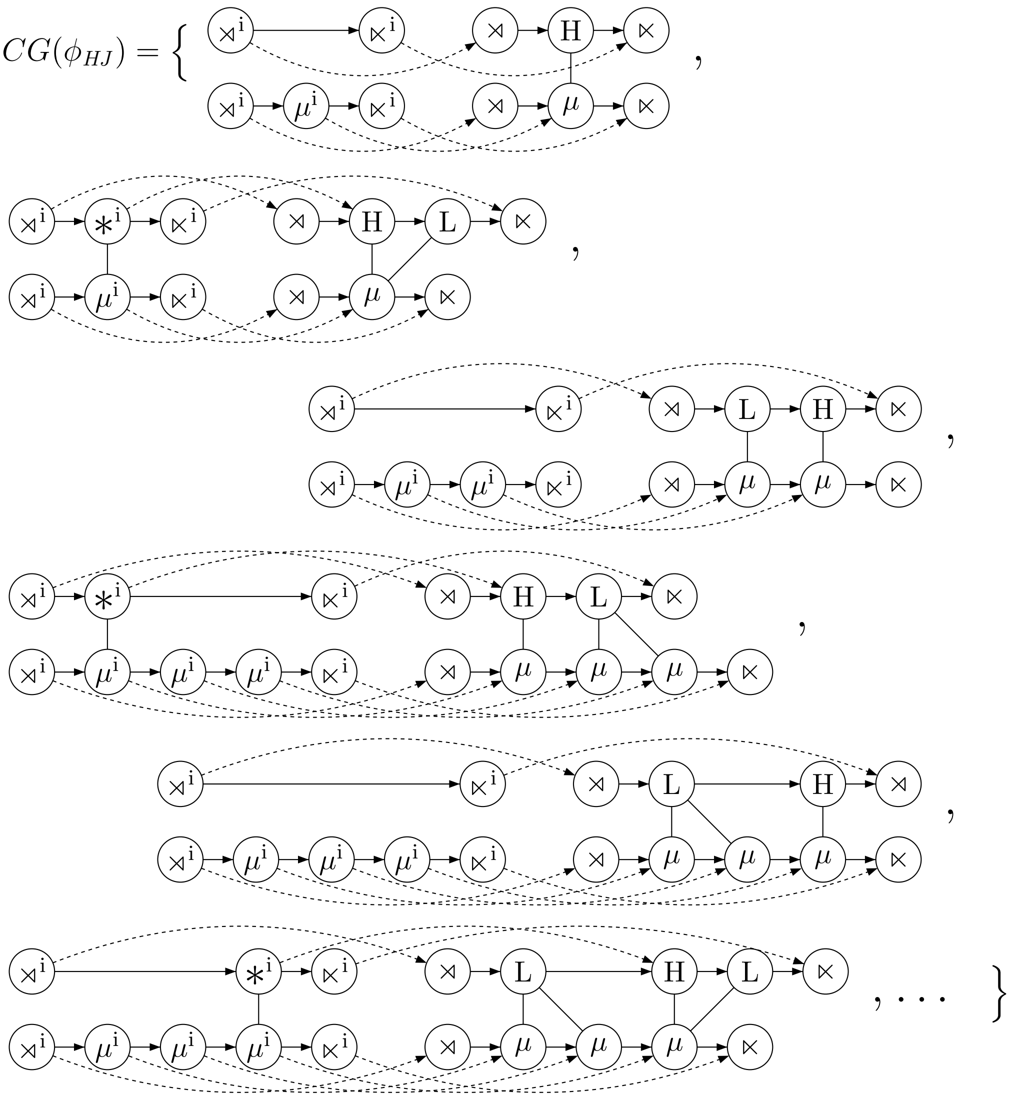
By adding $\neg\phi_{*\text{-}L}$ to our conjunction of negative literals, we can capture the generalization that accents do not delete in (single words in) Hirosaki Japanese. Interestingly, through this and the other constraints, we also get for free the generalization that underlying forms in Hirosaki Japanese have at most one accent. Consider any graph in $CG(\Gamma)$ which has on the input side of its graph more than one star node. Given that $CG(\Gamma)$ is built out of the concatenation of the correspondence graph literals in Γ in (7.27), each of these input star nodes can correspond to an L node, a H node, or an H node followed by an L in the output. Of course, $\neg\phi_{*\text{-}L}$ excludes the first choice for all of these star nodes. However, the other two choices are intractable, because if each star node corresponds to its own H in the output, multiple μ nodes in the output will be associated to H nodes, and thus either $\neg\phi_{HLH}$ or $\neg\phi_{H^2}$ will be violated. As such, any such graph thus cannot appear in the set of graphs described by the conjunction of graph literals. In other words, the graph literals so far discussed can specify a set of graphs in $CG(\Gamma)$ in which at most one star node appears in the input, without specifically stating that constraint.

We have thus covered all possible correspondences as defined by Γ in (7.27), and shown the graph literals which can restrict them to the ones that are valid for the Hirosaki Japanese pattern R_{HJ}. The full conjunction of negative literals in $\mathfrak{L}_G^{NL}$ utilizing all of these literals is given below as ϕ_{HJ} below in (7.46).

(7.46) $\phi_{HJ} = \neg\phi_{\rtimes\mathrm{L}\ltimes} \wedge \neg\phi_{HLH} \wedge \neg\phi_{H^2} \wedge \neg\phi_{NF\text{-}Cont} \wedge \neg\phi_{No*\text{-}L\ltimes} \wedge \neg\phi_{*\text{-}H\ltimes} \wedge \neg\phi_{*\text{-}L}$

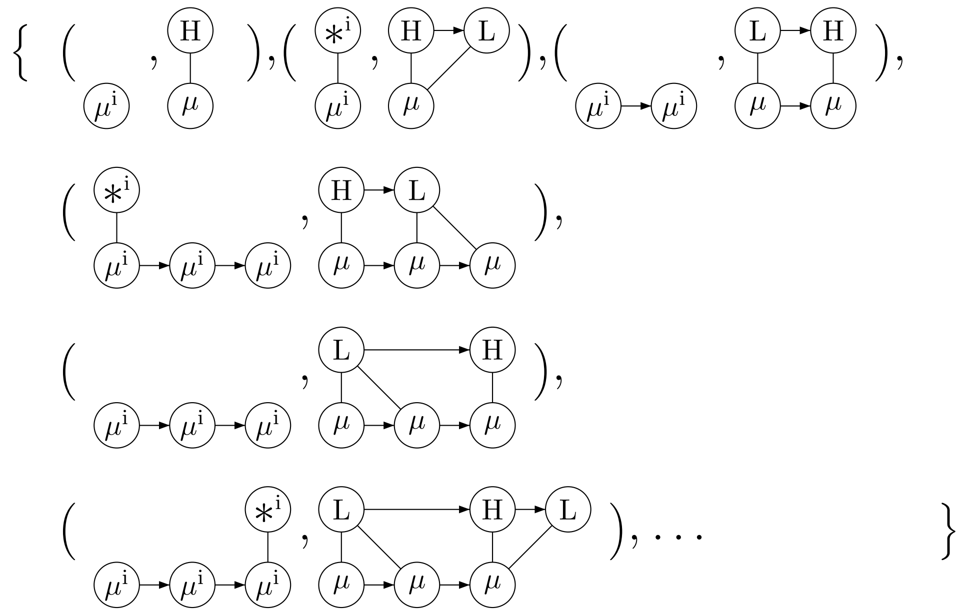
Let $CG(\phi_{HJ})$ be the set of graphs in $CG(\Gamma)$ which satisfy ϕ_{HJ}. Examples from this set are given below. As this is the first example, and as the constraints in ϕ_{HJ} make extensive reference to boundary nodes, the boundary nodes have been included (although they shall usually be omitted henceforth).

(7.47) $CG(\phi_{HJ}) = \{$



Recall that for a set of correspondence graphs CG, $R(CG)$ maps the graphs in CG to input/output pairs (Definition 26). For $CG(\phi_{HJ})$, $R(CG(\phi_{HJ}))$ is thus as in (7.48).

(7.48) $R(CG(\phi_{HJ})) =$



The reader can confirm that $R(CG(\phi_{HJ})) = R_{HJ}$.

7.3.3 Interim conclusion: graph relations through constraints on correspondence graphs

We have thus shown how a graph relation can be specified using a set of correspondences and banned subgraph constraints in $\mathfrak{L}_G^{NL}$. The particular case of Hirosaki Japanese was interesting for a number of reasons. First, while established to be a 'long-distance'-type pattern in Chapter 3, as transformation between APGs it is entirely describable with *local* constraints over correspondence graphs. Interestingly, the surface constraints already established in Chapter 6 for Hirosaki Japanese play a large role in specifying the graph transformation. This preserves the original insight of OT, which is that surface well-formedness constraints motivate phonological processes.

Two, we saw that it is possible to restrict ourselves to the correct set of underlying forms without explicitly writing constraints on the underlying forms. This distinguishes correspondence graph grammars from both rewrite-rules and OT grammars. In rewrite-rule grammars, *morpheme structure constraints* stipulate the structure of underlying forms through statements independent of the rewrite rules (Chomsky and Halle, 1968). In OT, the grammar says nothing about underlying forms (due to the principle of Richness of the Base) and thus a separate mechanism, Lexical Optimization, is recruited to specify underlying forms of morphemes (Prince and Smolensky, 1993, 2004).

A few words should be said about the particular correspondence graph primitives in Γ. Many sets of graph primitives are logically possible, but given that these are *linguistic* graph transformations, there are limits on what kind of correspondence graph primitives will appear. For example, an input H will never correspond to an output μ. There then clearly should be a *theory* of graph primitives for transductions, much like Definition 17 of APG primitives given in Chapter 4. However, it will be sufficient here to simply stipulate the graph primitives used to describe R_{HJ} and the following relations, as the focus of this chapter is on the ability of logical constraints to capture language-specific transductions. I will thus leave the development of a theory of the universal constraints on APG graph relations to future work. However, this Γ and the similar ones used in the remainder of this chapter follow a few assumptions. One, input nodes only correspond to 'like' input nodes on the surface (i.e., no H nodes corresponding to output μ nodes). Two, as has so far been assumed in the previous chapters, the output side of the graph is fully specified (i.e., each node on the timing tier is associated to a node on the melody tier).

7.4 More Autosegmental Transformations

This section builds on the previous section to show how some of the other tonal patterns discussed in this dissertation, when viewed in terms of input/output transformations, can be described with local constraints on correspondence graphs. The patterns discussed in this section are those of UTP, Mende, and Hausa. This section is not meant to be a full typological survey of the describability of tonal processes with local constraints. Its main purpose is to present a framework for applying the surface constraints introduced in previous chapters to phonological transformations over autosegmental graphs. As can be seen in this section and discussed further in the conclusion of this chapter, a full theory of tonal transformations will have to answer a lot of questions.

7.4.1 UTP

Recall that Unbounded Tone Plateauing (UTP), which is found in, for example, Luganda (Hyman and Katamba, 2010), multiple underlying H tones merge into a single H tone in the output associated to all TBUs in between. The following example is repeated from Chapter 2.

(7.49) /mu-tém-a/ + /bi-sikí/ → mutémá-bísíkí ‘log-chopper’

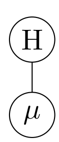
H H H

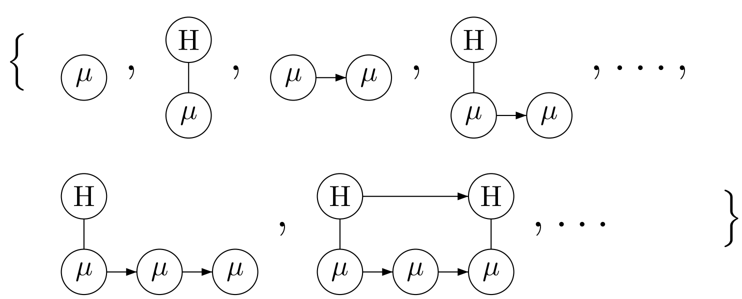
Hyman and Katamba (2010) formalize the process thusly:

(7.50) $\mu \quad \mu^n \quad \mu \quad \rightarrow \quad \mu \quad \mu^n \quad \mu$

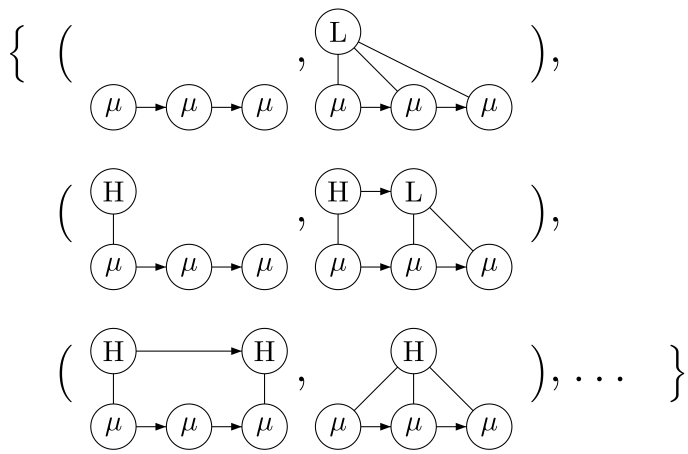
H H H

The surface pattern of UTP was shown in Chapter 6 to be describable by $\mathfrak{L}^{NL}_{G}$ constraints over $APG(\{\mathrm{H},\mathrm{L}\})$. The APGs corresponding to the underlying forms in (7.49) and (7.50) can be produced similarly to those for Hirosaki Japanese in the preceding section. Let us define an alphabet of graph primitives $\{\mu, \mathrm{H}\}$, also concatenated with the non-merging operation $\odot$. This alhabet of graph primitives and its corresponding set $APG(\{\mu, \mathrm{H}\})$ of graph primitives are given below.

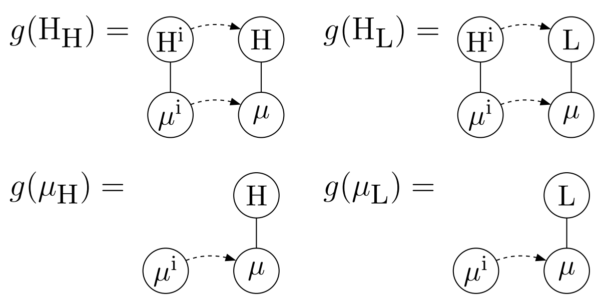
(7.51) $g(\mu) =$  $g(\mathrm{H}) =$ 

(7.52) $APG(\{\mu, \mathrm{H}\}) =$ 

As a process, UTP can thus be viewed as a relation between graphs in $APG(\{\mu, \mathrm{H}\})$ and graphs in $APG(\{\mathrm{H}, \mathrm{L}\})$. Call this relation R_{UTP}, which is given below (examples with three μ nodes are depicted).

(7.53) $R_{UTP} =$ 

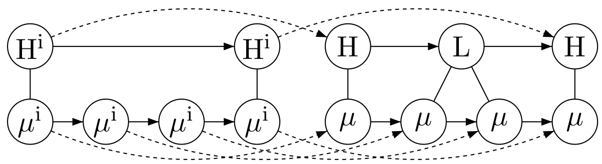
These pairs can be seen as exhibiting correspondences between unassociated μ nodes and μ nodes associated with an H in the input and μ nodes associated with H and L nodes in the output. We can generate these correspondences with an alphabet of correspondence graph primitives very similar to the one used in the previous section for Hirosaki Japanese:[11]

(7.54) 

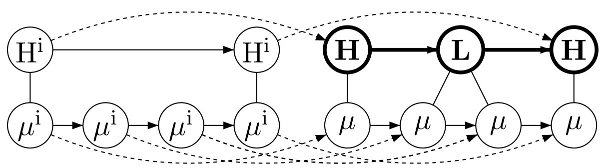
Let $\Gamma = \{\mathrm{H_H}, \mathrm{H_L}, \mu_\mathrm{H}, \mu_\mathrm{L}\}$. We can specify a subset of the corresponding $CG(\Gamma)$ which represents the relation R_{UTP} using constraints in $\mathfrak{L}_G^{NL}$.

[11] It is entirely possible to include the primitives from (7.27), but add a constraint to the grammar below describing R_{UTP} banning subgraphs corresponding to star nodes and contours. Again, this is because individual correspondence graph primitives can be banned as subgraphs (see also Footnote 8). In this way, R_{UTP} and R_{HJ} can be derived through constraints in $\mathfrak{L}_G^{NL}$ over graphs from the same $CG(\Gamma)$. However, for the sake of simplicity, the following exposition uses a smaller set of correspondence graph primitives.

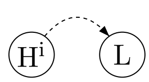
As with R_{HJ} in the preceding section, the constraint established in Chapter 6 which describes the surface pattern of UTP will be very useful. For examle, the following graph $g(\mathrm{H_H}\mu_\mathrm{L}\mu_\mathrm{L}\mathrm{H_H})$ is a valid graph in $CG(\Gamma)$, yet no ‘plateau’ is seen between the output H nodes.

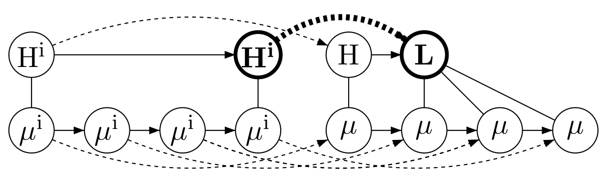
(7.55) $*g(\mathrm{H_H}\mu_\mathrm{L}\mu_\mathrm{L}\mathrm{H_H}) = *$ 

In Chapter 6, it was shown that the surface pattern of UTP was describable by banning a single subgraph, that of ϕ_{HLH}, which specifies two H nodes on the melody tier of the output. As in the above analysis of the Hirosaki Japanese transformation, this can also be used to specify a correspondence such as in (7.55).

(7.56) $*g(\mathrm{H_H}\mu_\mathrm{L}\mu_\mathrm{L}\mathrm{H_H}) = *$ 

Also like Hirosaki Japanese, input H tones do not delete in UTP. We thus also consider the H node analog of (7.45) $\phi_{*\text{-}L}$, $\phi_{H\text{-}L}$, in which an input H node corresponds to a L node. This specifies invalid graphs such as $g(\mathrm{H_H}\mu_\mathrm{L}\mu_\mathrm{L}\mathrm{H_L})$, which does not contain two H nodes in the output because the second input H node corresponds to an output L, below.

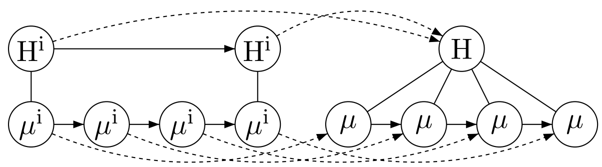
(7.57) a. $\phi_{*\text{-}L} =$ 

b. $*g(\mathrm{H_H}\mu_\mathrm{L}\mu_\mathrm{L}\mathrm{H_L}) = *$ 

Finally, in Hirosaki Japanese, multiply associated H tones in the output were also banned.[12] Thus, the only repair was to completely exclude graphs with multiple

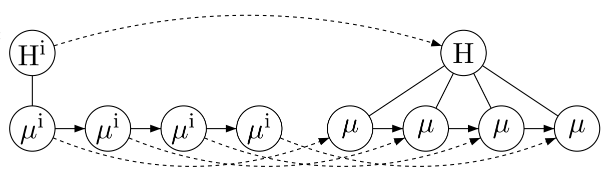
[12] Remember that, because no merging occurs in concatenation of the input side of the graphs in the $CG(\Gamma)$ we are using here, multiply associated H tones cannot occur in the input.

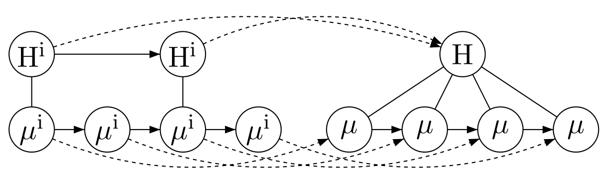
star nodes in the input. However, this is not the case in UTP, in which H tones often associate to multiple morae. Thus, there is one possible output for input graphs with multiple input H nodes: for these H nodes to correspond to the same H node in the output. In order for this to happen, the intervening μ nodes must also correspond to μ nodes associated to an H; this correspondence is represented by a μ_{H} graph primitive. Thus, the input side of the following graph is identical to those in (7.55) and (7.57b), but it contains neither the subgraphs of ϕ_{HLH} nor $\phi_{H\text{-}L}$.

(7.58) $g(\mathrm{H}_{\mathrm{H}}\mu_{\mathrm{H}}\mu_{\mathrm{H}}\mathrm{H}_{\mathrm{H}}) =$ 

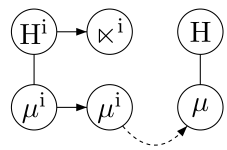
Thus, banning ϕ_{HLH} and $\phi_{H\text{-}L}$ is all that is required in order to ensure plateauing.

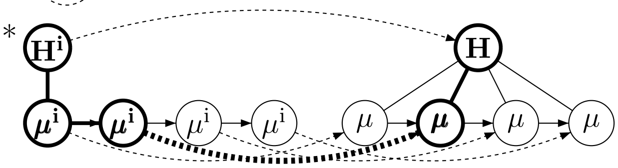
However, there need to be more constraints on when an input μ node which is not associated to a H node in the input corresponds to a μ node associated to a H node in the output. For example, consider the following correspondence graphs. In (7.59a), a single input H node corresponds to a multiply-associated output H node. In (7.59b), the plateau continues beyond the original position of the second H. Neither these represents an invalid input/output pair in R_{UTP}, as there is 'unmotivated' spreading of the output H.

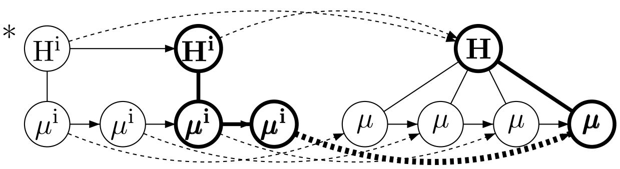
(7.59) a. $*g(\mathrm{H}_{\mathrm{H}}\mu_{\mathrm{H}}\mu_{\mathrm{H}}\mu_{\mathrm{H}}) = *$ 

b. $*g(\mathrm{H}_{\mathrm{H}}\mu_{\mathrm{H}}\mathrm{H}_{\mathrm{H}}\mu_{\mathrm{H}}) = *$ 

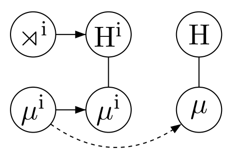
Such graphs contain subgraph in $\phi_{FH\text{-}SpreadR}$ below, which is contained by any graph in which the μ node following a final input H node corresponds to an output μ node associated to an H node. This situation can only occur when the final H node has spread to the right of the mora to which it had originally been associated.

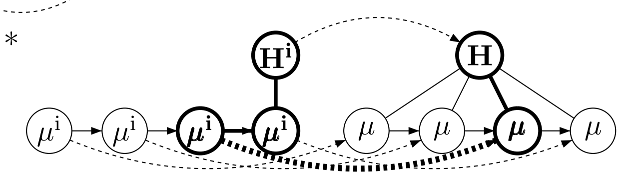
(7.60) a. $\phi_{\textit{FH-SpreadR}} =$ 

b. $*g(\mathrm{H_H}\mu_\mathrm{H}\mu_\mathrm{H}\mu_\mathrm{H}) = *$ 

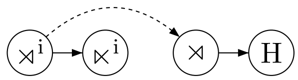
$*g(\mathrm{H_H}\mu_\mathrm{H}\mathrm{H_H}\mu_\mathrm{H}) = *$ 

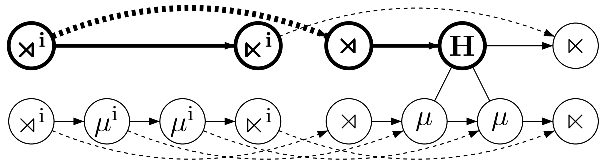
Similarly, we also need to be able to exclude similar spreading to the left of the initial node. This can be done with the mirror image of $\phi_{\textit{FH-SpreadR}}$, $\phi_{\textit{IH-SpreadL}}$.

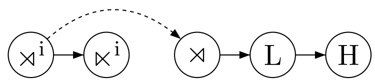
(7.61) a. $\phi_{\textit{IH-SpreadL}} =$ 

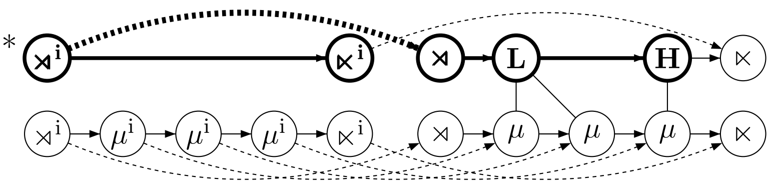
$*g(\mu_\mathrm{H}\mu_\mathrm{H}\mu_\mathrm{H}\mathrm{H_H}) = *$ 

The final issue concerns unspecified input μ nodes corresponding to μ nodes associated with H nodes when no H node appears in the input. For this, we must invoke two subgraph literals similar to $\phi_{\textit{No*-L}\ltimes}$ from the Hirosaki Japanese analysis. These literals, $\phi_{\textit{NoH-}\rtimes H}$ and $\phi_{\textit{NoH-}\rtimes LH}$, specify graphs in which a H node appears in the output when none appears in the input.

(7.62) a. $\phi_{NoH\text{-}\rtimes H} =$ 

b.$^*g(\rtimes\mu_{\mathrm{H}}\mu_{\mathrm{H}}\ltimes) = {}^*$ 

c. $\phi_{NoH\text{-}\rtimes LH} =$ 

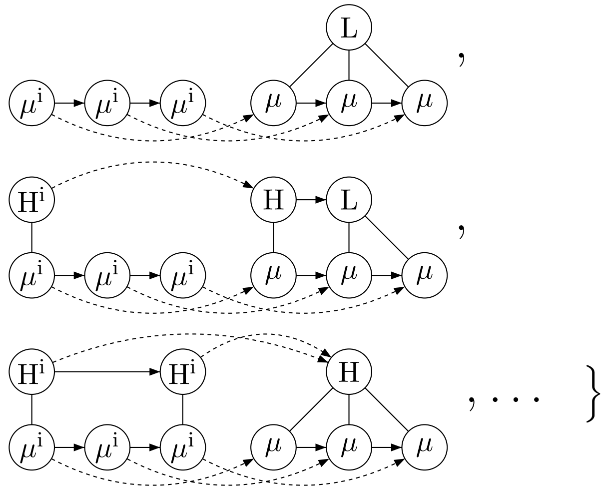
d.$^*g(\rtimes\mu_{\mathrm{L}}\mu_{\mathrm{L}}\mu_{\mathrm{H}}\ltimes) = {}^*$ 

This covers the correspondences of unspecified μ nodes both in the presence of input H nodes and in the absence of input H nodes. We thus have seen all subgraph literals with which we can describe R_{UTP}. The following statement $\phi_{UTP} \in \mathfrak{L}^{NL}_{G}$ bans all of the subgraphs discussed above.

(7.63) $\phi_{UTP} = \neg\phi_{HLH} \wedge \neg\phi_{H\text{-}L} \wedge \neg\phi_{FH\text{-}SpreadR} \wedge \neg\phi_{IH\text{-}SpreadL} \wedge \neg\phi_{NoH\text{-}\rtimes H} \wedge \neg\phi_{NoH\text{-}\rtimes LH}$

The reader can confirm that the set $CG(\phi_{UTP})$ of correspondence graphs in $CG(\Gamma)$is the set of graphs in which a multiply-associated H node appears in the output if and only if there is more than one H node in the input. This is because if there is more than one H node in the input, then they must merge and form a plateau, otherwise $\neg\phi_{HLH}$ will be violated. For any unassociated input μ nodes preceding the first H-toned μ or following the final L-toned μ, they cannot be associated to a H tone or violate either $\neg\phi_{FH\text{-}SpreadR}$ or $\phi_{IH\text{-}SpreadL}$.

(7.64) $CG(\phi_{UTP}) = \{$



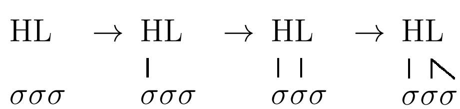
Thus, plateauing occurs exactly in between the first and last Hs in the input, which is the UTP generalization. In formal terms, $R(CG(\phi_{UTP})) = R_{UTP}$.

7.4.2 Directionality in Mende and Hausa

Finally, we can also look at the mappings in Mende and Hausa, in which underlying unassociated tones are associated to TBUs according to language-specific well-formedness constraints, as constraints over correspondence graphs. That these transformations involve underlying tones which are not associated to TBUs means that the set of correspondence graph primitives to be used will be slightly different than in Hirosaki Japanese and UTP.

To review the patterns, for most forms in Mende both spreading and contours occur on the right edge of the word. In terms of a mapping, this was traditionally analyzed as mapping underlying, unassociated tones from left to right, as originally exemplified in (2.49) in Chapter 2 (p. 49). This is repeated below in (7.65).

(7.65)



For the sake of brevity, although in rarer cases Mende forms show other patterns (see Chapter 2, §2.2.1.3 and Chapter 5, 5.3.4), the following discussion will focus on this particular pattern.

Hausa can be described as a near mirror image of Mende, in which spreading and contours occur on the left edge of the word. This has been traditionally analyzed as a right-to-left mapping of unassociated tones to TBUs, as in the following, repeated from (2.52) in Chapter 2 (p. 50).

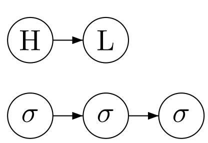
(7.66) HL → H L → HL → HL

$\sigma\sigma\sigma$ $\sigma\sigma\sigma$ $\sigma\sigma\sigma$ $\sigma\sigma\sigma$

As shown in Chapter 5, the patterns in Mende and Hausa can be distinguished by language-specific, local constraints on the well-formedness of the representations. The following will show how the underlying forms like those in (7.65) and (7.66) can be mapped to an output which adheres to these surface constraints directly, without the need of the above intermediate steps. In order to focus on this basic point, the following discussion will show how the surface constraints outlined in Chapter 5 can be recruited to constrain the 'directionality' portion of this mapping. Importantly, other small differences between Mende and Hausa, such as the inventory of melodies and the inventory of contours, will be abstracted away from (the reader is referred back to Chapter 5 to see the relevant surface constraints for these differences).

First, there is the question of what the set of graphs is to which the underlying representations in (7.65) and (7.66) belong. A graph representation of the underlying form in these examples is given below in (7.67).

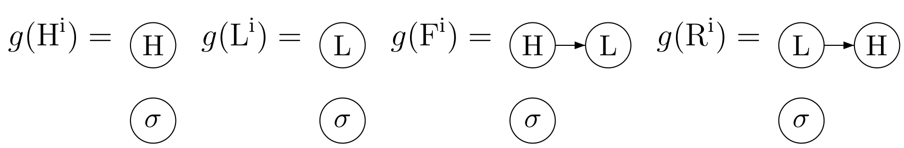
(7.67)



Such a graph is quite different from the other APGs we have seen so far in this dissertation, as there are no association edges at all. As such, it is not immediately clear how to generate it in a way that is comparable to the way the other graphs in this dissertation have been generated. While there are many logical possibilities, this discussion shall choose one, although its utility may not become clear until later.

An important point is that the underlying forms can be seen as adhering to the OCP. To be fair, there has been some debate about this, with some analyses (Dwyer (1978) for Mende, Newman (1986, 2000) for some morphemes in Hausa) arguing that some morphemes require an OCP-violating melody. For example, Dwyer (1978) posits an underlying LLH melody for Mende forms like [lèlèmá] 'mantis' which appear to violate the left-to-right association paradigm (see (2.24) in Chapter 2). However, as it was pointed out in Chapter 2, Leben (1978) shows how adherence to the OCP in such situations can be mantained through an accentual analysis. Again, the discussion here will abstract away from such cases, which can likely be accounted for using representations and constraints such as used for Hirosaki Japanese earlier in this chapter. Regardless, it is not an unreasonable assumption to posit that the URs for Mende and Hausa adhere to the OCP.

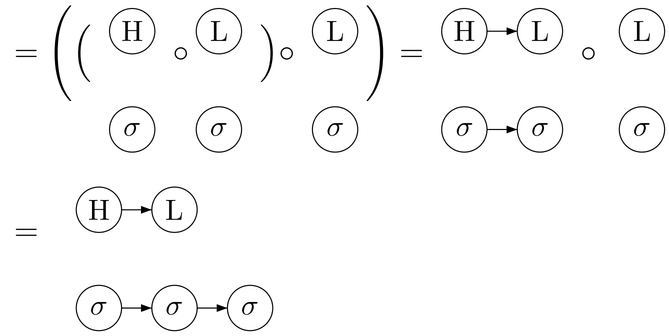
Thus, let us consider (7.67) to be a member of $APG(\{\mathrm{H^i}, \mathrm{L^i}, \mathrm{R^i}, \mathrm{F^i}\})$, defined as the set of graphs generated using the following graph primitives, concatenated through the original, OCP-adhering concatenation operation $\circ$. The 'i' subscript here that these input graph primitives are distinct from the associated versions used to generate the output graphs.[13]

(7.68) 

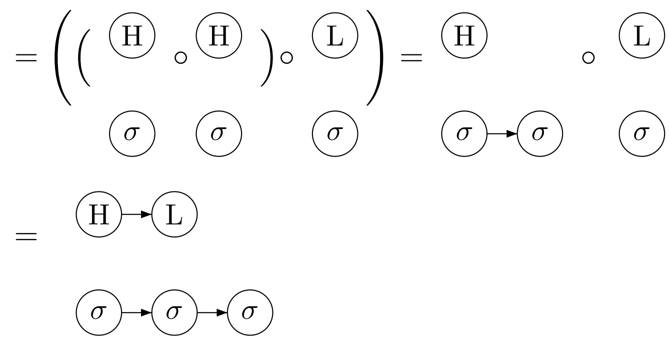
Note that, due to the fact that the OCP-adhering $\circ$ operation merges like, adjacent nodes on the same tier, there are several ways to generate (7.67). Two of these, $g(\mathrm{H^iL^iL^i})$ and $g(\mathrm{H^iH^iL^i})$, are shown below.

[13] Note that these primitives are incapable of generating the rising-falling contour, which appears in Mende but not in Hausa. For Mende, we can add this primitives, and for Hausa, as discussed at the end of Chapter 5, we can either ban this specific contour or by varying the set of primitives available to Hausa.

(7.69) a. $g(\mathrm{H^iL^iL^i})$



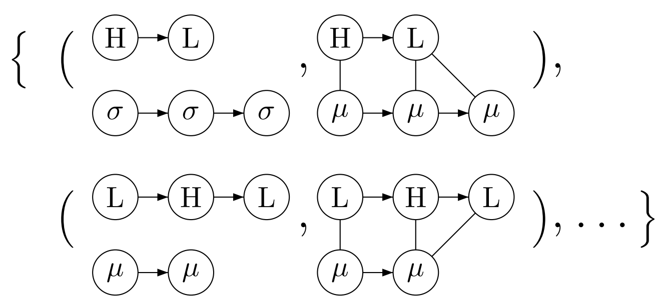
b. $g(\mathrm{H^iH^iL^i})$



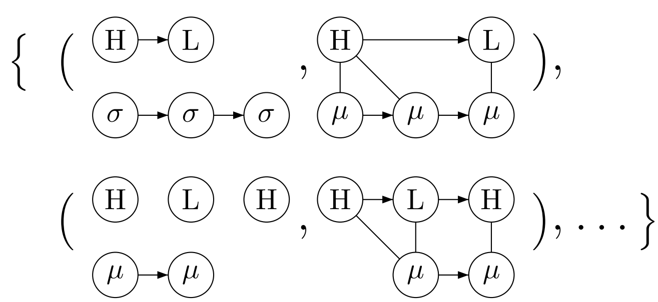
This fact will be important for characterizing the transformation from these representations to the surface representations.

The Mende and Hausa tone mapping generalizations can be seen as subsets of $APG(\{\mathrm{H^i, L^i, R^i, F^i}\}) \times APG(\{\mathrm{H, L, R, F}\})$, where the latter is the standard set of surface APGs. Let the two relations representing the Mende and Hausa generalizations be R_M and R_H, respectively.

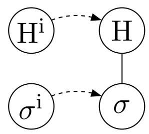
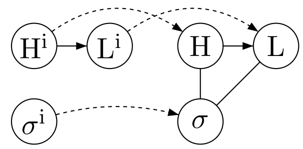
(7.70) $R_M =$

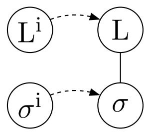
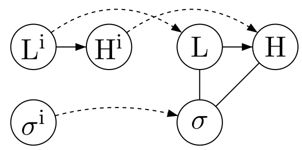


(7.71) $R_M =$



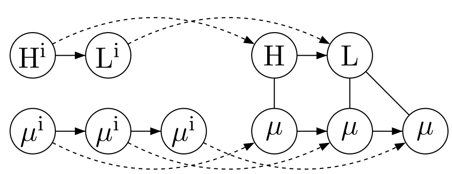
To create correspondence graphs for these two graph sets, let us use the following set of correspondence graph primitives linking the APG primitives from each set.

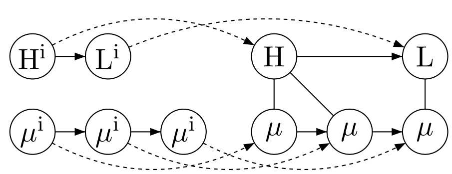
(7.72) $g(\mathrm{H_H}) =$  $g(\mathrm{F_F}) =$ 

$g(\mathrm{L_L}) =$  $g(\mathrm{R_R}) =$ 

This set is rather simple, as it does not contain correspondences involving a change of labels. For example, an input H node cannot correspond to an output L node. It is possible to include such graph primitives, but they would be banned anyway—neither Mende nor Hausa (at least in the generalizations on which we are focusing) delete H tones. As seen above in the UTP discussion, this can be done with a local constraint banning such a correspondence edge between two such nodes. It thus does not effect the locality of R_M or R_H patterns to exclude such primitives.

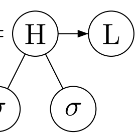
As both input and output sides of the graphs have to adhere to the OCP, let $CG(\{\mathrm{H_H}, \mathrm{L_L}, \mathrm{F_F}, \mathrm{R_R}\})$ be built out of a concatenation operation $\odot_{g'}$ which merges nodes on both the input and output melody tiers. Thus, this set $CG(\{\mathrm{H_H}, \mathrm{L_L}, \mathrm{F_F}, \mathrm{R_R}\})$ contains the sets of correspondence graphs representing the relations of both R_M and R_H. For example, $g(\mathrm{H_H L_L L_L})$ and $g(\mathrm{H_H H_H L_L})$ both have an unassociated input LH melody, however in the former this melody corresponds to an output in which the H is multiply associated, while in the latter it corresponds to an output in which the L is multiply associated.

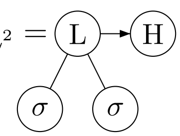
(7.73) a. $g(\mathrm{H_H L_L L_L}) =$ 

b. $g(\mathrm{H_H H_H L_L}) =$ 

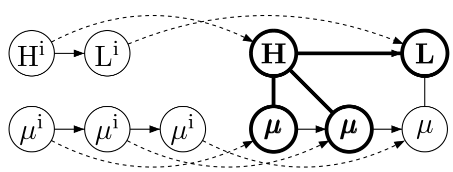
Again, this is thanks to merging on the melody tier on the input side of the graph, as shown above in (7.69). When a part of a correspondence graph built out of the primitives in (7.72), these various ways of building the input graph result correspond to different output graphs. As such, $CG(\{\mathrm{H_H}, \mathrm{L_L}, \mathrm{F_F}, \mathrm{R_R}\})$ contains correspondences that are valid in R_M but not in R_H, such as (7.73a), as well as correspondences which are valid in R_H but not in R_M, such as (7.73b).

To specify the different subsets of $CG(\{\mathrm{H_H}, \mathrm{L_L}, \mathrm{F_F}, \mathrm{R_R}\})$ which correspond to R_M and R_H, respectively, we can simply use the surface constraints directly from Chapter 5 to describe the two patterns. For example, Mende graphs conform to the statement $\neg\phi_{NF\text{-}H^2} \wedge \neg\phi_{LF\text{-}H^2}$ banning multiple association of non-final tones. These constraints, taken from (5.24) in Chapter 5 (p. 161), are repeated below in (7.74).

(7.74) a. $\phi_{NF\text{-}H^2} =$ 

b. $\phi_{NF\text{-}L^2} =$ 

The correspondence graph $g(\mathrm{H_H H_H L_L})) =$ (7.73b), for example, does not satisfy $\neg\phi_{NF\text{-}H^2} \wedge \neg\phi_{LF\text{-}H^2}$, because it contains the subgraph in $\phi_{NF\text{-}H^2}$:

(7.75) $g(\mathrm{H_H H_H L_L})) =$ 

Thus, while $CG(\{\mathrm{H_H}, \mathrm{L_L}, \mathrm{F_F}, \mathrm{R_R}\})$ contains a range of correspondences, it is enough to use the constraints from the analyses of the surface patterns of each language to specify the subsets of $CG(\{\mathrm{H_H}, \mathrm{L_L}, \mathrm{F_F}, \mathrm{R_R}\})$ that represent their mappings. This is because this variation only comes in the order of concatenation of the primitives, not from input nodes corresponding to output nodes of different labels (see discussion following (7.72)).

Let ϕ_M represent the full set of constraints on Mende given in (5.48) in Chapter 5 (p. 162) and ϕ_H represent the full set of constraints on Hausa given in (5.33) (p. 164) in the same chapter. These are given below.

(7.76) a.$\phi_M = \neg\phi_{HLH} \wedge \neg\phi_{NF\text{-}Cont} \wedge \neg\phi_{NF\text{-}H^2} \wedge \neg\phi_{NF\text{-}L^2}$

b.$\phi_H = \neg\phi_{NI\text{-}Cont} \wedge \neg\phi_{NI\text{-}L^2} \wedge \neg\phi_{NI\text{-}H^2} \wedge \neg\phi_{1\sigma\text{-}Cont} \wedge \neg\phi_R \wedge \neg\phi_{HLHL} \wedge \neg\phi_{LHLH}$

Thus, for $CG(\phi_M)$ and $CG(\phi_H)$, defined as the subsets of $CG(\{\mathrm{H_H}, \mathrm{L_L}, \mathrm{F_F}, \mathrm{R_R}\})$ satisfying the surface constraints ϕ_M and ϕ_H for Mende and Hausa, $R(CG(\phi_M)) = R_M$ and $R(CG(\phi_H)) = R_H$. Again, we have defined the input/output relation entirely by referring to the surface constraints. This in fact makes the analysis somewhat similar to Zoll (2003)'s analyses of these tone-mapping patterns, as her analysis also makes reference almost entirely to Markedness, and not Faithfulness constraints (as discussed in §5.4.1 on p. 176 in Chapter 5).

7.4.3 Interim conclusion: banned subgraph constraints and autosegmental transformations

This section built on the ideas exemplified in the graph-based analysis of autosegmental transformations in Hirosaki Japanese to further analyze UTP and the patterns in Mende and Hausa. As with Hirosaki Japanese, when analyzing these processes in this fashion, the local, surface graph constraints developed in the preceding chapters play a large role in specifying the correct graph relation, especially with regards to Mende and Hausa. However, this type of analysis was not without issues. The analysis of UTP required complex constraints, and the analysis of Mende and Hausa required very specific assumptions about the building of the input graphs. Exploring these issues further should be a primary question for future work.

7.5 Conclusion

This chapter developed a theory of phonological transformations, both over strings and over autosegmental structures, which allows us to take advantage of the two main ideas developed in this dissertation. One, we can build a restrictive set of correspondence graphs representing phonological relations through modified versions of the concatenation operation developed in Chapter 4. Two, we can then specify sets of these graphs, representing language-specific phonological transformations, using the banned subgraph grammars developed in Chapter 5. This provides the beginning of a theory of phonological transformations based on locality as it is defined in graphs. However, many questions remain. One, as mentioned above, it is unclear how this notion of locality over relations compares to that in Chandlee (2014).

Furthermore, there were some complex constraints, such as $\phi_{\textit{NF-SpreadR}}$ in the UTP analysis. As discussed at the end of Chapter 5, there is no simpler logic than conjunctions of negative literals, but given graph structures, there is a possibility of forming a more restrictive theory based on restricting the types of subgraphs we may use in our constraints. What is the nature of $\phi_{\textit{NF-SpreadR}}$? Are there any properties it has that make it a valid constraint, as opposed to any arbitrary graph with six nodes? The answers to these questions shall be left to future work.

Finally, there were two important components for the transformations described in this chapter: the set of graphs showing the possible correspondences, and the constraints restricting this set to the language-specific set of correspondences. These are, in implementation, independent. However, it is yet to be determined to what extent the locality of the transformations depends on the concatenation. As shown in §7.2.1, an important property of the particular way that the correspondence graphs were built is that they are *restrictive* in that they disallowed particular kinds of correspondences, such as those reversing a string of phonological units, in a principled way. The generalizations in this chapter would likely still be describable with local constraints given a different theory of how to generate the correspondence graphs, as long as some restrictive notion of correspondence is maintained.

Chapter 8

SOME REMAINING ISSUES

The previous chapters have shown how a theory of banned subgraph constraints can capture language-specific tone patterns, both local and long-distance, and also how the notion of a banned subgraph can be invoked to describe transformations over autosegmental structures. The focus of this chapter is to discuss two remaining issues with these constraints and sketch possible solutions.

First, banned subgraph constraints cannot describe a pattern in which a grammatical substructure—i.e., a substructure that we want to allow in the pattern—is a superstructure of a banned substructure. This is not an obvious problem in string descriptions of phonological patterns, but, as shall be reviewed below, such a situation arises in the autosegmental description of obligatory contours in Aghem (Hyman, 2014). The proposed solution to this *superstructure problem* is to enrich autosegmental representations to include information about local associations which do not occur. Such structure has appeared before in phonological theory, and, as argued in Chapter 3, the addition of representational information is preferable to increasing the power of the logic beyond banned subgraph constraints.

Second, a central issue in a theory of language-specific constraints is how speakers can learn these constraints. The second part of this chapter shows how there is a clear method for learning banned subgraph constraints, based on work in learning these constraints in strings. While this method is not necessarily efficient for graphs in general, as is explained in the chapter, the restricted structure of autosegmental representations suggests that an efficient algorithm for learning banned subgraph constraints over autosegmental graphs does exist. This is a major step towards showing that the theory of language-specific constraints put forward by this dissertation has a provably correct, cognitively plausible learning mechanism.

This chapter is structured as follows. The superstructure problem is identified and its solution is presented in §8.1, then §8.2 addresses the problem of learning banned substructure constraints over APGS.

8.1 Spreading in Aghem: The Superstructure Problem

This section reviews an issue with banned subgraph constraints which I shall call the *superstructure problem*, in which banned subgraph constraints cannot describe a graph set in which an allowed subgraph is a superstructure of a banned subgraph. This is a generalization of the particular problem of obligatory spreading rules involving contours, as shall be exemplified below. This problem, however, has a simple solution: to enrich APGs to include information indicating when associations have *not* occurred. Some of the below discussion has been adapted from Jardine and Heinz (in press), which also concerns the superstructure problem in autosegmental constraints.

8.1.1 Contours in Aghem

Consider the Bantu language Aghem (Hyman, 2014), in which a tones always spread one mora to the right:

(8.1) Aghem (Hyman, 2014)

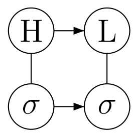
a. /é - nòm/ → [é - nôm] 'to be hot'

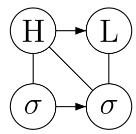
b. /é - nòm - sɔ/ → [é - nôm - sɔ̀] 'to heat (sth.)'

(8.2) a. H L | | e-nom → H L e-nom [é - nôm] 'to be hot'

b. H L | | e-nom-sɔ → H L e-nom-sɔ [é - nôm - sɔ̀] 'to heat (sth.)'

The root [nǒm] 'be hot' is pronounced with a falling tone in both examples in (8.1) because it is associated with both a H tone and an L tone. A constraint on surface well-formedness in Aghem making such spreading obligatory must then distinguish between the following two APGs.

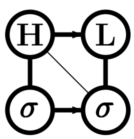
(8.3) a. * 

b. 

HL HF

In (8.3a), we have a structure in which a H tone is followed by a L tone, and both tones are associated to distinct, adjacent segments. I will call this the 'HL' graph, as it will be a useful example for the following discussion. In Aghem the HL structure is banned in favor of the structure in (8.3b), which I will call the HF graph, as it surfaces as a high followed by a falling tone mora.

The HL graph in (8.3a), which we want to exclude, is a subgraph of the HF graph in (8.3b), which we want to keep. This is highlighted in (8.4).

(8.4) 

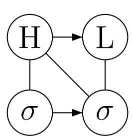
Thus, any banned substructure constraint which bans (8.3a) will also ban (8.3b). This means that from the point of SL graph constraints, the HL graph is indistinguishable from the HF graph. This is the general problem with patterns in which a contour is obligatory in a particular configuration: the configuration *without* the contour is a subgraph of the configuration *with* the contour. Let us call this the *superstructure problem*, as there is a subgraph we wish to allow in our graph set which is a superstructure of a banned subgraph. Note that this is not an issue for the banned sub*string* constraints discussed in Chapter 3; a constraint ¬HL banning the substring HL would still allow strings containing the substring HF. The difficulty with contours and banned subgraph constraints is thus due to the additional structural information in autosegmental representations.

As discussed in detail in Chapter 3, when a particular logical formalism cannot describe an attested pattern, there are two options for developing a theory which can.

The first option is to move to a more powerful logical language, while the second option is to enrich the representation. Just as with strings, the second option is preferable to the first, as it allows for a more restrictive theory of well-formedness. Furthermore, to solve the superstructure problem by moving to more expressive logics over graphs, we must move beyond the power of even full propositional logic, as its statements are still evaluated based on subgraphs.

8.1.2 Solution 1: More expressive logics

To see why, let us try and write a propositinal statement which requires a contour as in the Aghem pattern. Recall that a propositional logic allows not just negative literals (=banned subgraph constraints), but any statement built out of literals (positive or negative) and the usual boolean operators (recall $\mathfrak{L}^P$, the propositional logic over strings, from Chapter 3, §3.3). Thus a propositional logic $\mathfrak{L}^P_G$ for graphs, which properly includes the logic $\mathfrak{L}^{NL}_G$ of banned subgraph constraints, can require substructures with positive literals. For example, the following literal requires the HF subgraph:

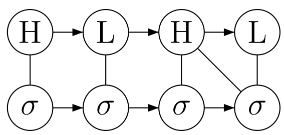
(8.5) $\phi_{HF} =$ 

Recall from Chapter 5, Definition 22 that a graph G satisfies a graph literal if and only if that literal is a subgraph of G. Thus, the set of graphs in which satisfy ϕ_{HF} are exactly those which contain the HF structure as a subgraph. However, this is not an accurate description of the pattern in Aghem, which states that *if* a H tone precedes a L tone, that H must also be associated to the same mora as the L. It is true that propositional logic allows us to write conditionals, such as the following (where ϕ_{HL} is the literal corresponding to the graph in (8.3a)).

(8.6) $\phi_{HL} \rightarrow \phi_{HF}$

This constraint does distinguish between the HL graph in (8.3a) and the HF graph in (8.3b). The HL graph violates (8.6) because it contains its antecedent HL structure, but not also its consequent HF structure. In contrast, the HF graph satisfies both.

However, (8.6) does not specify exactly the set of graphs that describe an Aghem-like pattern. The constraint in (8.6) only says that if the HL graph is present in a graph, then the HF graph must also be present. It says nothing about *where* the HF graph should be. Thus, the graph below in (8.7) satisfies (8.6), even though it contains an HL structure on the surface:

(8.7)


Thus $\mathfrak{L}^P_G$, while more powerful than $\mathfrak{L}^{NL}_G$, does not contain the kind of statements we need to describe Aghem.

One potential solution is then to move to first-order logic (FO), which is expressive enough to enforce properties of individual nodes (and, over strings, is known to be strictly more expressive than propositional logic (Rogers et al., 2013)). However, just as Chapter 3 showed for $\mathfrak{L}^P$ over strings, FO over graphs—henceforth $\mathfrak{L}^{\mathrm{FO}}_G$—overgenerates as a theory for tonal well-formedness.

Briefly, statements in $\mathfrak{L}^{\mathrm{FO}}_G$ are made of *predicates* representing properties of nodes, or relations between nodes, where nodes are represented by *variables*.[1] For example, $H(x)$ is true when the variable x refers to a node that is labeled with H. Given an APG $G = \{V, A, E, \ell\}$ and a node labeling alphabet $\{\mathrm{H}, \mathrm{L}, \sigma\}$ and tier boundary symbols $\{\rtimes, \ltimes\}$ (let's assume that they are the melody tier boundary symbols), we can have the following basic predicates:

[1] This is a very informal discussion of first order logic. A slightly more detailed introduction can be found in Jardine and Heinz (in press), and detailed mathematical introductions can be found in Enderton (1972) and Rogers et al. (2013).

(8.8) $H(x)$ True when x refers to a node in G labeled H

$L(x)$ True when x refers to a node in G labeled L

$\sigma(x)$ True when x refers to a node in G labeled σ

$\rtimes(x)$ True when x refers to a node in G labeled $\rtimes$

$\ltimes(x)$ True when x refers to a node in G labeled $\rtimes$

$arc(x, y)$ True when x and y refer to two nodes i, j in G such that $(i, j) \in A$

$edge(x, y)$ True when x and y refer to two nodes i, j in G such that $\{i, j\} \in E$

Sentences in $\mathfrak{L}_G^{\text{FO}}$ are thus made up of these predicates combined with the standard boolean connectives $\{\wedge, \vee, \neg, \rightarrow\}$ and where each variable is *quantified*. It will be enough to refer to the universal quantifier $\forall$:

(8.9) $(\forall x_1, x_2, ..., x_n)[\phi(x_1, ..., x_n)]$ is true for G when for *all* sets of nodes up to size n, $\phi(x_1, ..., x_n)$ is true for those nodes (where $\phi(x_1, ..., x_n)$ is some set of the above predicates using variables $\{x_1, ..., x_n\}$ and connected by the boolean connectives).

For example, the following statement is in $\mathfrak{L}_G^{\text{FO}}$.

(8.10) $(\forall x, y)[(H(x) \wedge arc(x, y)) \rightarrow L(y)]$

This is true in a graph if for *all* pairs x, y, if x is an H node and y follows H, then y must be a L node. For example, the HL graph, repeated below in (8.11a) satisfies (8.10), whereas (8.11b) does not.

(8.11) a.
$$\begin{tikzcd} \mathrm{H} \arrow[r] \arrow[d, dash] & \mathrm{L} \arrow[d, dash] \\ \sigma \arrow[r] & \sigma \end{tikzcd}$$
b.
$$\begin{tikzcd} \mathrm{H} \arrow[r] \arrow[d, dash] & \mathrm{H} \arrow[d, dash] \\ \sigma \arrow[r] & \sigma \end{tikzcd}$$

In (8.11b), when x refers to the first H node and y refers to the second, they satisfy the first part of the conditional $(H(x) \wedge arc(x, y))$ but not the second $(L(y))$.

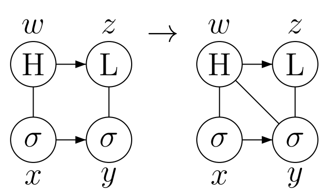
Thus, it does not satisfy (8.10). However, in (8.11a), the H and the L node satisfy both parts of this conditional (and for all other pairs of nodes, e.g. the two σ nodes, the first part of the conditional will be false, and thus it is satisfied).

With statements in $\mathfrak{L}_G^{\mathrm{FO}}$, we have the power to require structures locally, which is exactly what is needed to describe the Aghem contours pattern. The relevant statement is given below in (8.12).

$$\text{(8.12)}\quad \phi_{HF}^{FO} = \quad (\forall w, x, y, z)\big[\big(H(w) \wedge edge(w,x) \wedge L(z) \wedge edge(L,z) \wedge arc(x,y)\big) \\ \rightarrow edge(w,y)\big]$$

This formula forces a particular *structure*. If w is some H node associated to some σ node x, and z is some L node associated to some *sigma* node y, and x immediately precedes y, then w *must also be associated to* y. This conditional can be shown schematically as in (8.13), which explicitly marks the node variables.

(8.13)



This looks like the conditional of propositional literals $\phi_{HL} \rightarrow \phi_{HF}$ from (8.6) above, but the crucial difference is that the two parts of the conditional are 'anchored' by the variable labels: if there is an HL structure, then those *same nodes* must also make up a HF structure. The reader can confirm that the set of APGs which satisfies this statement is exactly the Aghem contour pattern. Thus, through the specification of structures using variables, $\mathfrak{L}_G^{\mathrm{FO}}$ statements can deal with the superstructure problem in a way that $\mathfrak{L}_G^{P}$ cannot.

However, $\mathfrak{L}_G^{\mathrm{FO}}$ is very powerful, and $\mathfrak{L}_G^{\mathrm{FO}}$ statements can also pick out *discontinuous* structures (as $\mathfrak{L}_G^{P}$ does as well). For example, consider the following $\mathfrak{L}_G^{\mathrm{FO}}$ sentence, which is a variation on ϕ_{HF}^{FO} above:

$$\text{(8.14)}\quad \phi_{IH,FH}^{FO} = \quad (\forall w, x, y, z)\big[\big(\rtimes(w) \wedge arc(w,x) \wedge H(x) \wedge \ltimes(z) \wedge arc(y,z)\big) \rightarrow \\ H(y)\big]$$

This sentence states that if there is some H node x preceded by a $\rtimes$ node w, and y is the node preceding the $\ltimes$ node z, then y must also be an H node. The reader can confirm that this conditional is only satisfied by graphs containing the following structure, which is discontinuous because there are no edges relating the nodes w or x to y or z.

(8.15) $\overset{w}{\rtimes} \rightarrow \overset{x}{\mathrm{H}} \quad \overset{y}{\mathrm{H}} \rightarrow \overset{z}{\ltimes}$

Then, it is clear that the set of graphs in $APG(\Gamma)$ over the standard set of graph primitives for $\Gamma = \{\mathrm{H}, \mathrm{L}\}$ which satisfy $\phi^{FO}_{IH,FH}$ are those graphs in which if the first syllable is H-toned, then the last syllable must also be H-toned, regardless of the tone values of the syllables in between. In other words, this is the set of graphs corresponding exactly to the set of strings $L(\phi_{IH,FH})$ discussed in detail in Chapter 3, §3.4.3, as an unattested phonological well-formedness pattern which can be described by logics higher than $\mathfrak{L}^{NL}$. Thus, this is one example of the *global* power of $\mathfrak{L}^{\mathrm{FO}}_{G}$ allowing it to describe unattested patterns. As first-order logic is strictly more powerful than propositional logic, it overgenerates in strictly more ways than propositional logic. For more examples, see Jardine and Heinz (in press) or Rogers et al. (2013).

Thus, by increasing the logical language to $\mathfrak{L}^{\mathrm{FO}}_{G}$, we can solve the superstructure problem, but as a theory of phonological well-formedness it clearly overgenerates. Future work may consider *fragments*, or restrictions, of $\mathfrak{L}^{\mathrm{FO}}_{G}$, but there is currently no clear candidate for autosegmental well-formedness.[2] A more straightforward strategy for dealing with the superstructure problem, which is the strategy advocated for earlier

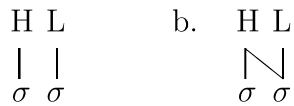
[2] Graf (2010a,b) and (Potts and Pullum, 2002) argue for using *modal logic* in phonology. Modal logic is known to be equivalent to the fragment of first order logic in which any statement can only refer to two variables (see Blackburn et al., 2006). However, this clearly cannot capture patterns like the one in Aghem, which minimally needs to be able to refer to a three-node structure: the H node, the L node, and the σ node to which they are both associated.

in the dissertation, is to instead enrich the structure. This is detailed in the following section.

8.1.3 Solution 2: Enriching the structure

To reiterate the particular instance of the superstructure problem in Aghem, we need to distinguish between the autosegmental HL structure, which is ungrammatical, and the HF structure, which is grammatical:

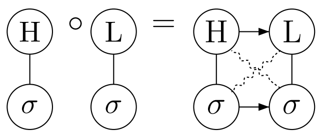
(8.16) a. * b.

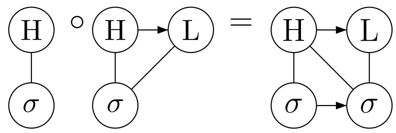


We have thus far seen that, given a faithful graph representation of these two autosegmental diagrams, these two cannot be distinguished by banned subgraph constraints, as the grammatical (8.16b) is a superstructure of the ungrammatical (8.16a). However, with the minimal addition of information to the structure, we can define a representation scheme such that this is no longer true in this particular situation.

The key is to make explicit when association does *not* occur. We can do so by making an addition to the notion of concatenation. Recall that the definition of the concatenation of two APGs defined in Chapter 4 distinguished between 'end' nodes ('end' nodes the last nodes on each tier in the first graph and the first nodes on each tier in the second graph) to be merged (i.e., identical end nodes on the melody tier) and 'end' nodes to be bridged (all other pairs of end nodes). We can add 'anti-association' edges in between the last node on one tier and the first node on the other tier *if neither of them are being merged*. Marking these anti-association lines with wavy, dotted lines, the concatenation of primitives to obtain the HL and HF structures would result in the following graphs:[3]

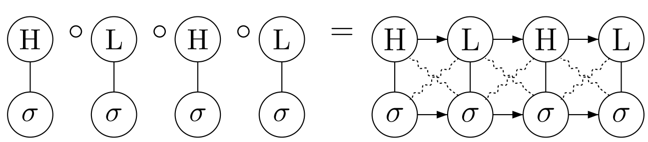
[3] To distinguish between edges representing association lines and edges representing antiassociation lines, we can use labeled edges to mark the two different kinds of association. This informal discussion will not go into the details of this, and so the reader is referred to, e.g., Engelfriet and Hoogeboom (2001) for definitions of graphs with labeled edges.

(8.17) a. 

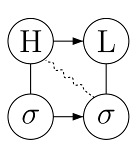
b. 

In (8.17a), neither the H or L nodes are candidates for merging, and so antiassociation lines are drawn between the H node and the σ node of the second primitive and the L and the σ node of the first primitive. In contrast, in (8.17b), the two H nodes from the primitives have been merged, and so no association lines are drawn from them. Note also that no antiassociation line comes from the first σ node to the L node on the melody tier; this is because the L node is not one of the 'ends' of the second primitive, as it is not the first node on its tier. This makes sense in terms of the No-Crossing Constraint: given the NCC, it is not possible that the L nodes and the first σ could be associated, as it would cross the association between the H nodes and the second σ.

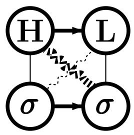
Note that, as they are derived from concatenation, these anti-association lines are 'local' in that they are only drawn between adjacent timing and melody nodes. For example, under this concatenation paradigm, the concatenation of four primitives would look like this:

(8.18) 

If we consider the set of APGs generated with this new concatenation operation, then the HL and HF structures now look very different, as can be seen in (8.17). Crucially, the HL structure (8.17a) is no longer a substructure of HF (8.17b). In fact, they can now be distinguished by banning the following subgraph:

(8.19) ϕ_{HL} = 

The new literal ϕ_{HL} can be interpreted as the following: it singles out a situation in which a H tone has failed to associate to a following L-toned syllable. This structure is a subgraph of the HL structure in (8.17a) (as highlighted below in (8.20)) but *not* of (8.17b). Thus, we can ban the HL structure with the negative subgraph constraint $\neg\phi_{HL}$.

(8.20) 

Thus, antiassociation lines allow us to deal with this particular instance of the superstructure problem by avoiding it completely. But how can they be interpreted phonologically? First, antiassociation lines make a claim that speakers have some knowledge of, for example, where an H tone 'ends'—i.e., the following syllable to which has an antiassociation line. How can we tell if this is psychologically real? For one, it is reasonable to posit that, in a language like Aghem, HF sequences are originally derived from a sequence of H and L syllables (this is reflected in Hyman (2011b)'s synchronic analysis). With antiassociation lines in the representation, this change can be interpreted as speakers making a switch from an antiassociation line (HL) to an association line (HF).

Furthermore, there is precedent for such structure in other analyses of phonology, although not as explicit. For example, take the following constraint from Walker (2011, 2014)'s recent analyses of vowel harmony:

(8.21) ∀HARMONY

For every feature F in a word, a violation is assigned to every vowel to which F is not associated.

Any implementation of ∀HARMONY must have some way of detecting associations that didn't occur. This is exactly what antiassociation lines are.

Of course, note that Walker's constraint requires 'non-local' antiassociation lines, spanning from F to any other potential target in the representation. In the revision to the concatenation paradigm proposed above, antiassociation lines remain 'local' in that they are constrained to adjacent tone/timing unit pairs. As such, this additional structure is not going to allow the same kind of overgeneration that befalls moving to more powerful logics like $\mathfrak{L}_G^{\mathrm{FO}}$. Thus, the more restrictive solution to the superstructure problem is to enrich the structure.

8.1.4 Interim summary: the superstructure problem

The preceding has dealt with a particular instance of what has been dubbed here as the *superstructure problem*: when a grammatical structure is a superstructure of an ungrammatical one. It has been shown that the addition of representational primitives is preferable solution to dealing with this problem than to increasing the logical power.

As detailed in Jardine and Heinz (in press), there can be other versions of the superstructure problem in phonology—in particular, when underspecification is used, an underspecified structure cannot be banned, as a specified structure will always be a superstructure of it. However, the result in that work is the same as what was seen here: this problem can be dealt with *locally* by enriching the representation. Future work can look for other instances of the superstructure problem in phonology, which can further our theory of representational primitives.

8.2 Learning

Having introduced a solution to the superstructure problem, we can now turn to the second problem introduced at the beginning of this chapter, which is how it is possible to learn language-specific banned subgraph constraints from positive data. The following presents a simple solution based on results in learning banned substructure constraints in strings. Briefly, the solution is a learning algorithm (henceforth "learner") which, given a set of input data from a particular pattern, remembers all connected substructures of size k. The *banned* substructures can then be inferred from this set. The upshot of this is that it is the *local* nature of these constraints that makes them learnable by focusing on connected substructures of a particular size.

This learner is couched in the paradigm of *exact identification in the limit in polynomial time and data* (de la Higuera, 2010). The question asked by this paradigm is: for a class of patterns, is there a learner which can efficiently induce the grammar for any target pattern in that class, given a finite set of data whose size is bound by the size of the grammar? The class of patterns in our case is the set of graph patterns describable in $\mathfrak{L}_G^{NL}$. While a proof is not offered here, the following shows how proven results in $\mathfrak{L}^{NL}$ string grammars can be applied to $\mathfrak{L}_G^{NL}$, and thus it is likely efficiently learnable from positive data.

What does this mean in terms of phonological theory? It means that, for the theory of language-specific banned subgraph constraints advanced in this dissertation, there exists a method for learning *any* grammar in this theory, and that this learning method is *cognitively plausible* in terms of the computations and data it requires. This method is idealized in the sense that the learner is only expected to learn from 'errorless' data; it is not stochastic, and thus treats each data point equally. This is of course an abstraction from the real world, in which children are exposed to speech errors, foreign words which do not adhere to their language's native phonology, etc. However, the question of how to deal with errors is largely orthogonal to a major question raised by the theory proposed in this dissertation, which is whether or not language-specific banned subgraph constraints are learnable *at all*, even under (reasonably) idealized conditions. As such, the following still forms a step towards solving the learning problem, because it shows how the local nature of these constraints allows them to be learned. As briefly discussed below, future work can build on this knowledge to develop a stochastic learner for learning from imperfect data.

This section is structured as follows. First, §8.2.1 defines the learning paradigm in order to make explicit the criteria for learning. Then §8.2.2 discusses how banned substructure constraints in strings are learned, and §8.2.3 extends this to banned subgraph constraints.

8.2.1 Exact identification in the limit

In order to claim that a grammar formalism is learnable we must define what the set of learning criteria is. The set of critera chosen here is that of *exact identification in the limit in polynomial time and data* (de la Higuera, 2010). Informally, this paradigm states that there must be a learner for the formalism which meets the following strict criteria. First, it must be able to identify any pattern describable by a grammar in the formalism *exactly*. Second, the learner must learn from positive data *only*, although it is assumed that it will not see any data not consistent with a target pattern. Third, it must do so efficiently and on a set of data whose size is bound by the size of the target grammar. While idealized in a number of ways that will be described below, these criteria help to ensure that the learner, while abstract, is at least cognitively plausible, as they set bounds on the number of computations the learner must make and the amount of data it must see in order to correctly learn. The following describes this learning paradigm in the abstract terms necessary to understand the learners described in the subsequent subsections.

Take a class $\mathcal{C}$ of patterns describable by some class $\mathcal{G}$ of grammars. The following will use $L \in \mathcal{C}$ to refer to a pattern in $\mathcal{C}$, $G \in \mathcal{G}$ to refer to a grammar in $\mathcal{G}$, and, as earlier in the dissertation, $L(G)$ to refer to the pattern described by G. 'Pattern' here refers to any potentially infinite set of objects, in particular the formal languages and graph sets that have been the object of much of the discussion in this dissertation. A class is just a (again, potentially infinite) set of patterns; for instance, the set of languages describable by $\mathfrak{L}^P$, or the set of graph sets describable by $\mathfrak{L}^{NL}_G$. A class $\mathcal{C}$ is learnable if there exists an algorithm $\mathcal{A}$ which satisfies a number of conditions. First, $\mathcal{A}$ must take in input data from a member of $\mathcal{C}$ and return a grammar in $\mathcal{G}$.

Definition 27 (Learner for a class) *A* learner for a class *$\mathcal{C}$ of patterns describable by a class $\mathcal{G}$ of grammars is an algorithm $\mathcal{A}$ which takes as input a finite* input sample *$D \subseteq L$ for some member L of $\mathcal{C}$ and returns a grammar $G \in \mathcal{G}$. Let $\mathcal{A}(D)$ refer to the output of $\mathcal{A}$ on D.*

The input sample D is the formalization of the experience of the learner; it represents, for example, the set of sentences a child hears, or, more relevant to phonology, the set of words a child hears. Note that the above assumes that D is *consistent* with L, that is, all members of D are members of L. Again, this is an idealization, assuming that the learner is exposed to no ungrammatical input.

Finally, note that Definition 27 does not define whether or not $\mathcal{A}(D)$ is *correct*. In order to do so, we need to define a *characteristic sample*. A characteristic sample is a minimal set of data the learner needs to see in order to succeed. We add the further restriction that *if* the learner has seen the characteristic sample, it must *always* return the correct grammar, no matter what other data it sees (still assuming that it will only see data consistent with the target pattern).

Definition 28 (Characteristic sample) *For a learner $\mathcal{A}$, a* characteristic sample *CS for a language $L \in \mathcal{C}$ is some finite subset of L such that for all input samples D of L such that $CS \subseteq D$, $\mathcal{A}(D) = G$ such that $L(G) = L$.*

In other words, the presence of the characteristic sample for L in an input sample is enough to guarantee that $\mathcal{A}$ correctly learns a grammar for L. Note that the criterion $L(G) = L$ means that $\mathcal{A}$ has to learn the target language *exactly*.

There are two more criteria for $\mathcal{A}$, which are that it must be 'reasonable' in the sense that it is efficient and there is some bound on the amount of data it needs to succeed.

Definition 29 (Identification in polynomial time and data) *A class $\mathcal{C}$ of patterns and $\mathcal{G}$ of grammars is identifiable in polynomial time and data if there exists a learner for the class $\mathcal{A}$ and two polynomial functions $p()$ and $q()$ such that*

a. For any input sample D of size m for any pattern $L \in \mathcal{C}$, $\mathcal{A}$ runs in $\mathcal{O}(p(m))$ time.

b. For each grammar $G \in \mathcal{G}$ of size n, there is some characteristic sample of $L(G)$ for $\mathcal{A}$ such that the size of the characteristic sample is at most $\mathcal{O}(q(n))$.

The term $\mathcal{O}(p(m))$ in Definition 29a simply means that the number of steps $\mathcal{A}$ has to take to return a grammar $\mathcal{A}(D)$ will grow at most polynomially as the size of

D increases. (How to measure the size of data will become clearer with the examples in the following subsections.) In other words, the larger D gets, there is some bound on the time it will take $\mathcal{A}$ to run (e.g., it won't run exponentially longer the more data it sees).[4] This helps to ensure that $\mathcal{A}$ is cognitively plausible, as we can reasonably assume that human brains can implement algorithms which run in polynomial time (which, in terms of time complexity, is reasonably efficient).

Similarly, term $\mathcal{O}(q(n))$ in Definition 29b indicates that the size of the characteristic sample needed for $\mathcal{A}$ to correctly identify $L(G)$ will grow at most polynomially as the size of G increases. In other words, no matter how large a grammar G is, the characteristic sample of $L(G)$ is bounded by the size of G. Again, this ensures cognitive plausibility because we know that $\mathcal{A}$ does not need some unreasonably large set of data in order to correctly learn a pattern.

These concepts are best illustrated through the learning of substructure constraints in strings, which is the subject of the following subsection.

8.2.2 Learning banned substructure constraints in strings

We can now see how the *local* nature of banned substructure constraints makes them learnable under this paradigm in strings. The basic idea is this: if a learner is given a fixed window k of substrings to examine, it can accurately learn any pattern describable by banned substructure constraints of size less than or equal to k.

If we take $\mathfrak{L}^{NL}$, the set of conjunctions of negative literals over strings, to be our set of grammars $\mathcal{G}$, then as discussed in Chapter 3, §3.4.4, the class $\mathcal{C}$ of patterns we are able to describe are the *Strictly Local* (SL) (McNaughton and Papert, 1971). Let SL denote this class. For example, L_{KJ}, the pattern L_{KJ} of Kagoshima Japanese tone, in which a high tone can appear in either ultimate or penultimate position, is a member of SL. This formal language, along with the statement in $\mathfrak{L}^{NL}$ that describes it, is repeated in (8.22) below from (3.3) and (3.7) in Chapter 3, §3.3.2 (p. 88). The

[4] For a primer on time complexity, the reader is referred to Cormen (2013).

statement of ϕ_{KJ} below uses the substring literals themselves instead of the naming convention for the statements used in that chapter; i.e., ¬HLL instead of $\neg\phi_{HLL}$.

(8.22) a. $\phi_{KJ} = \neg\text{HLL} \wedge \neg\text{HLH} \wedge \neg\text{HH} \wedge \neg\text{LL}\ltimes$

b.

In L_{KJ}	Not in L_{KJ}
H, HL, LH,	L, LL, HH, LLL,
LHL, LLH,	HLL, HHL, HHH,
LLHL, LLLH,	LHH, LLLL, HLLL,
..., LLLLHL,	..., HLLHLHL,
LLLLLH, ...	LHLHLH, ...

Recall that for any statement in $\mathfrak{L}^{NL}$, there is some value k for the longest literal in that statement; in ϕ_{KJ} above, it is 3, as HLL, HLH, and LL⋉ are all of length 3. Let SL_3 be the class of languages describable by a statement in $\mathfrak{L}^{NL}_{G}$ whose longest substring is of size 3. Clearly, L_{KJ} belongs to SL_3. To generalize, let SL_k refer to the class of all formal languages describable by statements in $\mathfrak{L}^{NL}$ whose longest literal is of length k. The following theorem is due to García et al. (1990); Heinz (2010b) (although these works did not talk about SL in logical terms):

Theorem 7 *For any fixed k, the class SL_k is identifiable in the limit from positive data in polynomial time and data.*

I omit a formal proof here, but the learner that accomplishes this is easily demonstrated. Define a function $\texttt{subs}_k$ which takes as an input a string and returns the substrings of length k of that string:

Definition 30 ($\texttt{subs}_k$) *For a string $w \in \Sigma^*$,*

$$\texttt{subs}_k(w) = \{u|\ u \text{ is a substring of } \rtimes w \ltimes \text{ and the length of } u \text{ is } k\}$$

This function works exactly as the scanning procedure described for evaluating constraints in $\mathfrak{L}^{NL}$ in Chapter 3, §3.4.4. We fix a window of size k and scan the input string w, remembering each substring seen in the window. For example, $\texttt{subs}_3$(LLHL) can be calculated as follows:

(8.23) $\boxed{\rtimes\ \mathrm{L}\ \mathrm{L}}\ \mathrm{H}\ \mathrm{L}\ \ltimes \quad \rtimes\ \boxed{\mathrm{L}\ \mathrm{L}\ \mathrm{H}}\ \mathrm{L}\ \ltimes \quad \rtimes\ \mathrm{L}\ \boxed{\mathrm{L}\ \mathrm{H}\ \mathrm{L}}\ \ltimes \quad \rtimes\ \mathrm{L}\ \mathrm{L}\ \boxed{\mathrm{H}\ \mathrm{L}\ \ltimes}$

$\rtimes$LL LLH LHL HL$\ltimes$

$$\texttt{subs}_3(\text{LLHL}) = \{\rtimes\text{LL, LLH, LHL, HL}\ltimes\}$$

Note that this picture is almost identical to the one in (3.10b) from §3.4.4 (p. 94). The $\texttt{subs}_k$ function thus builds on the *local* nature of the constraints; it remembers each scanned substring, without collecting any information about its relationship to other parts of the structure. Note also that the only calculations it makes are each step through the string, plus remembering the k symbols seen in the scanner at each step. Thus, the time complexity of $\texttt{subs}_k$ is linear in the size of w (which is sub-polynomial).

By extending $\texttt{subs}_k$ to sets of strings, we have almost all of the ingredients for a learning algorithm for SL_k. For a set D of strings, let $\texttt{subs}_k(D)$ be defined as the union of all of the substrings of size k of each string in D:

(8.24) $\texttt{subs}_k(D) = \bigcup_{w \in D} \texttt{subs}_k(w)$

(8.25) (Example) If $D = \{\text{LH, LLHL}\}$, then

$\texttt{subs}_3(D) = \{\rtimes\text{LH, LH}\ltimes\text{, }\rtimes\text{LL, LLH, LHL, LH}\ltimes\}$

This gives us what can be called the *observed substrings* of D; that is, the substrings of size k which appear in D. Given a fixed alphabet Σ, a learner can then assume the *banned* substrings of D are exactly those substrings of size k which are not observed in $\texttt{subs}_k(D)$. This is calculable because the set of all logically possible substrings of size k, $\texttt{subs}_k(\Sigma^*)$, is finite.[5] Let $\texttt{banned-subs}_k(D)$ be the function which does this calculation:

Definition 31 ($\texttt{banned-subs}_k$) *For a set of strings $D \subseteq \Sigma^*$,*

$$\texttt{banned-subs}_k(D) = \texttt{subs}_k(\Sigma^*) - \texttt{subs}_k(D)$$

---

[5] This follows from the fact that k is some finite number. The reader can confirm that $\texttt{subs}_k(\Sigma^*) = \Sigma^k \cup (\{\rtimes\} \bullet \Sigma^{k-1}) \cup (\Sigma^{k-1} \bullet \{\ltimes\})$, which is a finite set.

The algorithm for learning SL_k from a data set D is thus simply one that returns a statement in $\mathfrak{L}^{NL}$ made up of the banned substrings of D. Call this algorithm $\mathcal{A}_k^{\mathfrak{L}^{NL}}$:

Definition 32 ($\mathcal{A}_k^{\mathfrak{L}^{NL}}$) *For a finite set of strings $D \subset \Sigma^*$,*

$$\mathcal{A}_k^{\mathfrak{L}^{NL}}(D) = \neg u_1 \wedge \neg u_2 \wedge u_3 \wedge ... \wedge u_n$$

where

$$\{u_1, u_2, u_3, ..., u_n\} = \texttt{banned-subs}_k(D)$$

There are two important things about $\mathcal{A}_k^{\mathfrak{L}^{NL}}$, which are essentially equivalent. First, $\mathcal{A}_k^{\mathfrak{L}^{NL}}$ is extremely *conservative*; any substring it has not seen in the data it assumes to be a banned substring. Conversely, any substring it *has* seen in the data will not appear as a banned substring constraint, and thus counts as an allowed substring. This allows the learner to generalize beyond the finite set of data it has seen.

To see how, let us consider how $\mathcal{A}_3^{\mathfrak{L}^{NL}}$, which learns from substrings of size 3, behaves given two different input samples, $D = \{\text{LH, LLHL}\}$ and $D' = \{\text{LH}, \text{HL}, \text{LLHL}, \text{LLLHL}\}$. As seen above in the example in (8.25), $\texttt{subs}_3(D) = \{\rtimes\text{LH}, \text{LH}\ltimes, \rtimes\text{LL}, \text{LLH}, \text{LHL}, \text{LH}\ltimes\}$, so $\mathcal{A}_3^{\mathfrak{L}^{NL}}(D)$ returns a statement banning all substrings of size 3 except for this set:

(8.26) $\mathcal{A}_3^{\mathfrak{L}^{NL}}(D) =$ $\neg\rtimes\text{HL} \wedge \neg \rtimes \text{HH} \wedge \neg\text{LLL} \wedge \neg\text{LHH} \wedge \neg\text{HLL}\wedge$
$\neg\text{HLH} \wedge \neg\text{HHL} \wedge \neg\text{HHH} \wedge \neg\text{LL}\ltimes \wedge \neg\text{HH}\ltimes$

While $\mathcal{A}_3^{\mathfrak{L}^{NL}}(D)$ contains many of the banned substring constraints in ϕ_{KJ} from (8.22a), the set of strings which satisfy $\mathcal{A}_3^{\mathfrak{L}^{NL}}(D)$ is not equal to L_{KJ}. In fact, the reader can confirm that the set of strings that satisfy $\mathcal{A}_3^{\mathfrak{L}^{NL}}(D)$ is the following finite set:

(8.27) {LH, LHL, LLHL}

Note that $\mathcal{A}_3^{\mathfrak{L}^{NL}}$ has generalized from D, in that it now admits the string LHL, which is a new string that did not appear in D. It does this because it has removed the substrings $\rtimes$LH (from LH in D), LHL, and HL$\ltimes$(from LLHL in D) from the set of banned substrings.

We can see the full power of generalization when we consider D'. The set $\texttt{subs}_3(D')$ adds the substrings LLL and $\rtimes$HL (from the strings LLLHL and HL, respectively) to $\texttt{subs}_3(D)$.

(8.28) $\texttt{subs}_3(D) = \{\rtimes\text{LH, LH}\ltimes, \rtimes\text{LL, LLH, LHL, LH}\ltimes, \rtimes\text{HL, LLL}\}$

As such, $\mathcal{A}_3^{\mathfrak{L}^{NL}}(D')$ is very similar to $\mathcal{A}_3^{\mathfrak{L}^{NL}}(D)$, with the exception that the banned substring constraints ¬LLL and $\rtimes$HL have been removed.

(8.29) $$\mathcal{A}_3^{\mathfrak{L}^{NL}}(D') = \neg\rtimes\text{HH} \wedge \neg\text{LHH} \wedge \neg\text{HLL} \wedge \neg\text{HLH} \wedge \neg\text{HHL} \wedge \neg\text{HHH} \wedge \neg\text{LL}\ltimes \wedge \neg\text{HH}\ltimes$$

This statement is equivalent to ϕ_{KJ} from (8.22a), and thus describes L_{KJ}. Note that, because the learner deals with substrings of size 3, it cannot learn ¬HH from ϕ_{KJ} per se, but instead learns every banned substring constraint of size 3 that contains HH; i.e., $\neg\rtimes$HH, ¬LHH, ¬HHL, ¬HHH, and ¬HH$\ltimes$. The other banned substring constraints are all shared with ϕ_{KJ}.

It is easy to see that once $\mathcal{A}_k^{\mathfrak{L}^{NL}}$ has seen all the *allowed* substrings in a SL_k language it will correctly return a logical statement describing that language. In other words, for a statement $\phi \in \mathfrak{L}^{NL}$, the set of possible substrings of size k *not* banned by ϕ is the characteristic sample for $L(\phi)$ for $\mathcal{A}_k^{\mathfrak{L}^{NL}}$. Once $\mathcal{A}_k^{\mathfrak{L}^{NL}}$ has seen all such substrings in a SL_k language, it will correctly return ϕ, no matter what other strings in $L(\phi)$ are in the data. As seen from the example with L_{KJ} above, this set is very small in comparison to the size of the grammar. In general, the larger the set of banned substrings in the grammar (i.e., the larger the size of the grammar), the smaller the set of observed substrings (i.e., the size of the data) needed to identify that grammar.

This is significant for phonological learning, because it shows that a *local* pattern over substrings, as defined in terms of banned substring constraints, is efficiently learnable by a procedure that simply pays attention to small chunks of input words, as pictorialized in (8.23). Cognitively, it can be interpreted as a child learner paying attention to connected, finite sequences of sounds in the words spoken to them. Of course, there is the question of what the value of k is—that is, what is the size of these chunks that humans pay attention to? This point will be returned to at the end of this section. For more on this and related algorithms and their relationship to local and long-distance phonological patterns over strings, see Chandlee (2014), Heinz (2007, 2010a), Jardine and Heinz (2016), and the references therein. We can now see how this concept can be extended to connected, finite pieces of autosegmental structures.

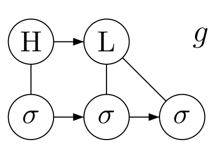
8.2.3 Learning banned substructure constraints in graphs

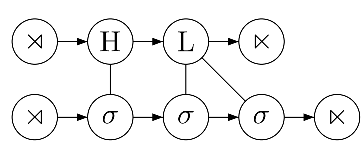
To extend the concepts in learning SL languages from the previous subsection to graphs, we can simply change the scanning function from $\texttt{subs}_k$, which returned sub*strings*, to $\texttt{subg}_k$, which returns sub*graphs* of size k. This can be defined for graphs in general, but let us define it for a set $APG(\Gamma)$ of APGs built from a set Γ of graph primitives (because, in the end, we are just interested in APGs).

Definition 33 ($\texttt{subg}_k$) *For a graph $g(w) \in APG(\Gamma)$, $w \in \Gamma^*$,*

$$\texttt{subg}_k(g(w)) = \{H |\ H \text{ is a subgraph of } g(\rtimes w \ltimes) \text{ with } k \text{ nodes.}\}$$

The scanning procedure now, instead of simply moving from right to left in a string, traverses over the edges of a graph two-dimensionally. Figure 8.1 illustrates partially how such a procedure produces the set of subgraphs of size 3 for the graph $g(\text{HLL})$ (given with boundaries in (8.30)).

(8.30) $g(\text{HLL}) =$ 

$g(\rtimes\text{HLL}\ltimes) =$ 

While the implementation is slightly different, the core idea is the same: the learner focuses only on connected substructures of a particular size. Note that there

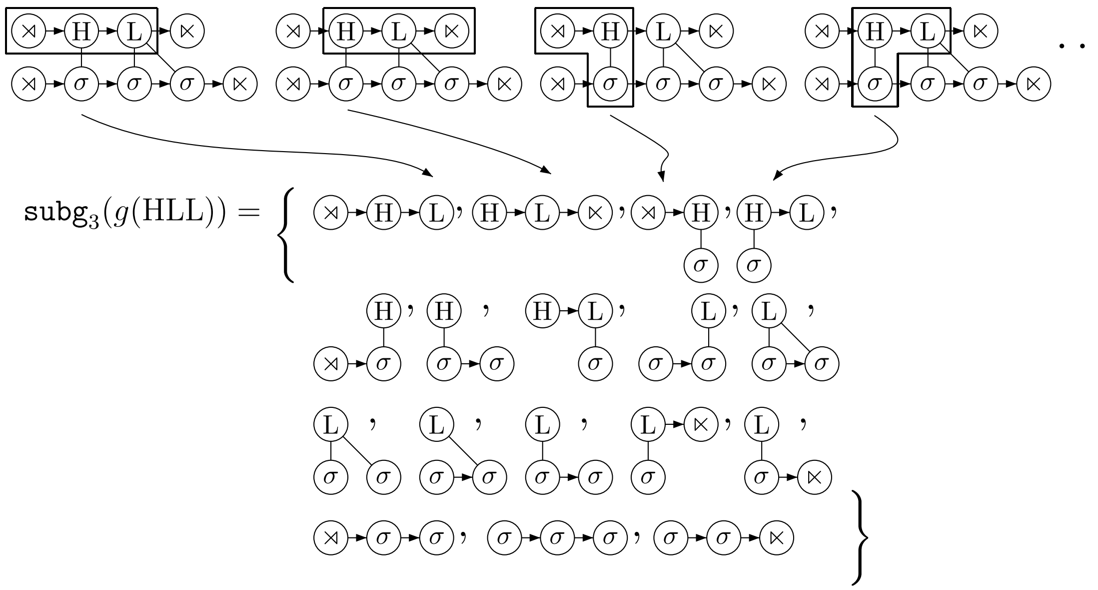


Figure 8.1: Scanning subgraphs of size 3 in the graph $g(\text{HLL})$

are more subgraphs of size 3 in $g(\rtimes\text{HLL}\ltimes)$ than there are, say, substrings of size 3 in the string $\rtimes$HLL$\ltimes$. In general, the number of subgraphs of a particular size in a graph can get very large very quickly, but there are reasons to believe that for the specific case of APGs there will be a reasonable bound on the number of subgraphs of a particular size. This point will be fully explained momentarily.

Having extended the idea of scanning for substrings to subgraphs, the remainder of the learning algorithm is identical to the one introduced in the previous subsection. If we extend $\texttt{subg}_k$ to sets of graphs, just as we did with $\texttt{subs}_k$ and sets of strings, then $\texttt{subg}_k(D)$ for some set D of graphs is the set of *observed subgraphs* of size k in D. We again design a learner who assumes that all subgraphs of size k it *doesn't* see are banned. This set can be defined as follows.

Definition 34 ($\texttt{banned-subg}_k$) *For a set of graphs $D \subseteq APG(\Gamma)$,*

$$\texttt{banned-subg}_k(D) = \texttt{subg}_k(APG(\Gamma)) - \texttt{subg}_k(D)$$

Parallel to its string counterpart (see Definition 31), $\texttt{subg}_k(APG(\Gamma))$ is the set of all subgraphs of size k obtainable from graphs in $APG(\Gamma)$. Again, because k is finite, this set is finite, and so $\texttt{banned-subg}_k(D)$ is computable.

We can then define $\mathcal{A}_k^{\mathfrak{L}_G^{NL}}$ as a procedure which from an input set D of graphs constructs a logical statement in $\mathfrak{L}_G^{NL}$ using $\texttt{banned-subg}_k(D)$.

Definition 35 **($\mathcal{A}_k^{\mathfrak{L}_G^{NL}}$)** *For a finite set of APGs $D \subset APG(\Gamma)$,*

$$\mathcal{A}_k^{\mathfrak{L}_G^{NL}}(D) = \neg G_1 \wedge \neg G_2 \wedge \neg G_3 \wedge ... \wedge \neg G_n$$

where

$$\{G_1, G_2, G_3, ..., G_n\} = \texttt{banned-subg}_k(D)$$

Given $\mathcal{A}_k^{\mathfrak{L}_G^{NL}}$, we can state the following theorem. Parallel to how SL_k are the formal languages described by $\mathfrak{L}^{NL}$, let GSL_k refer to the class of graph sets describable by statements in $\mathfrak{L}_G^{NL}$ whose largest banned subgraph constraint has a graph of size k nodes.

Theorem 8 *For any fixed k, the class GSL_k is identifiable in the limit from positive data whose size is polynomial in the size of the target grammar.*

We can be confident that, given a $\mathcal{A}_k^{\mathfrak{L}_G^{NL}}$ can correctly identify any graph set in GSL_k for the same reasons that $\mathcal{A}_k^{\mathfrak{L}^{NL}}$ can do so for any formal language in SL_k. To correctly identify a sentence $\phi \in \mathfrak{L}_G^{NL}$ (where the largest subgraph in ϕ is of size k) or an equivalent, $\mathcal{A}_k^{\mathfrak{L}_G^{NL}}$ needs to see all of the subgraphs in $\texttt{subg}_k(APG(\Gamma))$ that are *not* in ϕ. Parallel to the characteristic sample for $\mathcal{A}_k^{\mathfrak{L}^{NL}}$, a characteristic sample for $\mathcal{A}_k^{\mathfrak{L}_G^{NL}}$ for $L(\phi)$ is thus any set D of graphs for which the observed subgraphs of size k in D are all of the logically possible subgraphs of size k not banned by ϕ. As the size of this set of subgraphs necessary is bounded by the size of $\texttt{subg}_k(APG(\Gamma))$, and actually decreases with the size of ϕ (where we calculate the size of a set of subgraphs in the usual way by taking the sum of all of their nodes), we can be sure that the characteristic sample is polynomial in the size of the target grammar.

There remains the question of whether or not the scanning procedure outlined above can be done *efficiently*. Ferreira (2013) shows that connected subgraphs of size k of a graph G can be enumerated in time proportional to the size of G and the number of said subgraphs. This last qualification is not trivial, as, for graphs in general, the number of subgraphs of size k in a graph is not bounded polynomially by the size of the graph. In other words, even an efficient algorithm for scanning through subgraphs will run very slowly if there are many subgraphs to identify.

However, as established throughout Chapter 4, APGs are a very specific kind of graph with a number of constraints on their structure. For example, the No-Crossing Constraint, as well as the prohibition on association edges between nodes on the same tier, means that the number of undirected edges will be in direct proportion to the number of nodes in any APG. This likewise holds for the directed edges, which necessarily form linear orders on each tier. These restrictions on the number of edges—which again do not apply to graphs in general—thus severely restricts the number of connected subgraphs. Let us formalize this intuition with the following conjecture.

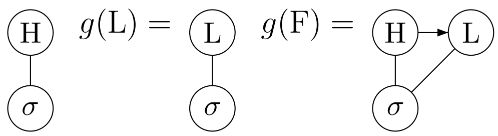
Conjecture 1 *Given a fixed k, the number of subgraphs of size k in an APG G is polynomially bounded by the number of nodes in G.*

Proving this conjecture will be left to future work. If it is true, however, then an algorithm such as Ferreira (2013)'s can enumerate the subgraphs of an input set of graphs in polynomial time. This would mean that $\mathcal{A}_k^{\mathfrak{C}_G^{NL}}$ can identify a target grammar in polynomial time as well as polynomial data, thus satisfying Definition 29 for identification in the limit from polynomial time and data. In intuitive terms, this means that $\mathcal{A}_k^{\mathfrak{C}_G^{NL}}$ is likely a cognitively plausible way, in terms of computational complexity, of learning the language-specific autosegmental constraints proposed by this dissertation.

8.2.4 Learning APG grammars from strings

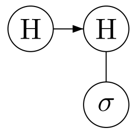
The previous discussion defined $\mathcal{A}_k^{\mathfrak{C}_G^{NL}}$ such that it takes a set of graphs as input. This may be criticized as not realistic as a model of phonological learning, as the input to child learners is arguably a linear stream of sound. However, Chapter 4 has already shown us how autosegmental graphs can be generated from such linear representations, as long as an alphabet of graph primitives is provided in advance. However, prior knowledge of an alphabet of graph primitives is not an unreasonable assumption for a learner. Consider the alphabet $\Gamma = \{\mathrm{H}, \mathrm{L}, \mathrm{F}\}$ of graph primitives with the mapping g to graphs as originally given in (4.27) (from Chapter 4, §4.5.1).

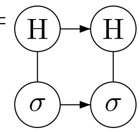
(8.31) $g(\mathrm{H}) =$ $g(\mathrm{L}) =$ $g(\mathrm{F}) =$

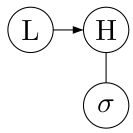


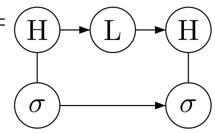
As detailed in that chapter, this set of primitives represents the knowledge that, for example, a F-toned syllable corresponds to the autosegmental structure $g(\mathrm{F})$ in which a HL sequence of tones is associated to a single syllable. Given the concatenation operation defined in Chapter 4, we know then that this knowledge can be generalized to strings of syllables. Thus, for example, the structure $g(\rtimes\mathrm{HLL}\ltimes)$ from the preceding discussion, can be generated from the string of syllables HLL. As such, any finite subset of $APG(\Gamma)$ can be generated from any finite subset of Γ^*. This means that for any pattern over a set of autosegmental graphs generatable by a set of graph primitives and concatenation, we can learn this pattern from a set of input strings.

However, as pointed out in Chapter 4, §§4.5.2 and 4.5.3, it is not simple enough to posit a universal set of graph primitives, as some languages vary on their interpretations of downstep. As shown there, in some languages a downstepped high TBU ($^!$H) may indicate a distinct H tone adjacent to another H, or it may indicate a H tone following a floating L. These two options are repeated from these sections in (8.32) below.

(8.32) a. $g({}^{!}\mathrm{H}) =$ 

$g(\mathrm{H}{}^{!}\mathrm{H}) =$ 

b. $g({}^{!}\mathrm{H}) =$ 

$g(\mathrm{H}{}^{!}\mathrm{H}) =$ 

Thus, the value for $g({}^{!}\mathrm{H})$ must be language-specific. However, it is not inconceivable how a learner might pick from these two options. Pulleyblank (1986), for example, shows how languages with floating tones present abundant evidence for their existence from morphological alternations. A learner could in principle, then, be devised to use this information to make the choice between (8.32a) and (8.32b). However, as this requires a theory of learning from morphological alternations, the articulation of such a learner shall be left to future work.

8.2.5 Interim conclusion: learning banned subgraph constraints

The central lesson of this section has been that a learner which remembers observed substructures of a particular size can efficiently learn patterns describable by banned substructure constraints. This has been demonstrated concretely both for patterns over strings and patterns over APGs. As the preceding chapters in this dissertation have shown how a range of tone patterns can be described by banned substructure constraints over APGs, this thus provides a major step towards the learning of tone.

A caveat is that the learner must know the size of substructures it is looking for; recall that both theorems stated that *given a value* k, patterns describable with banned substructures of size k can be learned. From the perspective of phonological learning, then, what is k, and is it reasonable for a child to have some value for k? As seen in Chapters 5 and 6 of this dissertation, the largest k-value in a banned subgraph constraint was 6, required for (5.58) in Northern Karanga Shona (see Chapter 5, §5.3.5, p. 174). Most of the other constraints had k values of 4 or 5. Thus, it can be that humans use a learner (or, perhaps, a set of learners) with a relatively small value for k.

Finally, as noted at the beginning of the section, the learning model considered here focused on how the local nature of banned substructure constraints contributes to their learnability. It thus abstracted away from the problem of learning from noisy data. However, noisy data is an independent issue, and can be dealt with by developing stochastic versions of banned substructure learners (Heinz and Rogers, 2010). There are also stochastic learning paradigms, such as the Probably Approximately Correct (PAC) framework (Valiant, 1984; Angluin and Laird, 1988). Thus, future work on learning from noisy data can build on the paradigm described here, but it would not change the central result: the local nature of banned subgraph constraints makes them learnable.

8.3 Conclusion

This chapter reviewed two issues which had not yet been addressed in this dissertation: the superstructure problem, in which the banned subgraph constraints in $\mathfrak{L}_G^{NL}$ cannot describe patterns in which a superstructure of a banned subgraph is allowed, and the learning problem, which concerns how the language-specific constraints advanced by this dissertation might be learned. While both of these problems are legitimate concerns, they both have clear solutions. The superstructure problem can be addressed while maintaining the restrictivity of banned subgraph constraints through enriching the structure of the APGs. This additional structure is not only minimal, but it has been proposed in some form in the previous literature. The learning problem can be addressed by borrowing the idea of substructure learning from strings, and it was shown that the learnability of local constraints in strings lifts to APGS. Thus, while both of these issues were only touched on here, there is a clear path forward for future work to fully develop their solutions.

Chapter 9

CONCLUSION

The main contribution of this dissertation was to develop an explicit notion of locality over autosegmental representations and argue for a theory of tonal well-formedness based on this notion. Specifically, it tested the following hypothesis with regards to two major types of tone patterns:

(9.1) **The $\mathfrak{L}_G^{NL}$ Hypothesis:** Surface well-formedness constraints in tonal phonology are local over autosegmental structures.

This hypothesis predicts that language-specific generalizations should be describable by *inviolable*, *language-specific* constraints which *ban substructures* of autosegmental representations. Analyses using these constrants to describe tone-mapping patterns, which exhibit language-specific restrictions on how tones map to tone-bearing units, and long-distance patterns, in which the generalization holds over unbounded stretches of tone-bearing units, were consistent with this hypothesis. The theory of tonal well-formedness based on the banned substructure version of locality was thus argued to be superior to previous theories of tone in that it both captured the local nature of these surface well-formendess generalizations, including ones which have been difficult to describe in Optimality Theory, and it made clear typological predictions that disclude complex, unattested patterns predicted to exist by standard versions of Optimality Theory. It was also shown to be superior to a string-based theory of computational locality in that it captured the full range of the patterns, without overgenerating unattested patterns, like the initial H, final H pattern, that are possible with more powerful string logics. Furthermore, it was also shown how surface locality over graphs could be integrated into a theory of phonological transformations, and it was shown that the language-specific constraints can be learned.

While the hypothesis in (9.1) was confirmed, two problematic cases were discussed. Describing the accent pattern of Wan Japanese Type β words required including morphological information on the melody tier, which deviated from previous analyses, although there were reasonable arguments for this choice. Additionally, a general 'superstructure problem', in which it is impossible to describe a pattern for which well-formed structures can be superstructures of ill-formed structures, was illustrated through the case of Aghem. Again, this issue was shown to be solvable with reasonable changes to the representation. Thus, while these two issues had solutions, they highlight the need to continue to test the hypothesis in (9.1) with regards to attested tone patterns. This dissertation provided evidence, through showing that it held true for major types of tonal generalizations, that it is a viable hypothesis, but the typology of tone is vast, and so future work can continue to explore whether or not this typology can be described in $\mathfrak{L}_G^{NL}$.

It should be emphasized again that the local nature of the tone generalizations observed by this dissertation is a *fact* independent of how they might be analyzed in terms of transformations. This means that for any theory of transformations, adhering to this fact makes it a stronger theory. While this dissertation offered one way of integrating this locality into a theory of transformations based on local constraints in correspondence graphs, there may be ways to integrate it into other theories. One can begin, for example, with Eisner (1997a)'s local theory of Optimality Theory, and Potts and Pullum (2002)'s logical implementation discussed in Chapter 7.

Other venues for future work stem from this dissertation's secondary contribution, which was to show that autosegmental representations are stringlike in that crucial properties such as the NCC and the OCP can be derived from concatenation of a finite set of graph primitives. Defining the universal set of APRs through graph concatenation served as a good working hypothesis for the purposes of this dissertation, but several issues were highlighted, particularly with respect to downstep and underspecification. While solutions for these issues were sketched out, a full articulation of concatenation as a theory of non-linear representation will be left for future work. Additionally, this dissertation demonstrated that this notion of graph concatenation is a natural one for phonology, as it was connected to tonal autosegmental structures, input/output correspondence, and to learning. It is quite possible then that connecting it to other areas of phonology, such as segmental or metrical structure, will yield further insights.

Finally, a major lesson demonstrated throughout this dissertation is that, in the face of a pattern beyond the expressivity of one's current hypothesis, enriching the representation is a more restrictive theoretical choice than increasing the power of the grammar. In other words, *representation really matters.* While necessarily limited in scope to autosegmental representations in tone, this dissertation has established a framework for studying phonological representation in a formal, explicit way. This is thus a substantive step towards understanding the nature of locality and representation in phonology.

BIBLIOGRAPHY

Angluin, D. and Laird, P. (1988). Learning from noisy examples. *Machine Learning*, 2:343–370.

Archangeli, D. and Pulleyblank, D. (1994). *Grounded Phonology*. Cambridge: MIT Press.

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

Blackburn, P., van Benthem, J. F., and Wolter, F., editors (2006). *Handbook of Modal Logic*, volume 3 of *Studies in Logic and Practical Reasoning*. Elsevier B.V.

Booij, G. (1997). Non-derivational phonology meets Lexical Phonology. In Roca, I., editor, *Derivations and Constraints in the Lexicon*. Oxford: Clarendon Press.

Breteler, J. M. W. (2013). A stratal OT analysis of the intonation of Japanese dialects with multiple intonational classes. Master's thesis, Utrecht University.

Büchi, J. R. (1960). Weak second-order arithmetic and finite automata. *Zeitschrift für Mathematische Logik und Grundlagen der Mathmatik*, 6:66—92.

Cassimjee, F. and Kisseberth, C. (2001). Zulu tonology and its relationship to other Nguni languages. In Kaji, S., editor, *Cross-linguistic studies of tonal phenomena: tonogenesis, Japanese accentology, and other topics.*, pages 327–359. Institute for the Study of Languages and Cultures of Asia and Africa (ILCAA).

Chandlee, J. (2014). *Strictly Local Phonological Processes*. PhD thesis, University of Delaware.

Chandlee, J., Athanasopoulou, A., and Heinz, J. (2012). Evidence for classifying metathesis patterns as subsequential. In Choi, J., Hogue, E. A., Punske, J., Tat, D., Schertz, J., and Trueman, A., editors, *Proceedings of the 29th West Coast Conference on Formal Linguistics*. Somerville, MA: Cascadilla Press.

Chandlee, J., Eyraud, R., and Heinz, J. (2014). Learning Strictly Local subsequential functions. *Transactions of the Association for Computational Linguistics*, 2:491–503.

Chandlee, J. and Heinz, J. (to appear). Strictly local functions. *Linguistic Inquiry*.

Chandlee, J. and Jardine, A. (2014). Learning phonological mappings by learning Strictly Local functions. In Kingston, J., Moore-Cantwell, C., Pater, J., and Staubs, R., editors, *Proceedings of the 2013 Meeting on Phonology (UMass Amherst)*, Proceedings of the Annual Meetings on Phonology. LSA.

Chandlee, J., Jardine, A., and Heinz, J. (2015). Learning repairs for marked structures. In *Proceedings of the 2015 Annual Meeting on Phonology*. LSA. To appear.

Chiośain, M. N. and Padgett, J. (2001). Markedness, segment realization, and locality in spreading. In Lombardi, L., editor, *Segmental phonology in Optimality Theory*, pages 118–156. Cambridge University Press.

Chomsky, N. and Halle, M. (1968). *The Sound Pattern of English*. Harper & Row.

Clements, G. N. (1977). Neutral vowels in Hungarian vowel harmony: an autosegmental interpretation. In *NELS 7*, pages 49–64.

Clements, G. N. and Hume, E. V. (1995). The internal organization of speech sounds.

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

Cormen, T. H. (2013). *Algorithms Unlocked*. Cambridge, MA: The MIT Press.

Courcelle, B., Engelfriet, J., and Nivat, M. (2012). *Graph structure and monadic second-order logic: A language-theoretic approach*. Cambridge University Press.

de la Higuera, C. (2010). *Grammatical Inference: Learning Automata Grammars*. Cambridge University Press.

de Lacy, P. (2011). Markedness and faithfulness constraints. In Oostendorp, M. V., Ewen, C. J., Hume, E., and Rice, K., editors, *The Blackwell Companion to Phonology*. Blackwell.

Dwyer, D. (1978). What sort of tone language is Mende? *Studies in African Linguistics*, 9.

Eisner, J. (1997a). Efficient generation in primitive Optimality Theory. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 313–320, Madrid.

Eisner, J. (1997b). What constraints should OT allow? Talk handout, Linguistic Society of America, Chicago. ROA#204-0797. Available at http://roa.rutgers.edu/.

Enderton, H. (1972). *A mathematical introduction to logic*. Academic Press.

Engelfriet, J. and Hoogeboom, H. J. (2001). MSO definable string transductions and two-way finite-state transducers. *ACM Transations on Computational Logic*, 2:216–254.

Engelfriet, J. and Vereijken, J. J. (1997). Context-free graph grammars and concatenation of graphs. *Acta Informatica*, 34:773–803.

Ferreira, R. (2013). *Efficiently Listing Combinatorial Patterns in Graphs*. PhD thesis, Università degli Studi di Pisa.

Gafos, A. (1996). *The articulatory basis of locality in phonology*. PhD thesis, Johns Hopkins University.

García, P., Vidal, E., and Oncina, J. (1990). Learning locally testable languages in the strict sense. In *Proceedings of the Workshop on Algorithmic Learning Theory*, pages 325–338.

Gerdemann, D. and Hulden, M. (2012). Practical finite state optimality theory. In *Proceedings of the 10th International Workshop on FSMNLP*, pages 10–19. ACL.

Gold, M. E. (1967). Language identification in the limit. *Information and Control*, 10:447–474.

Goldsmith, J. (1976). *Autosegmental Phonology*. PhD thesis, Massachussets Institute of Technology.

Graf, T. (2010a). Comparing incomparable frameworks: A model theoretic approach to phonology. *University of Pennsylvania Working Papers in Linguistics*, 16(2):Article 10.

Graf, T. (2010b). Logics of phonological reasoning. Master's thesis, University of California, Los Angeles.

Halle, M. (1978). Knowledge unlearned and untaught: What speakers know about the sounds of their language. In Halle, M., Bresnan, J., and Miller, G., editors, *Linguistic Theory and Psychological Reality*. MIT Press.

Halle, M. and Vergnaud, J. R. (1982). On the framework of Autosegmental Phonology. In van der Hulst, H. and Smith, N., editors, *The Structure of Phonological Representations (Part I)*. Foris, Dordrecht.

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

Hansson, G. O. (2001). *Theoretical and Typological Issues in Consonant Harmony*. PhD thesis, University of California, Berkeley.

Hansson, G. O. (2010). *Consonant harmony: Long-distance interaction in phonology.* Berkeley: University of California Press.

Haraguchi, S. (1977). *The Tone Pattern of Japanese: An Autosegmental Theory of Tonology.* Kaitakusha.

Harary, F. and Read, R. C. (1974). Is the null graph a pointless concept? In Bari, R. A. and Harary, F., editors, *Graphs and Combinatronics: Proceedings of the Capital Conference on Grpah Theory and Combinatronics at the George Washington University.* New York: Springer-Verlag.

Hayes, B., Kircher, R., and Steriade, D. (2004). *Phonetically based phonology.* Cambridge, UK: Cambridge University Press.

Heinz, J. (2007). *The Inductive Learning of Phonotactic Patterns.* PhD thesis, University of California, Los Angeles.

Heinz, J. (2009). On the role of locality in learning stress patterns. *Phonology*, 26:303–351.

Heinz, J. (2010a). Learning long-distance phonotactics. *Linguistic Inquiry*, 41:623–661.

Heinz, J. (2010b). String extension learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 897–906. Association for Computational Linguistics.

Heinz, J. (2011). Computational phonology part I: Foundations. *Language and Linguistics Compass*, 5(4):140–152.

Heinz, J., Rawal, C., and Tanner, H. G. (2011). Tier-based strictly local constraints for phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 58–64, Portland, Oregon, USA. Association for Computational Linguistics.

Heinz, J. and Rogers, J. (2010). Estimating strictly piecewise distributions. In *Proceedings of the 48th Annual Meeting of the ACL*. Association for Computational Linguistics.

Hewitt, M. and Prince, A. (1989). OCP, locality, and linking: the N. Karanga verb. In Fee, E. J. and Hunt, K., editors, *WCCFL 8*, pages 176–191. Stanford: CSLI Publications.

Hirayama, T. (1951). *Kyuusyuu hoogen Onchoo no Kenkyuu (Studies on the Tone of the Kyushu Dialects)*. Tokyo: Gakkai no shinshin-sha.

Hopcroft, J., Motwani, R., and Ullman, J. (2006). *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, third edition.

Hyman, L. (1987). Prosodic domains in Kukuya. *Natural Language & Linguistic Theory*, 5:311–333.

Hyman, L. (2011a). The representation of tone. In Oostendorp, M. V., Ewen, C. J., Hume, E., and Rice, K., editors, *The Blackwell Companion to Phonology*, volume II, chapter 45. Blackwell.

Hyman, L. (2011b). Tone: Is it different? In Goldsmith, J. A., Riggle, J., and Yu, A. C. L., editors, *The Blackwell Handbook of Phonological Theory*, pages 197–238. Wiley-Blackwell.

Hyman, L. (2014). How autosegmental is phonology? *The Linguistic Review*, 31:363–400.

Hyman, L. and Katamba, F. X. (2010). Tone, syntax and prosodic domains in Luganda. In Downing, L., Rialland, A., Beltzung, J.-M., Manus, S., Patin, C., and Riedel, K., editors, *Papers from the Workshop on Bantu Relative Clauses*, volume 53 of *ZAS Papers in Linguistics*, pages 69–98. ZAS Berlin.

Jardine, A. (2014). Logic and the generative power of Autosegmental Phonology. In Kingston, J., Moore-Cantwell, C., Pater, J., and Staubs, R., editors, *Supplemental proceedings of the 2013 Meeting on Phonology (UMass Amherst)*, Proceedings of the Annual Meetings on Phonology. LSA.

Jardine, A. (to appear). Computationally, tone is different. *Phonology*.

Jardine, A., Chandlee, J., Eyraud, R., and Heinz, J. (2014). Very efficient learning of structured classes of subsequential functions from positive data. In *Proceedings of the 12th International Conference on Grammatical Inference (ICGI 2014)*, JMLR Workshop Proceedings, pages 94–108.

Jardine, A. and Heinz, J. (2015). A concatenation operation to derive autosegmental graphs. In *Proceedings of the 14th Meeting on the Mathematics of Language (MoL 2015)*, pages 139–151, Chicago, USA. Association for Computational Linguistics.

Jardine, A. and Heinz, J. (2016). Learning tier-based strictly 2-local languages. *Transactions of the Association for Computational Linguistics*, 4:87–98.

Jardine, A. and Heinz, J. (in press). Markedness constraints are negative: an autosegmental constraint definition language. In *Proceedings of the 51st Annual Meeting of the Chicago Linguistics Society (CLS 2015)*.

Kager, R. (1999). *Optimality Theory*. Cambridge University Press.

Kay, M. (1987). Nonconcatenative finite-state morphology. In *Proceedings, Third Meeting of the European Chapter of the Association for Computational Linguistics*, pages 2–10.

Kiparsky, P. (2000). Opacity and cyclicity. *Linguistic Review*, 17:351–366.

Kisseberth, C. and Odden, D. (2003). Tone. In Nurse, D. and Philippson, G., editors, *The Bantu Languages*. New York: Routledge.

Kobayashi, Y. (1970). Accent in the Hirosaki dialect of Japanese. Master's thesis, The Ohio State University.

Konoshima, M. (1961). Hōgen no zittai to kyōtsūgoka no mondaiten—Aomori [the Aomori Dialect—Its reality and the problem of standardization]. In Tōjō, M., editor, *Hōgengaku Kōza [A course on dialectology]*, volume 2, pages 127–48. Tōkyōdō.

Kornai, A. (1991). *Formal Phonology*. PhD thesis, Stanford University.

Kornai, A. (1995). *Formal Phonology*. Garland Publication.

Koskenniemi, K. (1983). *Two-level morphology: a general computational model for word-form recognition and production*. Number 11. Helsinki: University of Helsinki Department of General Linguistics.

Kubozono, H. (2011a). Japanese pitch accent. In van Oostendorp, M., Ewen, C., Hume, E., and Rice, K., editors, *The Blackwell Companion to Phonology*. Oxford: Blackwell.

Kubozono, H. (2011b). Kikaijima nanbu, chuubu chiiki no akusento [The accent of southern and central Kikai-jima]. In *General Study for Research and Conservation of Endangered Dialects in Japan: Research Report on the Kikaijima Dialects*, pages 51–70. Tokyo: National Institute for Japanese Language and Linguistics.

Kubozono, H. (2012). Varieties of pitch accent systems in Japanese. *Lingua*, 122:1395–1414.

Lai, R. (2015). Learnable versus unlearnable harmony patterns. *Linguistic Inquiry*, 46:425–451.

Leben, W. R. (1973). *Suprasegmental phonology*. PhD thesis, Massachussets Institute of Technology.

Leben, W. R. (1978). The representation of tone. In Fromkin, V., editor, *Tone—A Linguistic Survey*, pages 177–219. Academic Press.

Leben, W. R. (2006). Rethinking autosegmental phonology. In Mugane, J., editor, *Selected Proceedings of the 35th Annual Conference on African Linguistics*, pages 1–9. MA: Cascilla Proceedings Project.

Li, W.-L. D. (2014). Prosodic structures of different Japanese dialects and the universality of the syllable. Master's thesis, University of Amsterdam.

Liberman, M. and Prince, A. (1977). On stress and linguistic rhythm. *Linguistic Inquiry*, 8:249–336.

Marlo, M. R. (2007). *The verbal tonology of Lunyala and Lumarachi: two dialects of Luluyia (Bantu, J.30, Kenya)*. PhD thesis, University of Michigan.

McCarthy, J. (2000). Harmonic serialism and parallelism. In Hirotani, M., editor, *NELS 30*, pages 501–524. Amherst, MA: GLSA Publications.

McCarthy, J. (2010a). Autosegmental spreading in Optimality Theory. In Goldsmith, J. A., Hume, E., and Wezels, W. L., editors, *Tones and Features: Phonetic and Phonological Perspectives*, pages 195–222. De Gruyer Mouton.

McCarthy, J. and Prince, A. (1993). Generalized alignment. Ms., University of Massachusetts, Amherst, and Rutgers University.

McCarthy, J. and Prince, A. (1995). Faithfulness and reduplicative identity. In Beckman, J., Dickey, L. W., and Urbanczyk, S., editors, *Papers in Optimality Theory*, number 18 in University of Massuchusetts Occasional Papers in Linguistics, pages 249—384. University of Massachusetts.

McCarthy, J. J. (1979). *Formal Problems in Semitic Phonology and Morphology*. PhD thesis, Massachusetts Institute of Technology.

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

McCarthy, J. J. (1989). Linear order in phonological representation. *Linguistic Inquiry*, 20(1):71–99.

McCarthy, J. J. (2003). Ot constraints are categorical. *Phonology*, 20:75–138.

McCarthy, J. J. (2010b). An introduction to harmonic serialism. *Language and Linguistics Compass*, 4(10):1001–1018.

McMullin, K. and Hansson, G. O. (to appear). Long-distance phonotactics as Tier-Based Strictly 2-Local languages. In *Proceedings of the Annual Meeting on Phonology 2015*.

McNaughton, R. and Papert, S. (1971). *Counter-Free Automata*. MIT Press.

Meyers, S. (1987). *Tone and the structure of words in Shona*. PhD thesis, University of Massachusetts, Amherst.

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

Nevins, A. (2010). *Locality in Vowel Harmony*. Number 55 in Linguistic Inquiry Monographs. MIT Press.

Newman, P. (1986). Tone and affixation in Hausa. *Studies in African Linguistics*, 17(3).

Newman, P. (2000). *The Hausa Language: An encyclopedic reference grammar*. New Haven: Yale University Press.

Odden, D. (1982). Tonal phenomena in Kishambaa. *Studies in African Linguistics*, 13(2):177–208.

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

Odden, D. (1994). Adjacency parameters in phonology. *Language*, 70(2):289–330.

Odden, D. (2013). *Introducing Phonology*. Cambridge University Press. Originally published 2005.

Partee, B., ter Meulen, A., and Wall, R. (1993). *Mathematical Methods in LInguistics*. Dordrect, Boston, London: Kluwer Academic Publishers.

Pater, J. (2004). Austronesian nasal substitution and other NC effects. In McCarthy, J., editor, *Optimality Theory in Phonology: A Reader*, pages 271–289. Oxford and Malden, MA: Blackwell.

Paulian, C. (1974). *Le Kukuya: langue teke du Congo*. Paris: S.E.L.A.F.

Potts, C. and Pullum, G. K. (2002). Model theory and the content of OT constraints. *Phonology*, 19:361–393.

Prince, A. and Smolensky, P. (1993). Optimality Theory: Constraint interaction in generative grammar. *Rutgers University Center for Cognitive Science Technical Report*, 2.

Prince, A. and Smolensky, P. (2004). *Optimality Theory: Constraint Interaction in Generative Grammar*. Blackwell Publishing.

Pulleyblank, D. (1986). *Tone in Lexical Phonology*. Dordrecht: D. Reidel.

Riggle, J. (2004). *Generation, Recognition, and Learning in Finite State Optimality Theory*. PhD thesis, UCLA.

Rogers, J. (1998). *A Descriptive Approach to Language-Theoretic Complexity*. Center for the Study of Language and Information.

Rogers, J., Heinz, J., Fero, M., Hurst, J., Lambert, D., and Wibel, S. (2013). Cognitive and sub-regular complexity. In *Formal Grammar*, volume 8036 of *Lecture Notes in Computer Science*, pages 90–108. Springer.

Rogers, J. and Pullum, G. (2011). Aural pattern recognition experiments and the subregular hierarchy. *Journal of Logic, Language and Information*, 20:329–342.

Sagey, E. (1986). *The Representation of Features and Relations in Non-Linear Phonology*. PhD thesis, Massachusetts Institute of Technology.

Shih, S. and Inkelas, S. (2014). A subsegmental correspondence approach to contour tone (dis)harmony patterns. In Kingston, J., Moore-Cantwell, C., Pater, J., and Staubs, R., editors, *Proceedings of the 2013 Meeting on Phonology (UMass Amherst)*, Proceedings of the Annual Meetings on Phonology. LSA.

Steriade, D. (1987). Locality conditions and feature geometry. *North Eastern Linguistic Society*, 18:595–618.

Thomas, W. (1982). Classifying regular events in symbolic logic. *Journal of Computer and Systems Sciences*, 25:360—376.

Topintzi, N. (2003). Issues of locality and morphology in the N. Karanga assertive and non-assertive patterns. *UCL Working Papers in Linguistics*, 15:327–354.

Uwano, Z. (1999). Classification of Japanese accent systems. In Kaji, S., editor, *Proceedings of the Symposium 'Cross-Linguistic Studies on Tonal Phenomena, Tonogenesis, Typology, and Related Topics*, pages 151–186. ILCAA.

Uwano, Z. (2012). Accent in some Kikai-jima dialects of Ryukyuan with particular reference to nouns in central and southern dialects. *Gengo Kenkyu [Journal of the Linguistic Society of Japan]*, 142:45—75.

Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142.

Walker, R. (2011). *Vowel patterns in language*. Cambridge: Cambridge University Press.

Walker, R. (2014). Nonlocal trigger-target relations. *Linguistic Inquiry*, 45:501–523.

Wiebe, B. (1992). Modelling autosegmental phonology with multi-tape finite state transducers. Master's thesis, Simon Fraser University.

Williams, E. S. (1971). Underlying tone in Margi and Igbo. *Linguistic Inquiry*, 7:463–484. Manuscript written 1971, LI version published 1976.

Yip, M. (1988a). The Obligatory Contour Principle and phonological rules: a loss of identity. *Linguistic Inquiry*, 19:65–100.

Yip, M. (1988b). Template morphology and the direction of association. *Natural Language and Linguistic Theory*, 6:551–577.

Yip, M. (2002). *Tone*. Cambridge University Press.

Zhang, J. (2000). The phonetic basis for tonal melody mapping. In Billerey and Lillehaugen, editors, *WCCFL 19 Proceedings*, pages 603–616. Also available as ROA 451.

Zoll, C. (2003). Optimal tone mapping. *LI*, 34(2):225–268.