University of California
Los Angeles

Inductive Learning of Phonotactic Patterns

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Linguistics

by

Jeffrey Nicholas Heinz

2007

The dissertation of Jeffrey Nicholas Heinz is approved.

Bruce Hayes

D. Stott Parker

Colin Wilson

Kie Zuraw, Committee Co-chair

Edward P. Stabler, Committee Co-chair

University of California, Los Angeles

2007

To Mika

Table of Contents

List of Figures

List of Tables

List of Algorithms

Acknowledgments

Ed Stabler's teachings, perspective, and kindness have inspired and contributed to this work and to my thought processes in uncountable ways. It is a privilege, honor, and pleasure to be his student. Kie Zuraw's unselfish support, generosity, critical eye, and linguistic savvy have improved this dissertation and my linguistic abilities significantly. Bruce Hayes and Colin Wilson are inspirational—I only hope this dissertation does some justice to their high scientific standards and eloquence. Their criticisms and suggestions have made this dissertation much, much better. I am also grateful for Stott Parker's insightful and thoughtful comments. In short, I cannot thank the members of my committee enough for their contributions to this dissertation and to my intellectual development. I assume responsibility for errors or flaws in either.

This dissertation was written with the support of a UCLA Dissertation Year Fellowship and I thank the university for providing such support.

There are many, many people who have contributed to my growth these many years at UCLA. It is in fact impossible to thank everyone who was here with me, even though they they have all contributed to the vitality of the department and the atmosphere that feeds the research process. To those I omit, please do not think of ill of me.

Ed Keenan, Marcus Kracht, Pam Munro, and Russell Schuh are heroes to me, each in their own way, and I thank them for their support and good cheer throughout these many years. Jason Riggle played a key role in influencing the direction of my research, as well as Greg Kobele. These two characters continue to inspire and motivate me to aim higher. I thank Sophie Streeter for her friendship these many years and for teaching me Kwara'ae. I also thank Andy Martin for his friendship, support and many insightful discussions. Likewise, it has been my good fortune to share invaluable times with Norm Apter, Peter Calabrese, Dieter Gunkel, Ingvar Lofstedt, Katya Pertsova, Sarah Van Wagnenen, and Shabnam Shademan. Additionally, where would I be without Leston Buell, Sarah Churng, Asia Furmanska, Vincent Homer, Tomoko Ishizuka, Sun-ah Jun, Ben Keil, Sameer Khan, Christina Kim, Ji Eun Kim, Nick LaCasse, Tatiana Libman, Ananda Lima, Ying Lin, Chacha Mwita, Kuniko Nielsen, Reiko Okabe, Michael Pan, Kevin Ryan, David Schueler, Carson Schütze, Rachel Schwartz, Molly Shilman, Tim Stowell, Megha Sundara, Harold Torrance, Stephen Tran, Heather Wilson, and Kristine Yu? Thank you.

I thank Al Bradley, Mary Crawford, and Betty Price for taking a chance on me in the spring of 2001. It was because of them I came to the UCLA campus, and was subsequently re-exposed to the world of academia. I also thank the professors I had as an undergraduate at the University of Maryland: Michael Brin, Norbert Hornstein, David Lebeaux, David Lightfoot, Mark Turner, Juan Uriagereka, James Yorke, and especially Linda Lombardi—my first phonology teacher—for nurturing my interest in linguistics, mathematics, and cognitive science so many years ago.

I sincerely thank Takahito and Harumi Mochizuki and Yuka, Mimi, and Katsumi Sugita for their continual encouragement and support.

I thank my parents, Eric and Loreen Heinz, for all they have taught me, from which everything was made possible, as well as for their love and support. I also thank my brother William Heinz and his wife Nancy Lawson their love and support these many, many years.

Finally, this dissertation could not have been written without the daily sunshine from Mika Mochizuki, who I love and admire. Our daughter Emma also brought smiles to my face on the days I thought nothing could break through thick fog. Thank you both, from the bottom of my heart.

Vita

1974	Born, Columbia, Maryland.
1992	Graduated, Atholton High School, Columbia, Maryland.
1996	B.A. Linguistics, University of Maryland, College Park.
1996	B.S. Mathematics, University of Maryland, College Park.
1997-1999	Mathematics Teacher, Peace Corps, Solomon Islands.
2000-2001	English Teacher, English Plus, Japan.
2001-2002	Student Affairs Officer, UCLA Music Department, California.
2005	M.A. Linguistics, University of California, Los Angeles.

Publications

Jeffrey Heinz. 2006. Learning Quantity Insensitive Stress Systems via Local Inference. *Proceedings of the Eighth Meeting of the ACL Special Interest Group in Computational Phonology at HLT-NAACL*, 21-30. New York, New York.

Abstract of the Dissertation

Inductive Learning of Phonotactic Patterns

by

Jeffrey Nicholas Heinz

Doctor of Philosophy in Linguistics

University of California, Los Angeles, 2007

Professor Edward P. Stabler, Co-chair

Professor Kie Zuraw, Co-chair

This dissertation demonstrates that significant classes of phonotactic patterns—patterns found over contiguous sounds, patterns found over non-contiguous segments (i.e. long distance agreement), and stress patterns—belong to small subsets of logically possible patterns whose defining properties naturally provide inductive principles learners can use to generalize correctly from limited experience.

This result is obtained by studying the hypothesis spaces different formulations of locality in phonology naturally define in the realm of regular languages, that is, those patterns describable with finite state machines. Locality expressed as contiguity (adjacency) restrictions provides the basis for n-gram-based patterns which describe phonotactic patterns over contiguous segments. Locality expressed as precedence—where distance between segments is not measured at all—defines a hypothesis space for long distance agreement patterns. Finally, both of these formulations of locality are shown to be subsumed by a more general formulation—that each relevant phonological environment is defined 'locally' and is unique—which I call *neighborhood-distinctness*.

In addition to patterns over contiguous and non-contiguous segments, it is shown that all stress patterns described in recent comprehensive typologies are, for small neighborhoods, neighborhood-distinct. In fact, it is shown that 414 out of the 422 languages in the typologies have stress patterns which are neighborhood-distinct for even smaller neighborhoods called '1-1'. Furthermore, it is shown that significant classes of logically possible unattested patterns do not. Thus, 1-1 neighborhood-distinctness is hypothesized to be a *universal* property of phonotactic patterns, a hypothesis confirmed for all but a few stress patterns which merit further study.

It is shown that there are learners which provably learn these hypothesis spaces in the sense of Gold (1967) and which exemplify two general classes of learners : string extension and state merging. Thus the results obtained here provide techniques which allow other hypothesis spaces possibly relevant to phonology, or other cognitive domains, to be explored. Also, the hypothesis spaces and learning procedures developed here provide a basis which can be enriched with additional, substantive phonological structure. Finally, this basis is readily transferable into a variety of statistical learning procedures.

CHAPTER 1

Introduction

> Many of the things you can count, don't count. Many of the things you can't count, really count. Albert Einstein

1 Thesis

The central thesis of this dissertation is that properties of natural language are directly related to properties of the learner and vice versa. In other words, the learning mechanism itself constrains the available hypothesis space in a nontrivial way. Because on one reading, this hypothesis is logically necessary, it is possible to misread this hypothesis as trivial. However, once we get beyond the logical necessity of a restricted hypothesis space, the strength of the thesis becomes clear:

(1) *Properties of the learning mechanism explain patterns found in natural language.*

The learner, and the class of patterns to be learned are in an intimate, not accidental or superficial, relationship.

It is easy to underestimate the significance of this thesis, as it is often taken for granted that this must be the case. However, as explained in §2 below, the theories in which natural language grammar learning has been most studied—in particular, the Principles and Parameters (Chomsky 1981) and Optimality Theory

(Prince and Smolensky 1993, 2004) frameworks—do not adopt this view. There, the proposed learning mechanisms (see below) operate over an additional layer of structure disconnected from any inherent properties of the hypothesis space. Thus, such learners are not tightly correlated to the the target class of patterns.

It follows from (1) that different classes of patterns are expected to have different learners. Learners of phonological patterns should not be the same as learners of syntactic patterns insofar as the two classes of patterns are not the same. Under this perspective, the human language learner is some combination of individual learners for specific domains (or even subdomains), e.g. phonetic categorization, word segmentation, phonology, morphology, syntax, and so on.

This dissertation explores the hypothesis in (1) in the domain of phonotactic patterns, which are the rules and constraints that determine which sequences of sounds are well-formed in a language. (Phonotactic patterns are explained in more detail in Chapter 2). In particular, this dissertation demonstrates that significant classes of phonotactic patterns—patterns found over contiguous sounds, and patterns found over non-contiguous segments (i.e. long distance agreement)—belong to small subsets of logically possible patterns whose defining properties naturally provide inductive principles learners can use to generalize correctly from limited experience. Strikingly, the defining properties of these hypothesis spaces directly relate to the notion of locality in phonology.

1.1 Locality and Learning

Locality has long been noted as key feature of phonological grammars. For example, McCarthy and Prince (1986:1) write "Consider first the role of counting in grammar. How long may a count run? General considerations of locality, . . . suggest that the answer is probably 'up to two': a rule may fix on one specified element and examine a structurally adjacent element and no other." Similarly Kenstowicz (1994:597) call this "...the well-established generalization that linguistic rules do not count beyond two ..." Also, in their Essay on Stress, (Halle and Vergnaud 1987:ix) also comment on the special role played by locality: "...it was felt that phonological processes are essentially local and that all cases of nonlocality should derive from universal properties of rule application." This is just a small sample of the research that has paid close attention to locality in phonology. In this dissertation, I thus ask: What contribution can this "well-established generalization" that "rules do not count beyond two" make to learning phonotactic patterns?

This dissertation answers this question by studying different formulations of locality in phonology. Locality expressed as contiguity (adjacency) restrictions provides the basis for n-gram-based patterns which describe patterns over contiguous segments. Locality expressed as precedence—where distance between segments is not measured at all—describe long distance agreement patterns. Finally, both of these formulations of locality are shown to belong to a more general formulation of locality—that each relevant phonological environment is 'local' and unique—which I call *neighborhood-distinctness* (to be defined in Chapter 5). It is shown that these properties naturally provide inductive principles learners can use to generalize in the right way from limited experience. It is also shown how a learner who generalizes using neighborhood-distinctness is in a sense *unable to count past two*, and can learn the kinds of phonotactic patterns found in the world's languages. In other words, it is shown that significant classes of attested patterns are neighborhood-distinct and significant classes of unattested patterns are not. Thus, neighborhood-distinctness is hypothesized to be a *universal* property of phonotactic patterns.

Although the phonotactic patterns investigated here are all neighborhood-distinct, this property itself is not necessarily sufficient for learning each class of phonotactic patterns. The additional properties of contiguity and precedence, discussed in chapter 3 and 4, provide additional inductive principles (which again limit the role of counting), which make learning these other classes of phonotactic patterns easier. Again, this follows from the thesis in (1), which expects different classes of patterns to have different kinds of learners.

1.2 Factoring the Learning Problem

Though this work focuses exclusively on the contribution particular notions of locality make to learning, there can be little doubt that many factors play a role in language acquisition by human children. Physiological, sociolinguistic, articulatory, perceptual, phonological, syntactic, and semantic factors are just a few of the ones which influence a child's acquisition of a grammar (including her phonotactic grammar). Given the complexity of human language and the complexity of a human child, it is likely that these factors, along with others, interact in complex ways. The methodological principle employed here is that the learning problem in linguistics is best approached by factoring—that is, by isolating particular inductive principles and studying how they allow language learners to generalize from their experience to particular language patterns in the right way (if at all).

In phonotactic learning, the role of substantive universals is certainly of particular interest. Although not uncontroversial, the hypothesis that the basic unit of a phonological grammar is the phonological feature, and not the segment, is widely accepted and may be considered a 'substantive' universal, though this notion has never been made quite clear (Jakobson et al. 1952). Thus it may come as a surprise to see the hypothesis spaces described in later chapters make little reference to phonological features. This is not because I reject phonological features or because I think they have no place in phonotactic learning. This is because the hypothesis spaces defined in later chapters follow from proposed formal universals of phonotactic patterns and are consequently best understood without the additional structure introduced by a feature system. Consequently, the symbols constituting the patterns under review are typically assigned an interpretation as a segment only in order to be concrete; in fact, those symbols can represent anything such as (underspecified) feature bundles, or some other rich phonological abstract entity. Despite recent rhetoric to the contrary (Hale and Reiss 2000, Port and Leary 2005), formal and substantive universals are not in conflict; they are in fact compatible.[1] (For further discussion of formal and substantive universals, see Chomsky (1965:9).[2])

In sum, this dissertation only makes clear the contribution of certain individual inductive principles relevant to locality. To do so, it abstracts away from other factors, which may concern those who find such factors relevant or important. Any time such abstractions are made, the picture of learning may be simplified and made less realistic. There is always the danger that if too many such abstractions are made, the resulting problem is trivial and uninteresting. However, as explained in Chapter 2, the abstractions made within these pages do not lead us to a trivialization of the learning problem. It is my belief that the more realistic, complicated picture of language learning will not be solved until we obtain a clear understanding of how learning can occur in simpler, nontrivial scenarios.

2 Other Approaches to Phonotactic Learning

There are at least two ways one can tell to what extent a learning algorithm is independent of the class of patterns it is trying to learn. The first is to see whether the same learning algorithm can be used in an entirely different domain. Recall that if the thesis in (1) is correct, we expect different patterns to have different learners. Thus if the same learning algorithm succeeds in two very different hypothesis spaces,

[1] What Port and Leary (2005) call ‘symboloids’ can be taken as proxies for symbols, for example.

[2] See also Mielke (2004), Lin (2005), Lin and Mielke (2007) for studies of how phonological features and segments might be learned from the acoustic signal.

then the learning algorithm itself tells us little about the nature of either hypothesis space.

The second way is to imagine a martian endowed with complete knowledge of the learning mechanism used by human children in some linguistic domain, such as phonotactic patterns. The extent to which the martian can determine, on the basis of this knowledge, the kinds of phonotactic patterns that exist tells us to what extent the learning mechanism is divorced from the hypothesis space. If the martian is unable to deduce anything about the range of possible patterns from its knowledge of the learning algorithm, then the learning algorithm is completely independent of the range of possible patterns. On the other hand, if the martian now knows something about the range of possible patterns, then the learning algorithm shapes the hypothesis space to some extent. The thesis in (1) follows from my belief that a martian who knew how phonotactic patterns are learned would in fact be able to deduce a great deal (if not everything) about the character of possible phonotactic patterns.

Based on these two diagnostics, earlier research on learning phonotactic grammars does not advance the thesis in (1). In the Principles and Parameters framework (Chomsky 1981), the properties of the proposed learners are not tightly correlated with properties of the patterns to be learned. Likewise, learners in the Optimality Theoretic framework (Prince and Smolensky 1993, 2004) operate completely independently of the content of the constraints, which determine the possible patterns. Connectionist learning models do not reveal which of their architectural properties allow which natural language patterns to be learned. To date, statistical learning models, such Bayesian models or maximum entropy models, focus primarily on searching a given hypothesis space as opposed to molding the shape itself. These points are elaborated on below.

2.1 Learning with Principles and Parameters

The Principles and Parameters framework (Chomsky 1981) (henceforth P&P) maintains that there exists a set of a priori parameters whose values determine possible human grammars. The learner's task is to set the correct values for the parameters of the grammar of the language being acquired. One influential learning algorithm in this framework is the Triggering Learning Algorithm (TLA) (Gibson and Wexler 1994), which sets parameter values according to 'triggers' that the learner observes in the linguistic environment. Others comment on different aspects of this model (Niyogi and Berwick 1996, Frank and Kapur 1996) (see also Niyogi (2006)), but Dresher (1999) comments on what I consider the least compelling property of Triggering Learning Algorithm:

> ...at the most general level...the learning algorithm is independent of the content of the grammar. ...for example, ...it makes no difference to the TLA what the content of a parameter is: the same chart serves for syntactic word order parameters as for parameters of metrical theory, or even for nonlinguistic parameters.

In other words, the triggers—which can be individual words, or sentences, or some information gleaned from them such as word order—can be related to any arbitrary parameter. The learning algorithm essentially consists of statements like "On observing Trigger A, set Parameter B to true." There need not be any relation at all between the trigger A, the parameter B, and the significance of B being set to the value true. Thus the TLA is an appropriate learning algorithm for any parameterized domain, linguistic or otherwise.

Dresher (1999) draws a distinction between the TLA and the ordered cue learning model of Dresher and Kaye (1990), which uses cues in the observed linguistic environment to set parameters instead of triggers. Dresher (1999:28) explains that, unlike triggers in the TLA model, "Cues must be appropriate to their parameters in the sense that the cue must reflect a fundamental property of the parameter, rather than being fortuitously related to it." The learner comes equipped with a priori cues whose content relates nontrivially to the content of the parameters. Thus in principle, the ordered cue based learner for syntactic patterns is different from the ordered cue based learner for phonological one because the content of the cues is different.

This is a step in the right direction, but neither Dresher and Kaye (1990) nor Dresher (1999) offer a precise explanation of what a "fundamental property" of a parameter would look like, or what properties of an associated cue make it appropriate. Thus it is not exactly clear how different the ordered cue based learner is from the TLA in this respect (see Gillis et al. (1995) for some discussion).

2.2 Learning with Optimality Theory

Optimality Theory (henceforth OT) is a theory of grammar where the range of possible grammars is determined solely by the ranking of an a priori finite set of constraints (Prince and Smolensky 1993, 2004). How learning can take place in this framework has been the subject of a number of studies, notably Tesar (1995, 1998), and Tesar and Smolensky (1998, 2000) but see also Boersma (1997), Pulleyblank and Turkel (1998), Hayes (1999), Boersma and Hayes (2001), Lin (2002), Pater and Tessier (2003), Pater (2004), Prince and Tesar (2004), Hayes (2004), Riggle (2004), Alderete et al. (2005), Merchant and Tesar (2006), Wilson (2006b), Riggle (2006), Tessier (2006).

OT learners, essentially characterized by Recursive Constraint Demotion (RCD) (Tesar and Smolensky 2000), take advantage of the structure afforded by OT grammars over some hypothesis space, but no advantage of the inherent nature of the hypothesis space itself. Indeed, Tesar and Smolensky (2000:7-8) write that

> OT is a theory of UG that provides sufficient structure at the level of the grammatical framework itself to allow general but grammatically informed learning algorithms to be defined... Yet the structure that makes these algorithms possible is not the structure of a theory of stress, nor a theory of phonology: it is the structure defining any OT grammar...

Thus, OT learners apply to any domain equally provided that domain can be described with a finite number of rankable constraints and the notion of strict domination. It follows that the content of universal constraint set is divorced entirely from the learning process. This is why Dresher (1999) directs his comments (quoted above) not only to the TLA, but also to the learning algorithms proposed in an OT setting.

Another way to understand this is to recognize that most descriptions of RCD make no reference to specific constraints, preferring instead to use constraints C_1, C_2 and so on which refer to any possible constraint (see, for example, Kager (1999)). Understanding the crucial aspects of RCD requires no knowledge of the content of the constraints. This is often taken as an advantageous aspect of the theory. RCD and its variants apply to learning syntactic grammars in the same way they apply to learning phonological grammars. There is no difference between how such grammars can be learned despite the fact that no one thinks the same kind of patterns are found in the syntactic domain as in the phonological domain. Consequently, the martian who knows the RCD algorithm cannot determine anything about the class of patterns to be learned. This is because the range of possible grammars is determined completely by the content of the constraints, which themselves are completely independent of the proposed learning algorithms.

Variants of RCD such as such as Biased Constraint Demotion, the Gradual Learning Algorithm (Boersma 1997, 1998, Boersma and Hayes 2001), and the Luce Choice Ranker (Wilson 2006b) and others (Pater and Tessier 2003, Pater 2004, Alderete et al. 2005, Merchant and Tesar 2006, Tessier 2006, Riggle 2006) offer proposals which modify aspects of RCD (or OT), but none of them address the concerns stated here. To do so would require either learners which make reference to the content of the constraints (so the same learner would not succeed with a different universal constraint set) or learners that at least discover certain kinds of constraints (see, for example, Ellison (1992, 1994b), Goldsmith (2006), and Hayes and Wilson (to appear)).

2.3 Learning with Connectionist Models

Connectionist models differ significantly from the symbolic approaches above. The connectionist model is a network of nodes and weighted connections, whose architecture determines the properties of the system. Connectionist models have been successful in modeling various aspects of phonology (Rumelhart and McClelland 1986, Goldsmith and Larson 1990, Gupta and Touretzky 1994, Shillcock et al. 1993, Christiansen et al. 1998) (see Elman (2003) for learning in other natural language domains).

Although the architecture does place limits on the kinds of pattern that can be learned (and ultimately circumscribe some hypothesis space), it is not known what this space looks like, or how it depends on the architecture. For example, altering the architecture of the network by adding or removing nodes or connections, may change results in a given learning setup. It is not clear how or why the results change in these cases. New techniques may render these models analytically accessible, but until then, it is not known how the architecture contributes to the shape of the hypothesis space.

2.4 Learning with Statistical Models

The general heading of learning with statistical models encompasses many different approaches: approaches based on OT, (Boersma 1997, Boersma and Hayes 2001), minimum distance length (Ellison 1994b), Bayes law (Tenenbaum 1999, Goldwater 2006), maximum entropy (Goldwater and Johnson 2003, Hayes and Wilson to appear) and approaches inspired by Darwinian-like processes (Clark 1992, Yang 2000). Advantages of models of these types is that they are robust in the presence of noise and are capable of handling variation.

Many of these models, including all of the ones cited above, are *structured* probabilistic models; i.e. they combine a hypothesis space informed by proposed linguistic structures with probabilities associated with those structures (see also Gildea and Jurafsky (1996), Albright and Hayes (2003b)). For example, Hayes and Wilson (to appear) use a maximum entropy method to discover constraints over a hypothesis space that is determined by trigrams, phonological features, projections, and metrical grids.

The learning models in these frameworks, however, are independent of the hypothesis space. This is not controversial. For example, Goldwater (2006:19) explains that "the focus of the Bayesian approach to cognitive modeling is on the probabilistic model itself, rather than on the specifics of the inference procedure." Yang (2000:22) describes one of the "virtues" of his approach this way: "UG provides the hypothesis space and statistical learning provides the mechanism." In other words, if UG provided some other hypothesis space, there would be no need to alter the statistical learning mechanism. This dissertation is primarily concerned with the shape of the hypothesis space as a consequence of the inference procedure, as opposed to a search within some given space. My opinion is not that probabilistic models are uninteresting or unimportant, but rather they have different goals such as handling input corrupted by noise or learning real-valued grammatical objects. It is my hope that the insights gained from the kinds of probabilistic models mentioned here will be integrated, in future research, with the kinds of inference procedures (and subsequent hypothesis spaces) that I present here.

2.5 Local Summary

The thesis in (1) states that properties of the learning algorithm explain properties of the hypothesis space. This is not a logical necessity, and perhaps it is false. It is perfectly possible that in human children the inherent properties of the hypothesis space and language-learning procedures are separate. This is the case for learners proposed in the P&P and OT frameworks, which divorce the inherent properties of the hypothesis space from the learner. In these frameworks, learning occurs because of a layer of structure provided by the framework (binary parameters or strictly ranked violable constraints) that exists independently of the inherent properties of the hypothesis space. Connectionist models, despite an array of results, are for the most part analytically unaccessible. Statistical approaches do not seek to mold a particular hypothesis space, so much as they aim to find an effective way of searching within it. Thus, earlier research fails to advance the hypothesis in (1).

3 Overview

In this chapter, I put forward a hypothesis that puts the learning process front and center in generative linguistics by claiming an intimate connection exists between the classes being learned and the learner. I argued that this approach has not been pursued in the dominant frameworks in generative linguistics today. I explore this hypothesis in subsequent chapters by proposing formal universals of phonotactic patterns over contiguous segments, patterns over non-contiguous segments, and stress patterns. There I show that these universals naturally define hypothesis spaces whose inherent structure naturally present inductive principles learners can use to generalize correctly from limited experience. Although these hypothesis spaces are independent of substantive universals which surely play a role in phonotactic learning, they are not incompatible with them, and I support research which seeks to develop formal, substantive, learning-based theories of phonology. Finally, although these hypothesis spaces and learners are discrete, this is also not a relevant feature of these spaces—they can be made gradient and are compatible with a variety of statistical learning techniques.

Chapter 2 formulates precisely the learning problem that the remaining chapters tackle. It introduces what is meant by phonotactic patterns in natural language, and review the kinds of phonotactic patterns discussed in the other chapters of the dissertation. It then explains the advantages of representing phonotactic patterns as regular sets (i.e. describable by finite-state acceptors) when addressing the learning problem. Finally, it provides a learning framework in which the proposed learners of later chapters can be studied, and describes a strategy for identifying these various learners.

Chapter 3 reviews a popular method for learning patterns over contiguous segments (n-gram models (Manning and Schütze 1999, Jurafsky and Martin 2000)). It shows how a categorical version of these n-gram models makes clear the hypothesis space upon which such models are based, and the relevant inductive principle used in learning patterns in this space. Furthermore, I demonstrate that this inductive principle can be instantiated in two general families of largely unexplored inductive principles, which I call *string extension learning* and *state-merging*.

Chapter 4 considers long distance agreement patterns like consonantal harmony and vowel harmony. It is shown that these patterns define a subset of the regular languages which I call *precedence languages*. It is shown that learning patterns in this class is simple because the learner only makes distinctions on the basis of precedence, and not distance. Like n-gram languages, equivalent learners for this class can be described as both string extension and state-merging learners (though it is easier to state it as a string extension learner).

Chapter 5 introduces the concept of neighborhood-distinctness with respect to stress patterns found in the world's languages. Intuitively, a pattern is neighborhood-distinct if the relevant phonological environments are unique. A typological survey shows that 107 of 109 stress patterns in the survey are neighborhood-distinct. It is also shown that many logically possible stress patterns are not-neighborhood distinct. Thus it is established that neighborhood distinctness approximates the attested patterns in a nontrivial way. The Forward Backward Learner (FBL) is presented which uses this property to make inferences from limited input and it is shown that it succeeds on 100 of the 109 patterns. Unlike the n-gram and precedence learners, the FBL is only stated in state-merging terms.

Chapter 6 examines in detail the neighborhood-distinct class of patterns, and the class of patterns learnable by the algorithm presented in Chapter 5. It is shown that trigram and precedence languages are neighborhood-distinct. Thus the notion of locality embodied in neighborhood-distinctness subsumes the notions of locality based on contiguity and precedence. I also lay out a strategy for developing a better understanding of the range of the FBL learning function based on the observation that neighborhood-distinctness is actually a composition of properties. This line of inquiry reveals other interesting learners and hypothesis spaces relevant to phonotactic learning.

Chapter 7 concludes the dissertation, with a few remarks about the goals, results, and further research.

With the exceptions of Chapters 5–7, most chapters in this dissertation contain appendices which are mathematical in character. I chose to relegate the mathematical foundations of the results presented in these chapters to appendices. Although the main ideas are communicated in the body of each chapter, this does not mean the mathematical appendices should be skipped. On the contrary, the results in those appendices are the best evidence in support of the hypothesis in (1). The extent to which the learners and the language classes they learn are intertwined becomes most apparent by studying their properties precisely. The appendices are separate only as an organizational aid. I have tried to make them as complete as possible; i.e. the appendices, in order, can take a willing individual with no background in mathematics to the ultimate results. This is possible primarily because there is no concept in these pages, mathematical or otherwise, that is not at its core, very simple.

Appendices

A–1 Mathematical Preliminaries

This appendix introduces the simplest mathematical concepts and the notations which express those concepts that are used throughout the mathematical appendices in subsequent chapters. Partee et al. (1993) cover much of the same basics with somewhat more context. Angluin (1982) gives a precise, very concise, clear introduction to some of these ideas as well.

A–1.1 Sets

A set is some (possibly empty) collection of elements. The empty set is denoted $\emptyset$. For any set S, let $|S|$ denote the cardinality of S. Given two sets A and B, A is a subset of B (written $A \subseteq B$) iff for all $a \in A$, $a \in B$. The set of natural numbers $0, 1, 2, \ldots$ is denoted $\mathbb{N}$. The following standard set operations are defined here.

$$\begin{array}{lccl} \text{(2)} \quad \text{union} & A \cup B & = & \{x : x \in A \text{ or } x \in B\} \\ \text{intersection} & A \cap B & = & \{x : x \in A \text{ and } x \in B\} \\ \text{difference} & A - B & = & \{x : x \in A \text{ and } x \notin B\} \\ \text{powerset} & 2^A & = & \{X : X \subseteq A\} \\ \text{product} & A \times B & = & \{(a,b) : a \in A \text{ and } b \in B\} \end{array}$$

Because union and intersection are commutative (e.g. $A \cup B = B \cup A$), it is easy to extend these operations over sets of sets. We use the symbols $\bigcup$ and $\bigcap$ to indicate the union and intersection of sets, respectively. For example if S is a collection of sets then $\bigcup S$ indicates the set of elements which belong to some set in S.

A–1.2 Relations and Partially Ordered Sets

A relation R between two sets A and B is a subset of $A \times B$. If $(x, x') \in R$, we often write xRx'. A relation $R \subseteq S \times S$ may have one or more of the following properties:

(3) antisymmetry Whenever xRy and yRx, it is the case that $x = y$.

reflexivity For all $x \in S$, it is the case that xRx.

irreflexivity For all $x \in S$, it is not the case that xRx.

symmetry Whenever xRy, it is the case that yRx.

asymmetry Whenever xRy, it is not the case that yRx.

transitivity Whenever xRy and yRz, it is the case that xRz.

A relation R on a set S is a *reflexive partial order* iff R is reflexive, transitive, and antisymmetric. If there is a reflexive partial order relation $\leq$ over a set S, we call S a *partially ordered set* and often write $(S, \leq)$.

Given some partially ordered set $(S, \leq)$, and some $x \in S$ and $T \subseteq S$, x is a *lower bound* for T iff for all $y \in T$, $x \leq y$. x is a *greatest lower bound* for T iff x is a lower bound for T and for all y such that y is a lower bound for T, $y \leq x$.

Similarly, x is an *upper bound* for T iff for all $y \in T$, $y \leq x$. x is a *least upper bound* for T iff x is an upper bound for T and for all y such that y is a upper bound for T, $x \leq y$.

A *lattice* is a partially ordered set $(S, \leq)$ such that for all $x, y \in S$, $\{x, y\}$ has a greatest lower bound and a least upper bound.

A–1.3 Equivalence Relations and Partitions

A set π of nonempty subsets of S is a *partition* of S iff the elements of π are pairwise disjoint (i.e. for any $A, B \in \pi$, $A \cap B = \emptyset$) and their union equals S (i.e. $\bigcup \pi = S$). The elements of π are called *blocks* of the partition. The *trivial partition* of S is the one where each block contains a single element of S. The *unit partition* is the partition which contains exactly one block (i.e. $\pi = \{S\}$).

Each block in partition π is identified by an element of $x \in S$ which is denoted $B(x, \pi)$. A partition π_0 *refines* partition π_1 iff every block in π_1 is a union of blocks of π_0. This is denoted $\pi_0 \leq \pi_1$. In this case we also say π_1 is *coarser* than π_0 and that π_0 is *finer* than π_1. The finest partition (the trivial partition) refines any partition of S and the coarsest partition (the unit partition) is the partition which only refines itself. Note that $\leq$ is reflexive partial order. Note further that the set of all possible partitions of S, denoted Π, forms a lattice structure under the $\leq$ relation (Grätzer 1979). In other words, there is a greatest lower bound and a least upper bound for every pair of partitions in Π.

If $S_0 \subseteq S$ then the *restriction* of π to S_0 is the partition π_0 which consists of all nonempty sets B which are are the intersection of S_0 and a block of π.

A relation R on a set S is an *equivalence relation* iff R is reflexive, symmetric, and transitive. An equivalence relation $\sim$ on a set S naturally induces a partition $\pi_\sim$ of S: for all $x, y \in S$, $B(x, \pi_\sim) = B(y, \pi_\sim)$ iff $x \sim y$.

A–1.4 Functions and Sequences

A function f is a relation between two sets A and B such that if $(a, b) \in f$ then $(a, b') \in f$ iff $b = b'$. We write $f(a) = b$ and call b the value of f at a. A is called the *domain* of f and B the *co-domain.* This is often indicated by writing $f : A \to B$.

A function $f : A \rightarrow B$ is a *total function* iff for all $a \in A$, there is a $b \in B$ such that $(a, b) \in f$. A function which is not total is *partial* and is undefined for those elements in A for which for all $b \in B$, $(a, b) \notin f$. A function $f : A \rightarrow B$ is called *one-to-one*, or *injective*, iff for distinct $a, a' \in A$, $f(a) \neq f(a')$. A function $f : A \rightarrow B$ is called *onto*, or *surjective*, iff for all $b \in B$, there is $a \in A$ such that $f(a) = b$. A function is a *bijection* iff it is one-to-one and onto. If $f : \mathbb{N} \rightarrow A$ is a bijection then the cardinality of A is said to be *countably infinite* and f is sometimes called an *enumeration* of A.

Given a function $f : X \rightarrow Y$, f naturally induces an equivalence relation $\sim_f$ over X: for all $x_1, x_2 \in X$, $x_1 \sim_f x_2$ iff $f(x_1) = f(x_2)$. We say in such a case that x_1 and x_2 are *f-equivalent.* Thus the function f, by way of the equivalence relation it induces, also induces a partition π_f over X such that $B(x_1, \pi_f) = B(x_2, \pi_f)$ iff $x_1 \sim_f x_2$ (i.e. iff $f(x_1) = f(x_2)$).

Given $f : A \rightarrow B$ and $g : B \rightarrow C$, the composition of f and g, denoted $f \circ g$, has domain A and co-domain C and is for all $a \in A$ is given by $g(f(a))$. Note that if $f(a)$ is undefined, then so is $f \circ g(a)$.

A countably infinite sequence of elements from a set A is given by a total function $\sigma : \mathbb{N} \rightarrow A$. The elements of the sequence are naturally thought of as $f(0), f(1), f(2), \ldots$. Similarly, a finite sequence of length k is given by $\sigma : \{0, 1, 2, \ldots k\} \rightarrow A$. A finite sequence σ of length $k + 1$ may be concatenated with another (not necesarily finite) sequence τ yielding the sequence υ where

1. $\upsilon(i) = \sigma(i)$ for all $i \leq k$

2. $\upsilon(i) = \tau(i - k - 1)$ for all $i > k$

We write $\upsilon = \sigma \diamond \tau$, or more simply, $\upsilon = \sigma\tau$. Sometimes the notation is relaxed and we write, for example, $\sigma \diamond f(2)$ to indicate we are appending an element at the end of a finite sequence.

A–1.5 Strings and Formal Languages

A string is a sequence. We often use Σ to denote a fixed finite set of symbols, the *alphabet*. Let Σ^n, $\Sigma^{\leq n}$, Σ^*, Σ^+ denote all strings formed over this alphabet of length n, of length less than or equal to n, of any finite length, and of any finite length strictly greater than zero, respectively. The codomain of a string is called the *range* and is the set of symbols which are in the string. The empty string is the unique string of length zero and is denoted by λ. Thus $range(\lambda) = \emptyset$. The length of a string u is denoted by $|u|$, e.g. $|\lambda| = 0$. The reverse of a string u is denoted u^r (note: $\lambda^r = \lambda$). Clearly, $(u^r)^r = u$. Let uv denote the concatenation of two strings u and v. A string u is a *prefix* of another string w iff there exists v in Σ^* such that $w = uv$. Similarly, u is a *suffix* of another string w iff there exists v in Σ^* such that $w = vu$.

A language L is some subset of Σ^*. The reverse of a language L, denoted by L^r, is the set of strings u^r where u is in L (so $(L^r)^r = L$). The concatenation of two languages L_1 and L_2, denoted $L_1 \cdot L_2$, is equal to $\{uv : u \in L_1 \text{ and } v \in L_2\}$. Let $L^k, L^{\leq k}$ denote all strings in L with length exactly k and with length less than or equal to k, respectively.

Definition 1 The *prefixes* of language L are given by

$$Pr(L) = \{u : \exists v \text{ so that } uv \in L\}$$

Definition 2 The left-quotient of language L and string w, or the *tails* of w given L, is denoted by

$$T_L(w) = \{u : wu \in L\}$$

Consequently, $T_L(w) \neq \emptyset$ iff $w \in Pr(L)$. Note also that for any $u \in Pr(L)$, $T_L^0(u) = \{\lambda\}$.

CHAPTER 2

Establishing the Problem and Line of Inquiry

> . . . language makes infinite employment . . . of finite means. . . –*Wilhelm von Humboldt (1836)*

> How comes it that human beings, whose contacts with the world are brief and personal and limited, are nevertheless able to know as much as they do know? –*Bertrand Russell (1935)*

> . . . if we are to be able to draw inferences from these data. . . we must know . . . principles of some kind by means of which such inferences can be drawn. –*Bertrand Russell (1912)*

1 Phonotactic Patterns and Phonotactic Knowledge

Phonotactic patterns are the patterns which govern the distribution of sounds in well-formed words in a language. By word, I mean some domain over which a phonotactic rule or constraint is said to apply. This dissertation assumes that such domains are known, and do not have to be discovered along with the phonotactic constraints or rules themselves.

Identifying the phonotactic rules or constraints, and the principles which underly them—that is, developing a theory which characterizes a speaker's knowledge of these patterns—has been one focus of generative phonology. This dissertation categorizes phonotactic patterns into three groups: patterns over contiguous segments, patterns over non-contiguous segments, and stress patterns over syllables. In the remainder of this section, I briefly review these three groups of patterns here and some evidence that the patterns are rule- or constraint-governed and that speakers know these rules and constraints.[1]

§2 explains why I choose to represent phonotactic grammars as regular sets—that is, with finite state machines. §3 makes explicit two learning frameworks, due to Gold (1967) and Valiant (1984), and shows why the simplifications made in this dissertation do not lead to a trivialization of the learning problem. Finally, §4 presents the general research strategy adopted in later chapters of this dissertation. §5 summarizes the main ideas of this chapter.

1.1 Patterns over Contiguous Segments

Many phonotactic patterns are stated as restrictions over sequences of adjacent sounds. This section provides a few examples of this kind of phonotactic pattern and presents the arguments linguists make that these constraints or rules are known by speakers.

Halle (1978:294) introduces these ideas this way:

> The native speaker of a language knows a great deal about his language that he was never taught. An example of this untaught knowledge is illustrated in (1), where I have listed a number of words chosen from different languages, including English. In order to make this a fair test,

[1] Of course there are other ways phonotactic patterns could be categorized. We might separate articulatorily motivated processes from perceptually motivated ones. We might separate processes which engage similar sounds from processes which engage dissimilar ones. Categorizing phonotactic patterns in these kinds of ways might lead us to discover other kinds of inductive principles plausibly active in human language learning. Although I do not pursue these other categorization schemes here, I encourage others to pursue them and their consequences for human language learning.

> the English words in the list are words that are unlikely to be familiar to the general public, including most crossword-puzzle fans:
>
> (1) ptak thole hlad plast sram mgla vlas flitch dnom rtut
>
> If one were to ask which of the ten words in this list are to be found in the unabridged Webster's, it is likely that readers of these lines would guess that *thole, plast,* and *flitch* are English words, whereas the rest are not English. This evidently gives rise to the question: How does a reader who has never seen any of the words on the list know that some are English and others are not? The answer is that the words judged not English have letter sequences not found in English. This implies that in learning the words of English the normal speaker acquires knowledge about the structure of words. The curious thing about this knowledge is that it is acquired although it is never taught, for English-speaking parents do not normally draw their children's attention to the fact that consonant sequences that begin English words are subject to certain restrictions that exclude words such as *ptak, sram,* and *rtut*, but allow *thole, flitch,* and *plast.* Nonetheless, in the absence of any overt teaching, speakers somehow acquire this knowledge.

Halle's first point is that this knowledge exists and can be characterized. If it did not exist, then it could not be applied in novel situations, as all English speakers do in his exercise above. A number of laboratory studies also demonstrate the same point (Greenberg and Jenkins 1964, Ohala and Ohala 1986, Pierrehumbert 1994, Coleman and Pierrehumbert 1997a, Vitevitch et al. 1997, Frisch et al. 2000, Treiman et al. 2000, Bailey and Hahn 2001, Hay et al. 2003, Hammond 2004).

The knowledge in Halle's example can be characterized at one level by listing the impermissible word-initial consonant clusters such as **pt,* **hl,* **sr,* **mg,* **vl,* **dn,*

**rt*. It may also be characterized in more general terms by identifying properties these consonant clusters share that separate them from well-formed word-initial consonant clusters. This section is not so much concerned with the right characterization of this knowledge as it with establishing the fact that English speakers have characterizable knowledge about which contiguous sequences of consonants are licit at the beginnings of words—knowledge which can be applied in novel situations. This is the meaning of Humboldt's expression in the leading quote of this chapter (which is often cited by Chomsky).

Halle's second point—that this knowledge was acquired without being taught—illustrates that speakers, based on their finite experience, have made generalizations that allow them to know things beyond their limited experience. How children do this is a deep mystery and relates to the question asked in the second quote leading this chapter by Bertrand Russell, although he is not referring to language per se.[2] The problem of language learning is the problem of induction: how one passes from true statements of one's experience (such as one's linguistic observations) to true statements about the nature of the world (or one's language).

The next example comes from Japanese, in which the only words with a sequence of two contiguous consonants (CC) are ones where the two consonants are identical (i.e. a geminate), or where the first consonant is a nasal.

The words in (1) are well-formed words in Japanese, but the (English) words in (2) are not.

[2] It is worth pointing out that the position of scientists trying to understand the rules of nature is the same as that of children trying to understand the rules of their language. There are in fact infinitely many hypotheses that are consistent with the (finite amount of) data observed at any given moment. Yet, both the scientist and the child appear to be able to generalize beyond this limited experience to make predictions about novel situations.

(1)	a.	kawaru	‘substitute’
	b.	ɡakkoː	‘school’
	c.	aŋko	‘red bean paste’

(2)	a.	tejbl̩	‘table’
	b.	klʌb	‘club’
	c.	krɪsməs	‘Christmas’

Evidence that Japanese speakers ‘know’ this constraint comes from word borrowing. When new words are borrowed into a language from other language, there is often a conflict between what is well-formed in the borrower’s language and what is well-formed in the loaner’s language. Often, this conflict is resolved so that the word in the loaner’s language is changed to conform to the well-formedness rules and constraints of the borrower’s language. When Japanese speakers borrow words into English, for example, the words are altered so as to conform to the constraints over contiguous sequences of sounds as shown in (3).

(3)	a.	teːburu	‘table’
	b.	kurabu	‘club’
	c.	kurisumasu	‘Christmas’

Kisseberth’s (1973) study of Yawelmani Yokuts presents an additional kind of evidence that speakers have knowledge of phonotactic constraints. In Yawelmani Yokuts well-formed words do not have contiguous sequences of three consonants. Evidence that speakers are aware of this constraint, which I denote *CCC, comes from adjustments made to the concatenation of morphemes which otherwise would result in three adjacent consonants.

For example, in certain circumstances, sequences of three contiguous consonants are broken up by deleting a consonant. Thus underlying /hall+hatin+iːn/ 'will desire to lift up' is not realized as *[hallhatiniːn], which has the illegal consonant sequence [llh]. Instead, speakers of Yawelmani Yokuts delete the [h] so that /hall+hatin+iːn/ is realized as *[hallatniːn] (there is also a vowel deletion for other reasons).

In other circumstances, a vowel is inserted to break up sequences of three contiguous consonants. Thus underlying /dːiyl+t/ 'was being guarded' is not realized as *[diːylt], a form which violates the phonotactic constraint. Rather, /diːyl+t/ is realized as [deːylit] (there is also vowel lowering for other reasons).

Kisseberth (1973) presents rules which predict where the consonantal deletion and vowel epenthesis occur. The fact that these rules have something in common—the elimination of potential contiguous sequences of three consonants—has been argued to be evidence that native speakers of Yawelmani Yokuts possess knowledge that can be characterized as *CCC.

This section presented three examples of phonotactic constraints over contiguous segments and evidence that such constraints are part of the speaker's linguistic competence. This evidence comes not only from observing a systematic, nonarbitrary distribution of sounds in the words in the language, but also from the speakers ability to apply the knowledge of this distribution in novel situations. The simplest kind of novel situation is simply to ask a speaker if a word which disobeys the phonotactic generalizations is well-formed, as compared to an otherwise identical word which obeys the phonotactic. Another kind of novel situation occurs when speakers borrow words from other languages. As in Japanese, native speakers alter words they take from other languages to conform to the phonotactic constraints present in their native language. Finally, the faithful concatenation of morphemes in a language can result in a sound sequence which disobeys a phonotactic constraint. Thus in many languages, as in Yawelmani Yokuts, the actual pronunciation of a concatenation of morphemes often deviates in regular ways from the faithful sequence so as to obey the otherwise violated phonotactic constraint.

1.2 Patterns over Non-contiguous Segments

Phonotactic patterns over noncontiguous segments are those which can be stated as restrictions over sequences of non-adjacent sounds. Consonantal harmony and disharmony cases present the clearest example of constraints of this type (Hansson 2001, Rose and Walker 2004). Whether vowel harmony patterns belong to this class is less clear because vowels can appear phonetically contiguous despite intervening consonants. Vowel harmony patterns and the relevant issues are discussed in greater detail in Chapter 4.

A classic example is Navajo sibilant harmony (Sapir and Hojier 1967, Fountain 1998). In well formed words in this language, sibilants agree in the feature [anterior]. At the segmental level this means that no words contains two segments with different values of anteriority within the word. That is, no word contains a sound from the set of [+anterior] sibilants in Navajo [s,z,ts,ts',dz] and a sound from the [-anterior] sibilant set [ʃ,ʒ,tʃ,tʃ',dʒ]. Thus there are words like the two given in (4), but there are no words like those given in (5).

(4) a. **ʃ**iːteː**ʒ** ‘we (dual) are lying’
b. da**s**doːli**s** ‘he (4th) has his foot raised’

(5) a. ***ʃ**iːteː**z** (hypothetical)
b. *da**s**doːli**ʃ** (hypothetical)

The sibilants in (4) are said to agree with respect to the feature [anterior].

The fact that arbitrarily many arbitrary segments separate agreeing sibilants in

Navajo distinguish Long Distance Agreement patterns from ones over contiguous segments, such as the *CCC constraint active in Yawelmani Yokuts.[3]

As in the Yawelmani Yokuts example, evidence that speakers have knowledge of this constraint comes from the fact that there are regular adjustments to words formed by the concatenation of morphemes so that the word conforms to the phonotactic constraint. For example, the perfective prefix *si*-is realized as [si] when attached to stems without sibilants, e.g. [si-ti] 'he is lying', but is realized as [ʃi] when attached to a stem with a [-anterior] sibilant as in [ʃi-teːʒ] 'they (dual) are lying.'

Note that the directionality of the agreement only becomes relevant when forms are compared in alternation. When we consider only legal surface forms, the directionality of the agreement can safely be ignored.

1.3 Stress Patterns

Some languages have rules or constraints which indicate which syllables in words are stressed. Stressed syllables are those that are somehow emphasized, often by strength, length, or pitch (Lehiste 1970). For example, English speakers say [tʌ.ˈmej.to], and neither [ˈtʌ.mej.to] nor [ˈto.mʌ.to]. Languages in which stress falls predictably within words form another class of phonotactic patterns, different from the other two kinds discussed above.

Stress patterns differ from the patterns over contiguous and non-contiguous segments in at least two ways. First, stress is suprasegmental, i.e. a property of the syllable, and not a property of the segment (Lehiste 1970). Second, stress patterns may be iterative, that is applied to alternating syllables (as shown in the example below) (Hayes 1995). Segmental patterns do not appear to have this property; e.g. linguists have not discovered rules in natural language where every other segment

[3] See Martin (2004) regarding the role of distance and the domain of sibilant harmony in Navajo compounds.

must be nasalized.[4]

For example, the words below are from Pintupi, a language which assigns stress predictably. The examples below are reproduced from Hayes (1995:62), who cites Hansen and Hansen (1969:163).

(6)	a.	σ́ σ	páɳa	‘earth’
	b.	σ́ σ σ	t^{j}úʈaya	‘many’
	c.	σ́ σ σ̀ σ	máɭawàna	‘through from behind’
	d.	σ́ σ σ̀ σ σ	púɭiŋkàlat^{j}u	‘we (sat) on the hill’
	e.	σ́ σ σ̀ σ σ̀ σ	t^{j}ámulìmpat^{j}ùŋku	‘our relation’
	f.	σ́ σ σ̀ σ σ̀ σ σ	tíɭirìŋulàmpat^{j}u	‘the fire for our benefit flared up’
	g.	σ́ σ σ̀ σ σ̀ σ σ̀ σ	kúran^{j}ùlulìmpat^{j}ùɻa	‘the first one who is our relation’
	h.	σ́ σ σ̀ σ σ̀ σ σ̀ σ σ	yúmaɻìŋkamàrat^{j}ùɻaka	‘because of mother-in-law’

When the column at left is examined, the stress pattern becomes apparent. It can be described adequately with the following two rules.

(7) 1.Assign secondary stress to nonfinal odd syllables, counting from left.

2.Assign primary stress to the initial syllable.

Note that the rules above apply, regardless of the length of the words. Judging by (h) in (7), it is clear that words in Pintupi can become quite long, and there

[4] However, see Gonzalez (1999) for some cases which might be analyzed this way.

appears to be no principled way to establish an upper bound on the length of Pintupi words. I assume, along with earlier researchers, that knowledge of the rules above means that a Pintupi speaker knows how to assign stress to a novel word (such as a borrowing) with more syllables.

1.4 Nonarbitrary Character of Phonotactic Patterns

The patterns found in natural language are not arbitrary. This follows provided the grammars which generate these patterns are ultimately constrained in some fashion. This line of thinking partly motivates the idea of Universal Grammar: Language patterns are not, for the most part, arbitrary because before the child has heard a single utterance, aspects of the child's grammar are fixed. These predetermined properties of the child grammar determine, through their interaction with the linguistic environment, mature characteristics of the adult grammar.

I take these predetermined properties to be the inductive principles children use to generalize from their limited experience to rules or constraints. This point relates to the third quote leading this chapter by Bertrand Russell (though again he is not referring to language). It is possible under this view that some logically possible grammatical hypotheses are never entertained by children, no matter their linguistic experience—thus constraining the kinds of attested natural language patterns nontrivially.[5] In this sense, the learner—i.e. the way children generalize—is the carver which shapes the possible natural language patterns. It follows that the properties found in natural language can help reveal properties of the learner, that is the relevant inductive principles.

[5] For related, extensive discussion, see *Apsects* (Chomsky 1965).

2 Phonotactic Grammars

This dissertation henceforth assumes that phonotactic constraints and rules are real and that they form part of the speaker's linguistic competence. Thus, it is assumed that, at some level, speakers of English know that words with a word-initial [pt] sequence are ill-formed. It is assumed that Navajo speakers would recognize a word which contains sibilants with different values of anteriority as less well-formed than a word which is the same in all respects except its sibilants agree in anteriority. Likewise, it is assumed that speakers of Pintupi recognize a string of sounds with a non-Pintupi stress pattern as less acceptable than the same string which obeyed the Pintupi stress rule.

I henceforth use the word *language* to refer to a (possibly infinite) set of words, which are sequences of elements drawn from some finite set, the *alphabet*. The alphabet can be thought of as symbols which represent possible human sounds, allowing for some variation.[6] I also assume that the grammar of a language L is a device which is generative—the words it recognizes are *well-formed* and belong to L, and the words it does not recognize are not well-formed and do not belong to L. Hence a grammar is finite device that generates potentially infinitely many words. This formulation of language is due to Chomsky (1957).

The phonotactic patterns described above are languages in this sense, and I will use the word *pattern* interchangeably with *language*. Every possible sequence of sounds which obey the constraint or rule belongs to the pattern, and thus is in the set. Likewise, every sequence which violates the rule or constraint is not in the set. Thus each rule or constraint cleaves the set of all possible words in two.

[6] This is compatible with construals of the symbols as rich phonological entities, e.g. as representing natural classes by way of underspecified feature bundles (Jakobson et al. 1952), as 'symboloids' (Port and Leary 2005), or as intervals of time coded for gestural information, cf. gestural scores (Browman and Goldstein 1992, Gafos 2002). See also the discussion in Chapter 1 §1.2.

For example, the aspect of the grammar for Yawelmani Yokuts which characterizes the *CCC constraint should accept any sequence of sounds which does not contain a contiguous sequence of consonants of length three. Any sequence which has a sequence of three contiguous consonants should be rejected by this grammar. Later in §2.2, the phonotactic rules and constraints described above are given precise grammars which characterize, in these simplified terms, the phonotactic knowledge described above.

Many researchers have recently argued categorically separating words into two groups, ill-formed and well-formed, underestimates the phonotactic knowledge speakers possess (Coleman and Pierrehumbert 1997b, Zuraw 2000, Frisch et al. 2004, Hayes and Wilson to appear, *inter alia*). Speakers are said to have gradient, as opposed to categorical knowledge. In other words, speakers do not make a binary distinction but instead distinguish many levels of well- and ill-formedness. Evidence for these claims is comprehensively reviewed in Hayes and Wilson (to appear).

This thesis does not consider gradient phonotactics. This is not because I think speakers do not have gradient knowledge, or that modeling gradient phenomena is somehow unimportant or uninteresting. It is simply because I think it is useful to abstract away from this complicating issue to get a clear picture of the kinds of patterns that are to be learned and the inductive principles necessary to learn them. By setting aside the issue of gradience, we see more clearly the contribution particular inductive principles can make, and the character of the resulting hypothesis spaces.

2.1 The Chomsky Hierarchy

On the basis of this conception of language, the complexity of certain groups of languages in the space of logically possible languages is shown in Figure 2.1 (Chomsky

1956b,a). The figure shows the space of recursively enumerable languages. A recursively enumerable language is one which can be generated by a Turing machine (Papadimitriou 1993). The smallest group in the figure, the Finite Languages, consist of those languages with finite cardinality (i.e. in this context, finitely many words). The next group up are the regular languages, followed by the context-free, the mildly context sensitive, and then context-sensitive languages. Each group of languages is more complex in the sense that the kind of grammar that is required to adequately describe languages is more permissive (see Sipser (1997), Hopcroft et al. (2001) for further details). This dissertation pays special attention to the regular languages for reasons given in §2.2 below.

Chomsky (1956b,a) demonstrates that certain syntactic patterns found in natural languages are not regular. Similar work by Shieber (1985), and more recently by Kobele (2006), have also argued that certain natural language syntactic constructions belong to even higher levels of the hierarchy. What is striking about the picture in Figure 2.1 is that the phonotactic patterns discussed above all belong to the class of *regular languages.*

2.2 Phonotactic Patterns as Regular Sets

Regular sets are those languages that can be generated by finite state acceptors (Hopcroft et al. 2001).[7] A language-theoretic characterization of the regular languages due to Nerode is given in the appendix to this chapter. I find it useful to represent phonotactic patterns as regular sets, and consequently to consider finite state representations of the grammars which generate these patterns. There are three reasons for this.

The first is that the virtually all phonotactic patterns are describable as regular

[7] They also relate nontrivially to what are known as finite Markov chains. There are many textbooks on Markov processes, a good introduction is given by Häggström (2002).

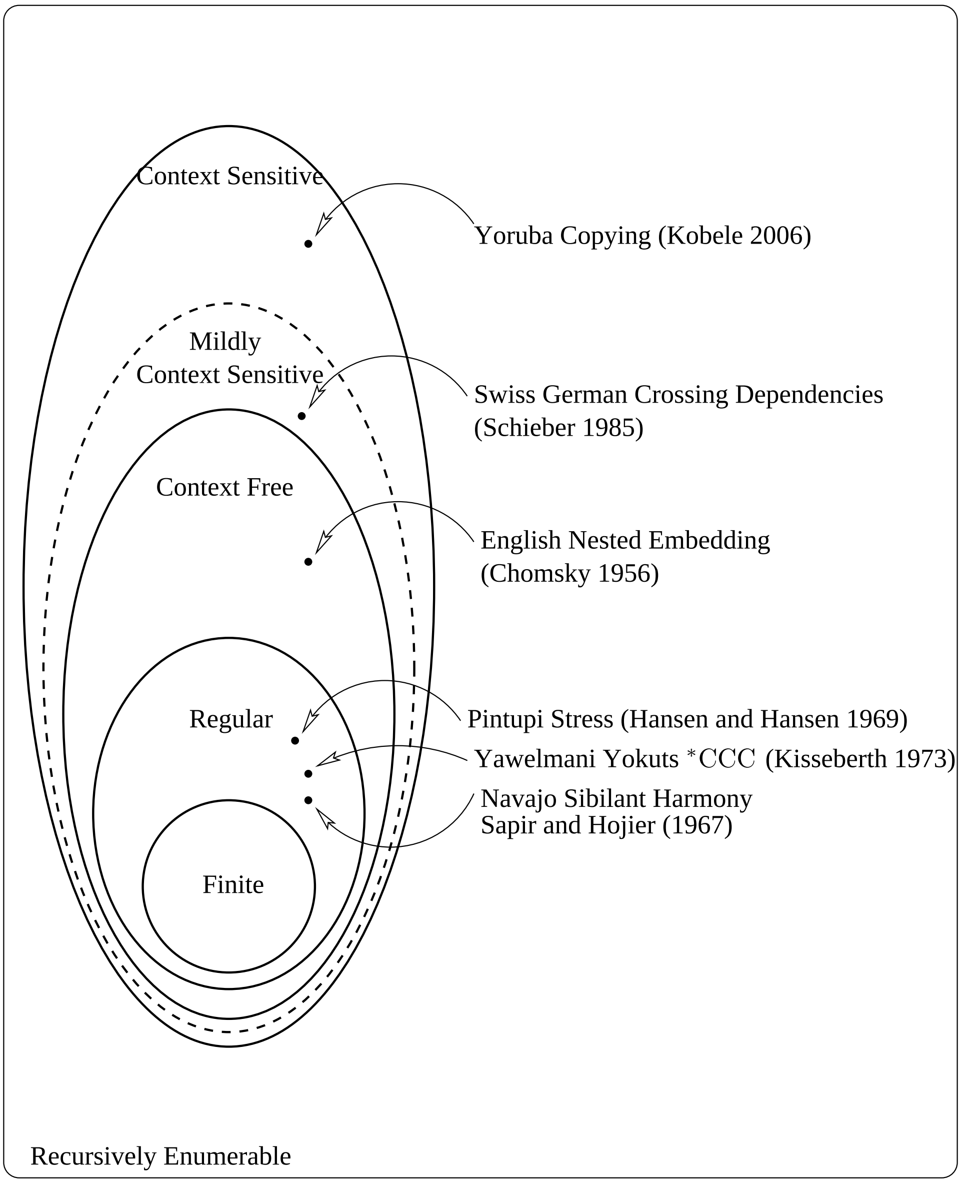

Figure 2.1: The Chomsky Hierarchy

sets.[8] To see this, consider first that most phonological processes are described as functions which map underlying forms to surface forms. This is true both in the rule-based formalisms associated with Sound Pattern of English (SPE) (Chomsky and Halle 1968) and in more recent parallel based formalisms, e.g. Optimality Theory (McCarthy and Prince 1993, Prince and Smolensky 2004). It has long been observed that almost all of these phonological processes are regular (Johnson 1972, Kaplan and Kay 1981, 1994). Specifically this means that the function which maps underlying forms to surface forms is a finite state function, and can be represented with a finite state transducer. For phonological grammars, the range of this function is properly interpreted as set of possible legal surface forms; i.e well-formed words. Consequently, phonotactic patterns are regular because of the well known fact that the range of a finite state function is a regular set (Hopcroft et al. 2001).

Second, these regular sets can be related to traditional phonological grammars. It is a simple matter to convert a finite state transducer, which maps underlying forms to surface forms, to a finite state acceptor which accepts only the well-formed words of the language. Johnson (1972) and Kaplan and Kay (1994) have shown how to construct this finite state transducer which maps underlying forms to surface forms from traditional SPE-style rule-based phonological grammars. Similarly, Riggle (2004), building on work by Ellison (1994), Eisner (1997), Frank and Satta (1998), and Albro (1998), shows how to construct a finite state transducer with OT grammars. Therefore, it is it is possible to compute a phonotactic acceptor for well-formed words given those traditional grammars.

Third, not only can finite state descriptions of grammars be translated to and from other representations of grammars (e.g. ordered SPE-style rules or ranked

[8] One possible exception to this is constraints that hold only in inherently reduplicated words. For example, one potential case is in Toba Batak (Tuuk 1971), where it appears certain word-internal consonant clusters are only found in inherently reduplicated words. Assuming this is part of a speaker's phonotactic knowledge, such a constraint cannot be stated adequately as a regular set. Thanks to Bruce Hayes for bringing up this example.

OT constraints), insights in this domain can be extended if it is determined that more complex types of grammars are needed. For example, Albro (2005) makes restricted extensions to a finite state system in order to handle reduplication. Also if the working assumption that phonotactic constraints are categorical is relaxed, stochastic finite state automata are a natural extension in which gradient well-formedness patterns can now be described.

2.3 Examples

Generally, throughout this dissertation, I will adopt the following principle as a way to represent phonotactic constraints or rules as finite state machines.

(8) A finite state accepter represents some phonotactic constraint (rule) iff every word accepted by the accepter obeys the constraint (rule) and every word rejected by the accepter violates the constraint (rule).

2.3.1 *CCC in Yawelmani Yokuts

For example, the machine in Figure 2.2 represents the *CCC* constraint in Yawelmani Yokuts. In finite state diagrams in this dissertation, unless mentioned otherwise, start states are indicated by hexagons, and final states with double peripheries (see Sipser (1997) or Hopcroft et al. (2001) for good introductions to finite state machines). In Figure 2.2, the 'C' labels on the transitions indicate any consonant, and the 'V' labels on the transitions indicate any vowel. Notice any word with three contiguous consonants is rejected by this machine because there is no transition labeled 'C' departing from state 2. Also notice that every word this machine accepts obeys the *CCC constraint.

This machine does accept words with hundreds of contiguous vowels, but such

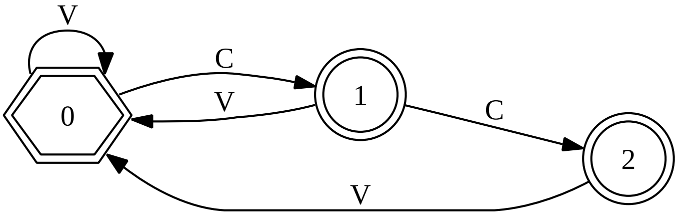

Figure 2.2: *CCC in Yawelmani Yokuts

words do not violate the *CCC constraint; such words violate other constraints which this constraint does not represent (perhaps something like *VVV).

2.3.2 Navajo Sibilant Harmony

The machine in Figure 2.3 represents the Navajo sibilant harmony pattern. In Figure 2.3, the ‘C’ labels on the transitions indicate any consonant except sibilants, the ‘V’ labels any vowel, the ‘s’ labels any [+anterior] sibilant, and the ‘ʃ’ labels any [-anterior] sibilant.

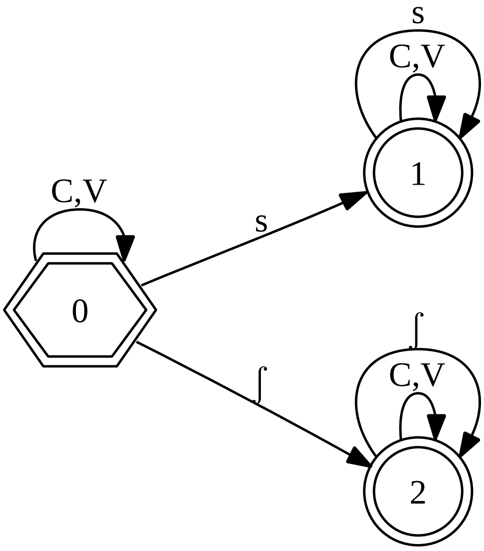

Figure 2.3: Navajo Sibilant Harmony

Every word this machine accepts obeys the rule of Navajo sibilant harmony, and every word it rejects disobeys it. It is true that this machine accepts words like [tnʃʃʃʃttttttʃiiii]—but this violates other constraints on well-formedness (e.g. syllable structure constraints). Finally, just like the grammar for the *CCC constraint above, this grammar recognizes an infinite number of legal words, just like the generative grammars of earlier researchers.

2.3.3 Pintupi Stress

Finally, the machine in Figure 2.4 represents the stress pattern of Pintupi. Comparison with the stress patterns on Pintupi words in (6) reveals that every word this machine accepts obeys the Pintupi stress rule and every word it rejects violates it. Also, note there is no upper bound on the length of words this machine recognizes.

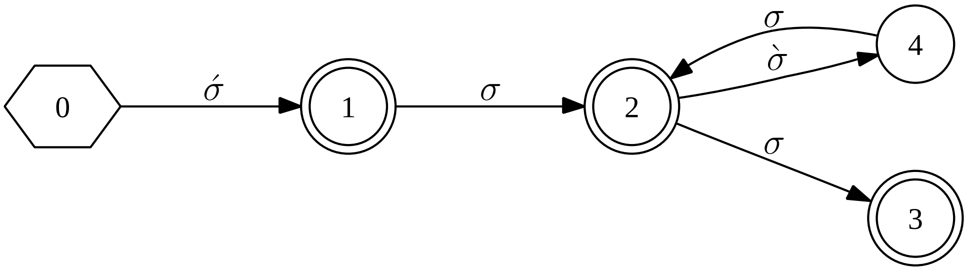

Figure 2.4: The Stress Pattern of Pintupi

2.4 Local Summary

These finite state representations of phonotactic rules and constraints accept infinitely many words, just like the generative grammars of previous researchers.

It is also important to recognize that these acceptors are only different descriptions of the same finite state function described by more traditional phonological grammars. For example, if Riggle's (2004) algorithm were applied to the different OT analyses of the Pintupi stress pattern given in Gordon (2002) and Tesar (1998), and the resulting transducers were stripped of their input labels and hidden structural symbols (such as foot boundaries) in the output labels, the acceptors obtained would be the same as the one in Figure 2.4.[9]

It is now possible to state the problem this dissertation addresses more precisely:

(9) *How can these finite state phonotactic grammars be acquired from a few words generated by them?*

3 Addressing the Learning Problem

In this section, I make explicit the learning framework adopted in this dissertation. The view of the learning process I find useful is schematized in Figure 2.5. Further reading on the viewpoint offered here can be found in Nowak et al. (2002), Niyogi (2006), Jain et al. (1999) and Osherson et al. (1986).

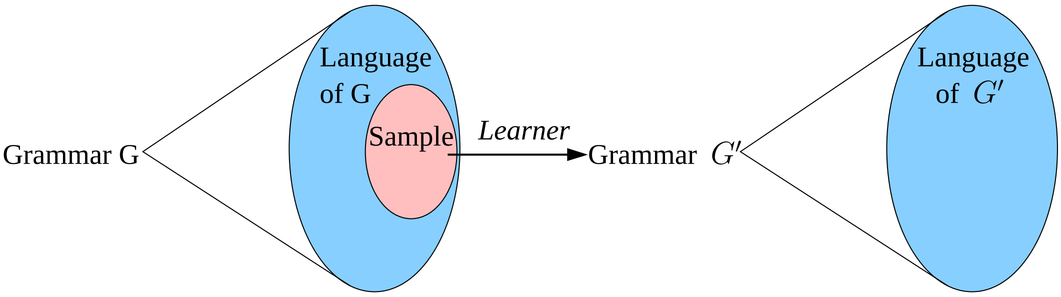

Figure 2.5: The Learning Process

The idea is there is a grammar G which generates a possibly infinite language $L(G)$. However, the learner does not have access to every well-formed expression in $L(G)$. The learner only has access to a small finite sample from $L(G)$. The learner is thus a function mapping finite samples to grammars. In this way the learner

[9] By 'same', I mean they both recognize the same language.

resembles a human child: given limited exposure to $L(G)$, the learner returns some grammar G'. We are interested in questions like:

(10)
1. Which learners are sure to discover a grammar G' such that the language of G is the same as the language of G'? That is, which learners generalize from limited experience in the right way?
2. Which learners can do this for samples drawn from any language which belong to some class of languages (such as those languages which contain patterns over contiguous segments)?
3. What kind of sample does the learner need to succeed in this way?

These questions are just a few of the ones that can be asked. For example, instead of asking question 1 in (10), it is possible to inquire which learners 'almost' succeed—that is, arrive at a grammar which generates a language not exactly like the language of G, but like one 'close' to it.

There are many frameworks within which the above questions can be addressed, and others are expected to be discovered (Angluin 1992). The rest of this section reviews two different established instantiations of the learning framework schematized in Figure 2.5—the Gold framework (Gold 1967), and the Probably-Approximately Correct (PAC) framework (Valiant 1984, Kearns and Vazirani 1994)—and discusses some of the important features of each framework. Although the content of the following subsections is comprehensively reviewed elsewhere (e.g. Niyogi (2006)),[10] the review here is useful as there is confusion about the relevance and significance of these learning frameworks (see Johnson (2004) for an overview of some of the confusion surrounding the Gold framework, for example).

Both the Gold and PAC frameworks make precise the philosophical problem

[10] A compact review is given in Stabler (2007).

of induction (e.g. Russell 1912, Popper 1959). Both of these frameworks show that most classes of languages shown in the Chomsky Hierarchy (Figure 2.1) are not learnable. They demonstrate that those classes are too large and that there are therefore necessarily too many distinctions for any learner to possibly make. In other words, in both frameworks, it is known that there is no procedure which returns the acceptors described above no matter the linguistic environment. Consequently, there is no known answer to the question posed in (9).

However, there is a silver lining. As described below, researchers have discovered subclasses of languages which cross-cut the Chomsky Hierarchy which can be learned in interesting ways. Thus suggests a research strategy to be discussed below in §4.

3.1 The Gold Learning Framework

Gold (1967) lets the target language to be any recursively enumerable language L. The input to the learner is an infinitely long sequence of well-formed words belonging to L, called a *text*. The fact that every element of the text is a well-formed expression may be taken to reflect that children primarily use positive evidence in language acquisition (E. Newport and Gleitman 1977, Marcus 1993). Additionally, every possible well-formed word of L is guaranteed to occur somewhere in this text. Thus it is assumed under the Gold framework that there are no 'accidental gaps'; if the learners wait long enough, they are bound to hear any particular well-formed word which belongs to some language L.

At each point the learner is allowed to make a guess as to what the target language is. Although the input stream to the learner is infinitely long, learners do not have an infinite amount of time to succeed, however. Crucially learners are *not* functions from infinitely long texts to grammars. Rather, the learner is a function from its finitely many observations of well-formed words (i.e. the text it has seen up to some point) to grammars. The learner succeeds if and only if it can be shown that for any text there is a point where the learner hypothesizes a grammar that generates L exactly and no later word in the sequence makes the learner alter her hypothesis. This is called *exact identification in the limit.*

It should be clear that this framework is not intended to provide the most realistic model of language learning. Children do not receive a perfect stream of well-formed expressions. Also, if children learned language exactly, then languages may very well not change over time (Niyogi 2006). However, the purpose of the Gold framework is to provide a platform where the problem of induction and generalization can be studied clearly. Thus there is a strict requirement that the acquired language be exactly the same as the target language. In all other respects, however, the Gold framework is generous. The input to the learner contains no errors. The learner itself need not be computable in time or space. Thus, the Gold framework puts the question of generalization squarely before us. Given the exacting goal of language identification in the limit, but generous conditions for the learner to operate in, how can one generalize correctly from experience to the target language (if at all)?

Gold's most significant results were overwhelmingly negative. He shows that any 'superfinite' class of languages—any class which contains all finitely-sized languages and at least one infinitely sized language—*cannot* be identified exactly in the limit. In other words no learner can learn every language in this class, no matter how much perfect data the learner receives, no matter how much space or time the learner needs to develop a hypothesis with respect to the observed data. Consequently, supersets of this class—such as the regular languages, the context-free languages, the context-sensitive languages, and the class of languages which can be represented with grammars—are also not identifiable in the limit. The appendix to this chapter gives a formal treatment of the Gold framework, including these important negative results.

Thus, it is not known how the phonotactic acceptors presented above, which represent the phonotactic knowledge native speakers possess, can be acquired from limited experience. Even if the linguistic environment is perfect and not corrupted by noise, even if the child may use immense amounts of time or space to construct hypotheses—we do not know of algorithms which can generalize from this highly idealized experience to yield the kinds of acceptors described above. This is the problem this dissertation aims to solve.

3.2 The Probably-Approximately Correct (PAC) Framework

The Probably Approximately Correct (PAC) framework,[11] developed by Valiant (1984), is a statistical learning framework which arrives at results like those in the Gold framework. The strict criterion of exact identification in the limit is relaxed—here, the learner is required to converge to approximately the right language with high probability. Also, whereas learners in the Gold framework are only able to observe well-formed words, the PAC framework allows learners access to negative evidence—that is, knowledge of which forms are ill-formed.

As in the Gold framework, words are presented to the learner from a stream. However, the learner is not guaranteed to see at some point each word belonging to the target language. Instead the words are drawn according to a probability distribution (which is not known to the learner). If it can be shown that there is a point in any text after which the learner's hypothesized language is sufficiently similar to the target language with high confidence no matter the initial probability distribution then the learner is said to be probably-approximately correct (PAC).

[11] Anthony and Biggs (1992), Kearns and Vazirani (1994) provide good introductions to the PAC framework.

The terms 'sufficiently similar' and 'high confidence' are parameters of the system; i.e. the point in the text by which point the learner must converge is dependent only on these parameters, and not anything else (such as the probability distribution over the well-formed expressions).

As in the Gold framework, key results are negative. Most of the major classes in the Chomsky hierarchy are not PAC-learnable. In fact, not even the class of finite languages is PAC-learnable (recall that it is Gold-learnable). Thus, changing learning frameworks does not change the basic results. An appendix at the end of the chapter formalizes the PAC model and proves these negative results.

3.3 Summary of Negative Results

Both the Gold and PAC learning frameworks make clear the fundamental problem of induction that lies at the heart of the learning problem in linguistics. If the class of patterns to be learned is too large and unrestricted, then no learner makes enough distinctions to learn any pattern within that class. In both frameworks, the core classes of the Chomsky Hierarchy are too vast (Figure 2.1) for any learner to succeed. It is important to realize that although the PAC and Gold frameworks are different—one is not a generalization of the other nor is one an instantiation of the other—they agree in this respect.

Thus the simplifying measures adopted in this dissertation do not lead to a trivialization of the learning problem. It is not known, even if grammars are categorical (which they probably are not), even if the input to the learner is perfect (which it is not), which inductive principles children could employ to generalize correctly from finite experience to the kinds of phonotactic patterns discussed earlier in this chapter.

It is true that any finite collection of languages is identifiable in the limit and

PAC-learnable, and that both the P&P and OT frameworks predict only finitely many grammars (because there are finitely many a priori parameters or constraints). Therefore, it follows trivially that they are Gold- and PAC- learnable. We may wonder then what the finite state characterizations buy us if we are after (at least) Gold-learnability. However, the learner in Gold's proof makes no interesting generalizations at all because it takes no advantage of any interesting property of the languages within the class, much less of any interesting property that those languages share to the exclusion of logically possible languages outside the class. Even if the hypothesis space is finite (and therefore trivially Gold-learnable), it is still necessary to develop a learner which learns this particular class as opposed to some other logically possible class that happens to be finite (see discussion in Chapter 1 §2).

3.4 Positive Results

Despite the negative results in the Gold and PAC frameworks, the learning problem is not insurmountable or unapproachable, however. It is known that certain hypothesis spaces which cut across the major Chomskyian classes are learnable within these frameworks. For example, Angluin (1982) demonstrates that the reversible languages are efficiently identifiable in the limit. Similarly Kanazawa (1996) demonstrates that a certain class of languages recognizable by categorical grammars are also identifiable in the limit. Yokomori (2003) also shows that a subset of the context free languages is identifiable in the limit. Interesting language classes can be learned in variants of the PAC model (i.e. where the point in the text is determined by additional parameters in the system, e.g. Clark and Thollard (2004)). Typically, these precedents provide learners whose inductive principles relate directly to properties of the target class. This is a very natural notion—that the properties of the target class and properties of the learner are tightly intertwined and are not accidentally related.

4 A Research Strategy

Results in formal learning theory suggest a strategy: identify properties which define some small class of languages—which contain natural language patterns—whose properties naturally provide inductive principles for learning. What are these properties for the subset shown in Figure 2.6? Answering this question is beyond the scope of the present work which only seeks to shed light on how phonotactic patterns of the kind described earlier can be learned. Thus the problem tackled here is really the one restricted to regular sets as shown in Figure 2.7.

However, it is possible to push this idea even further. Recall from chapter 1 the main thesis in (1) repeated here:

(11) *Properties of the learning mechanism explain patterns found in natural language.*

In Chapter 1, we saw that it follows naturally from this thesis that different classes of patterns are expected to have different learners. In this chapter, we have seen within the domain of phonotactics, the classes of patterns that are found are not the same. If this perspective is right, then different inductive principles may be necessary to learn different subsets of the regular languages which, as shown in Figure 2.8.

This idea has precedent. For example, in phonology, researchers have already proposed that phonotactic grammars be learned prior to the grammars which govern phonological alternations (Albright and Hayes 2003b, Hayes 2004, Prince and Tesar 2004). This proposal is consistent with acquisition studies which show that it

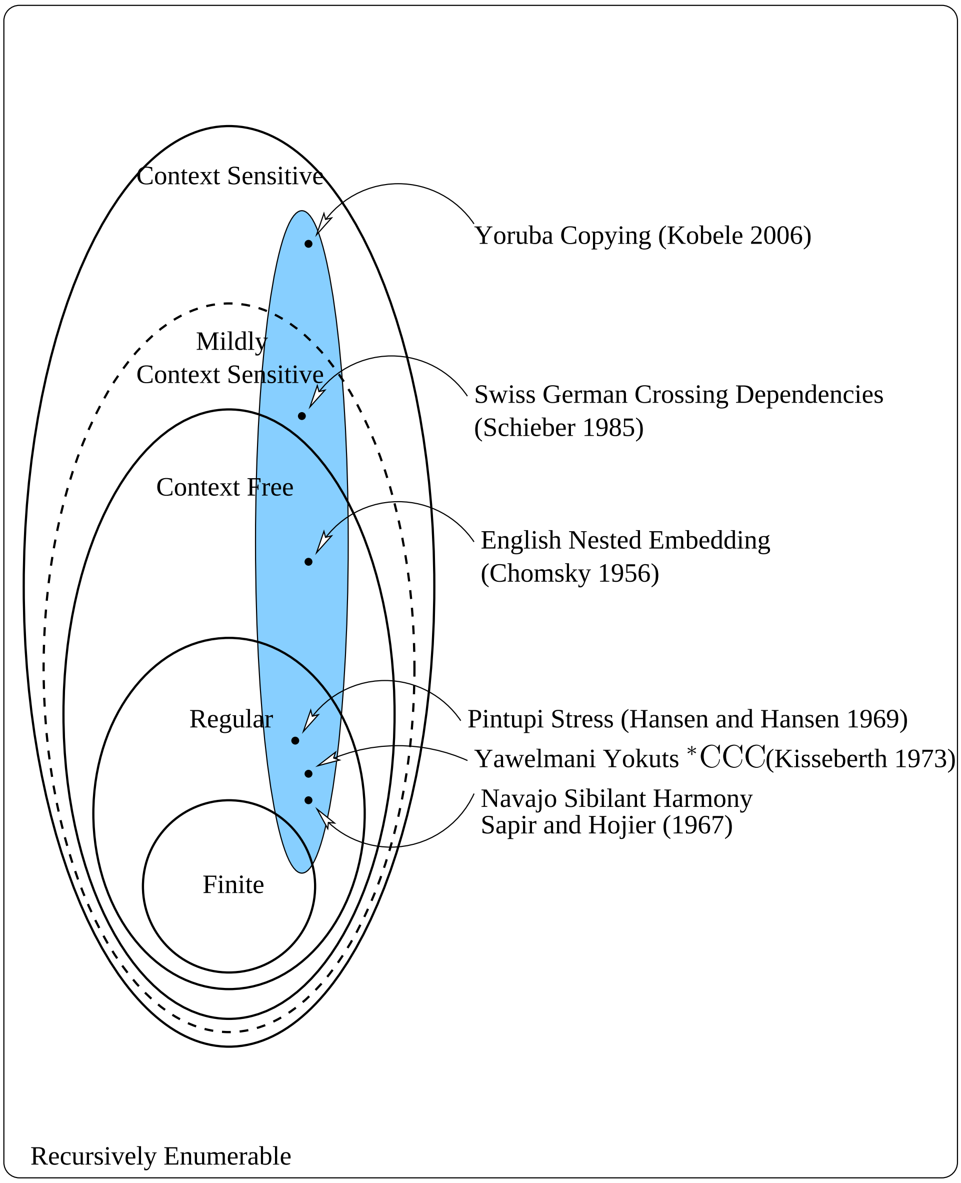

Figure 2.6: Locating Human Languages in the Chomsky Hierarchy

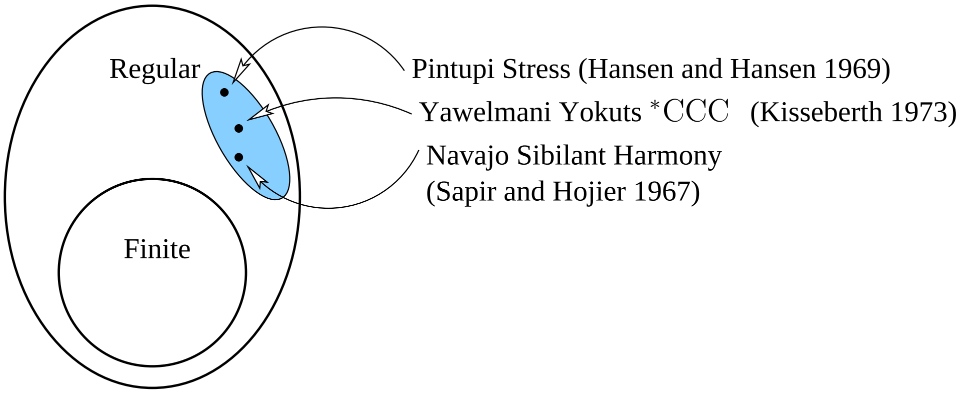
Regular
Finite
Pintupi Stress (Hansen and Hansen 1969)
Yawelmani Yokuts *CCC (Kisseberth 1973)
Navajo Sibilant Harmony
(Sapir and Hojier 1967)

Figure 2.7: Locating Phonotactic Patterns in the Regular Languages (I)

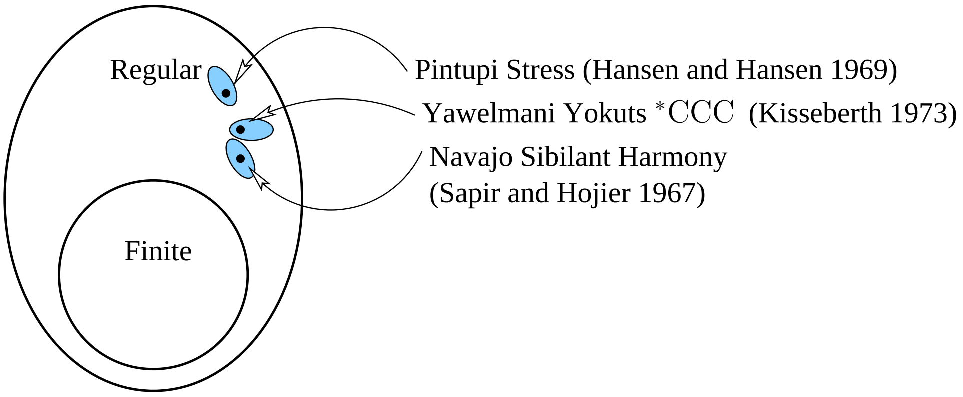
Regular
Finite
Pintupi Stress (Hansen and Hansen 1969)
Yawelmani Yokuts *CCC (Kisseberth 1973)
Navajo Sibilant Harmony
(Sapir and Hojier 1967)

Figure 2.8: Locating Phonotactic Patterns in the Regular Languages (II)

is likely that children possess some phonotactic knowledge at an early age (Friederici and Wessels 1993, Jusczyk et al. 1993a,b, 1994), probably before they have knowledge of morphology and alternations. On this empirical basis, phonologists are already proposing different learning mechanisms for different aspects of the whole phonological grammar. Of course one can question how far one should go in developing different learners for different classes of patterns. However, diversification often precedes unification, and I expect the same principle to be at work here.

When we have a better understanding of different learning functions, and how the ranges of these learning functions relate to the patterns found in natural language, linguists will be in a much better position to find principles or properties of human languages and learning that make it possible for children to acquire a language despite their limited experience. I believe the strategy adopted here leads us closer to this goal.

5 Summary

This chapter introduced three broad classes of phonotactic patterns, explained what phonotactic knowledge is, and argued that it is useful to represent such knowledge as finite state acceptors. This chapter introduced two frameworks which allow the learning problem in linguistics to be studied carefully and concluded that the simplifying assumptions made in this dissertation did not lead to a trivialization of the problem. Finally, results in formal learning theory suggest that key properties of natural languages will be intertwined with the properties of the learner.

Appendices

B–1 A Formal Treatment of the Gold Framework

I assume some enumeration of the recursively enumerable (r.e.) languages for the purposes of this section. Thus each r.e. language is uniquely identified by a natural number. This number is a grammar in the sense that one can recover the language by looking it up in the enumeration. The language of a particular grammr G is denoted $L(G)$. I assume the notations introduced in Appendix A–1.

B–1.1 Definitions

Here the Gold framework is established precisely and two significant results are given. We bgein by defining the linguistic environment as a text.

A *text* is an infinite sequence over $\Sigma^* \cup \{\epsilon\}$ where ϵ represents a non-expression. I.e. $t : \mathbb{N} \rightarrow \Sigma^* \cup \{\epsilon\}$. As is usual, the ith element of t is denoted $t(i)$. We denote with $t[i]$ the finite sequence $t(0), t(1), \ldots t(i)$. Following Jain et al. (1999), let SEQ denote the set of all possible finite sequences $\{t[i] : t \text{ is a text and } i \in \mathbb{N}\}$.

I use the word *content* to describe the range of a text. The *content* of a text t is

$$\mathit{content}(t) = \{w \in \Sigma^* : \exists n \in \mathbb{N} \text{ such that } t(n) = w\}$$

A text t is a *text for a language* L iff $\mathit{content}(t) = L$. Similarly, the content of a first part of a text is defined:

$$\mathit{content}(t[i]) = \{w \in \Sigma^* : \exists n \leq i \text{ such that } t(n) = w\}$$

A text t is a *positive text* for a language L iff $\mathit{content}(t) = L$. In other words, every expression in L occurs somewhere in a positive text for L.

A *learner* is a function ϕ which maps finite sequences to grammars, where each grammar G determines by the enumeration some language $L(G)$. Thus, $\phi : SEQ \to \mathbb{N}$. Note that a learner can be a partial function. A learner *converges on a text* t iff there exists $i \in \mathbb{N}$ and a grammar G such that for all $j > i$, $\phi(t[j]) = G$.

A learner ϕ *identifies a language* L *in the limit* iff for any positive text t for L, ϕ converges on t to grammar G and $L(G) = L$. Finally, we extend the notion of identification in the limit to classes of languages. A learner ϕ *identifies a class of languages* $\mathcal{L}$ *in the limit* iff for any $L \in \mathcal{L}$, ϕ identifies L in the limit. In such a case, we say $\mathcal{L}$ is *identifiable in the limit*.

These definitions establish the Gold framework. It is easy to see directly from these definitions that if a class of languages $\mathcal{L}$ is not identifiable in the limit, then no superset of $\mathcal{L}$ is either.

B–1.2 Any Superfinite Class is not Gold-learnable

The negative results of the Gold framework are made precise in Theorem 4. First, however, we begin by proving the Gold-learnability of any language class which is a finite collection of languages. Then we establish Angluin's (1980) characterization of Gold-learnable language classes, by which it becomes easy to see that no 'superfinite' (all finite languages and at least one infinite languages) class of languages is identifiable in the limit.

Theorem 1 Let $\mathcal{L}_{fin}$ be the class of finite languages. That is, $L \in \mathcal{L}_{fin}$ iff $|L|$ is finite. Then $\mathcal{L}_{fin}$ is identifiable in the limit.

Proof: Consider any $L \in \mathcal{L}_{fin}$, and let $\phi(t[i])$ be the first grammar in the enumeration such that $L(G) = content(t[i])$. Consider any positive text t for L. Then since $|L|$ is finite and $content(t) = L$, the $|content(t)|$ is finite also. Thus there is $i \in \mathbb{N}$ such that $content(t[i]) = L$ and $content(t[i-1]) \subset L$. Hence for any $j \geq i$,

$content(t[j]) = L$ as well. Therefore, $\phi(t[i]) = G$ such that $L(G) = L$ and for any $j > i$, $\phi(t[j]) = G$ such that $L(G) = L$ as well. $\square$

In order to establish the main result, it will be necessary to review Blum and Blum's (1975) theorem on locking sequences, which define a necessary condition on any Gold-learnable class of languages.

Theorem 2 Suppose a learner ϕ identifies a language L in the limit. Then there is some $\sigma \in SEQ$ such that

1. $content(\sigma) \subset L$
2. $\phi(\sigma) = G$ where $L(G) = L$.
3. for any $\tau \in SEQ$ such that $content(\tau) \subseteq L$, $\phi(\sigma \diamond \tau) = \phi(\sigma)$.

In other words, if a learner identifies a language L in the limit, then there is point in some text in which the learner is 'locked' into a particular grammatical hypothesis.

Proof: The proof is by contradiction. If the theorem is not true, it must be the case that for every $\sigma \in SEQ$ such that (1) and (2) above are true, there is a $\tau \in SEQ$ such that $content(\tau) \subseteq L$, but $\phi(\sigma \diamond \tau) \neq \phi(\sigma)$.

If this is true, then it is possible to construct a positive text for L with which ϕ fails to converge, thus contradicting the initial assumption that ϕ idenitifies L in the limit. It will be helpful to consider some positive text $t(0), t(1), t(2), \ldots, t(n), \ldots$. Construct the new text q recursively as follows. Let $q^{(0)} = t(0)$. Note that $content(q^{(0))}$ is a subset of L. $q^{(n)}$ is determined according to the following cases:

Case 1. $\phi(q^{n-1}) = G$ where $L(G) = L$. Then let $q^{(n)} = q^{(n-1)} \diamond \tau \diamond t(n)$. We know τ exists by the reductio assumption. Note also that $content(q^{(n)})$ is a subset of L.

Case 2. $\phi(q^{n-1}) \neq G$. Then let $q^{(n)} = q^{(n-1)} \diamond t(n)$. As in the other case, *content*$(q^{(n)})$ is a subset of L.

Since *content*$(t) = L$ (because t is a positive text for L), and since an element of the t is added to q at every step in its construction, *content*$(q) = L$. I.e. q is a positive text for L.

However, ϕ fails to converge on the text q because for every $i \in \mathbb{N}$ such that $\phi(q^{(i)}) = G$ where $L = L(G)$, there is a later point $q^{(i+1)}$, where $\phi(q^{(i+1)})$ does not equal G by the construction above (Case 1). Therefore, we contradict the original assumption that ϕ identifies L in the limit and the reductio assumption is false, proving the theorem. □

Now it is possible to state the a property of all classes of languages which are identifiable in the limit. A crucial concept is the *characteristic sample* of a language in some class, defined below.

Definition 3 Any finite $S \subset \Sigma^*$ is a *characteristic sample* of a language $L \in \mathcal{L}$ iff $S \subseteq L$ and for any $L' \in \mathcal{L}$ which contains S, $L \subseteq L'$.

If a learner guesses language L upon observing a characteristic sample for L (for some class of languages which includes L, then it is guaranteed that the learner has guesses the smallest language in the class which contains the sample. Thus the learner has not overgeneralized as no other language in the class of languages which includes the sample is strictly contained within L.

Angluin's theorem states that a class of languages is identifiable in the limit iff every language in the class has a characteristic sample.[12]

Theorem 3 (Angluin 1980) Let $\mathcal{L}$ be some collection of languages. $\mathcal{L}$ is identifiable in the limit iff there exists a characteristic sample for each $L \in \mathcal{L}$.

[12] This version of the theorem is less powerful than the one given in Angluin (1980) which shows these classes of languages have computable learners).

Proof: ($\Rightarrow$) Suppose $\mathcal{L}$ is identifiable by ϕ. By Theorem 2, there is a locking sequence σ for L. We show that the *content*(σ) is a characteristic sample for L. First, since locking sequences are finite, *content*(σ) is finite too. Now for contradiction assume that there is some L' in $\mathcal{L}$ such that *content*$(\sigma) \subseteq L'$, and $L' \subset L$. Then ϕ fails to identify L' on a text t for L' where $t = \sigma \diamond \tau$ since, by Theorem 2, $\phi(\sigma) = G$ where $L(G) = L$.

($\Leftarrow$) Assume that every $L \in \mathcal{L}$ has a characteristic sample S_L (if there is more than one pick one for each L). Assume some enumeration of grammars and let $\phi(t[i])$ be the first grammar in the enumeration such that $S_{L(G)} \subseteq$ *content*$(t[i]) \subseteq L(G)$ if it exists, otherwise let it be the first grammar in the enumeration.

Now consider any $L \in \mathcal{L}$, any text t for L and let G be the n-th grammar in the enumeration, but the first such that $L(G) = L$. Since $S_{L(G)}$ is finite, there is an i_1 such that $S_{L(G)} \subseteq$ *content*$(t[i_1]) \subseteq L(G)$. Thus for all $j \geq i_1$, $\phi(t[j])$ returns G unless there is some G' earlier in the enumeration such that $S_{L(G')} \subseteq$ *content*$(t[i_1]) \subseteq L(G')$ (where $L(G') \in \mathcal{L}$).

However, we can find $i_2 \geq i_1$ which ensures that no such G' exists. To see this, suppose there is some G' earlier in the enumeration such that $S_{L(G')} \subseteq$ *content*$(t[i_1]) \subseteq L(G')$. Then $L(G')$ cannot properly include L because $S_{L(G')}$ is a characteristic sample for G' e and both $L, L(G') \in \mathcal{L}$. Thus there must be some sentence s in L that is not in $L(G')$. Since text t is a text for L, there is a k such that $s \in$ *content*$(t[k])$. Thus for any $j \geq k$, $\phi(t[j]) \neq G'$ since $s \notin L(G')$ and thus *content*$(t[j]) \not\subseteq$. Thus, for each G_m (such that $L(G_m) \in \mathcal{L}$) which occurs earlier in the enumeration than G (i.e. $m < n$), there is some k_m such that *content*$(t[k_m]) \not\subseteq L(G_m)$. It is easy to see now that by simply letting i_2 be the largest element of $\{i_1\} \cup \{k_m : 0 \leq m < n\}$, we guarantee that for any $j \geq i_2$, $\phi(t[j]) = G$. □

Consequently, it is now easy to show that no superfinite class of languages is identifiable in the limit.

Theorem 4 (Gold 1967) Let $\mathcal{L}$ be any class of languages equal to $\mathcal{L}_{fin} \cup \{L'\}$ where $|L'|$ is countably infinite. Then $\mathcal{L}$ is not identifiable in the limit.

Proof: Every finite subset L_0 of L' is such that there exists $L \in \mathcal{L}$, $L_0 \subseteq L \subseteq L'$. Thus there is no characteristic sample for L' and by Theorem 3, no ϕ identifies $\mathcal{L}$ in the limit. □

Since no superset of non-Gold-learnable class of languages is identifiable in the limit, we immediately obtain Gold's (1967) result.

Corollary 1 The class of regular, context-free, context-sensitive, and recursively enumerable languages are not identifiable in the limit.

B–2 A Formal Treatment of the PAC Framework

B–2.1 Definitions

The PAC learning framework was first introduced by (Valiant 1984). See Anthony and Biggs (1992), Kearns and Vazirani (1994) and Niyogi (2006) for good introductions.

We assume a probability distribution P over Σ^*. I.e. for all $w \in \Sigma^*$, $0 \leq P(w) \leq 1$ and $\Sigma_{w \in \Sigma^*} P(w) = 1$.

Since the PAC framework allows the learner access to both positive and negative evidence, the definition of a text must be redefined to accomodate the inclusion of negative evidence. Let the *characteristic function* for a language L be defined as follows:

$$f_L(w) = \begin{cases} 1 & \text{iff } w \in L \\ 0 & \text{otherwise} \end{cases}$$

A *(pac) text* for a language L is an infinite sequence:

$$\{(w_0, c_0), (w_1, c_1), \ldots : w_i \in \Sigma^* \text{ and } c_i = f_L(w_i)\}$$

As before, Let SEQ_{pac} denote the set of all texts restricted to finite length.

A *(pac) learner* is a function ϕ from first which maps elements of SEQ_{pac} to grammars. Note that a learner can be a partial function.

In order to quantify the amount of error that exists between a learner's hypothesized language and the target language, it is necessary to introduce a distance metric over the language space. For any two languages L, L' and define the distance between the two languages as follows:

$$d(L, L') = \sum_{w \in \Sigma^*} |f_L(w) - f_{L'}(w)| P(w)$$

Now we can define what it means for a language class to be PAC-learnable. A class of languages $\mathcal{L}$ is PAC-learnable if and only if there exists a learner ϕ which, for any probability distribution P over Σ^*, for any $L \in \mathcal{L}$, and for all $0 \leq \epsilon \leq 1/2$, $0 \leq \delta \leq 1/2$, then with probability greater than $1 - \delta$, there exists $m(\epsilon, \delta)$ such that for any $\sigma \in SEQ_{pac}$ such that $|\sigma| > m(\epsilon, \delta)$, $d(L(\phi(\sigma)), L) < \epsilon$.

B–2.2 The VC Dimension

Blumer et al. (1989) prove a necessary and sufficient condition on classes that are PAC-learnable: they have finite Vapnik-Cherveonkis (VC) dimension. The VC dimension is a measure of class complexity and so this result establishes a deep nontrivial relationship between class complexity and learning. The proof in Blumer et al. (1989) is too involved to reproduce here, but I use this result to establish that even the class of finite languages is not learnable.

The following definitions establish the meaning of the VC dimension (Kearns and Vazirani 1994). Given a space X and a concept class $A \subseteq 2^X$. We say that for a sample $S \subseteq X$, A *shatters* S if and only if A picks out every subset of S, i.e. if and only if $\forall x \in 2^S, \exists G \in A$ such that $G \cap S = x$. The *Vapnik Chervonekis Dimension of X*, denoted VCD(X), is equal to the cardinality of the largest set S shattered by X. If arbitrary large finite sets of S can be shattered by C, then VCD(X) $= \infty$.

B–2.3 The Class of Finite Languages is not PAC-learnable

We use Blumer et. al.'s (1989) result to show that no class of languages shown in the Chomsky Hierarchy in Figure 2.1 is PAC-learnable.

Theorem 5 $\mathcal{L}_{fin}$ has infinite VC dimension.

Proof: Consider any finite sample $S \subset \Sigma^*$. Clearly, for any subset T of S, there exists $L \in \mathcal{L}$ such that $L \cap S = T$, namely $L = T$. Thus S is shatterable. Since S was arbitrary, VCD($\mathcal{L}$) is infinite. □

Corollary 2 The class of finite languages is not PAC-learnable.

Corollary 3 The classes of regular, context-free, context-sensitive, and recursively enumerable languages are not PAC-learnable.

B–3 Finite State Acceptors

B–3.1 Definition

An acceptor A is a quadruple (Q, I, F, δ) such that Q is finite, I and F are subsets of Q and δ is map from $Q \times \Sigma$ to subsets of Q. The set Q contains the *states* of A. I denotes the *initial states* and F denotes the *final states* of A. The *transition function* is denoted by δ.

B–3.2 Extending the Transition Function

The transition function can be extended naturally to map a set of states and a string to a set of states. First,

$$\delta(Q, a) = \{\delta(q, a) : q \in Q\}$$

Next, the transition function is extended over strings:

$$\delta(Q, u) = \begin{cases} \text{if } u = \lambda \text{ then } Q \text{ else} \\ \bigcup\{\delta(\delta(\{q\}, w), a) : q \in Q,\ a \in \Sigma \text{ and } u = wa\} \end{cases}$$

For $p, q \in Q$, if $q \in \delta(\{p\}, u)$ we say u *transforms* p to q. Likewise, if $Q_1 = \delta(Q_0, u)$ we say u *transforms* Q_0 to Q_1. We will often write $\delta(q_0, u) = q$ when $\delta(q_0, u) = \{q\}$.

Lemma 1 For any acceptor $A = (Q, I, F, \delta)$, and for any $q \in Q$, if u *transforms* q to q then for all $n \in \mathbb{N}$ u^n transforms q to q.

Proof: Let $A = (Q, I, F, \delta)$, and consider $q \in Q$ and $u \in \Sigma^*$, such that u *transforms* q to q. We do proof by induction. Clearly when $n = 1$, $u^1 = u$ transforms q to itself. Assume for some n, u^n transforms q to itself. Since u transforms q to itself, $u^n u = u^{n+1}$ must transform q to itself. This completes the induction and the lemma is proved. □

B–3.3 The Language of an Acceptor

An acceptor $A = (Q, I, F, \delta)$ *accepts* the string u iff $F \cap \delta(I, u) \neq \emptyset$. The *language of acceptor* A, denoted $L(A)$ is all $u \in \Sigma^*$ such that A accepts u.

B–3.4 Binary Operations

The *product* of two acceptors $A = (Q_A, I_A, F_A, \delta_A)$ and $B = (Q_B, I_B, F_B, \delta_B)$ is equal to $C = (Q_C, I_C, F_C, \delta_C)$ where

$$\begin{aligned} Q_C &= Q_A \times Q_B \\ I_C &= I_A \times I_B \\ F_C &= F_A \times F_B \\ \delta_C((q_A, q_B), a) &= \{(q'_A, q'_B) : q'_A \in \delta_A(q_A, a) \text{ and } q'_B \in \delta_B(q_B, a)\} \\ & \quad \text{for all } q_C \in Q_C \text{ and } a \in \Sigma \end{aligned}$$

We write $A \times B = C$. We also call A and B *factors* of C.

The *sum* of two acceptors $A = (Q_A, I_A, F_A, \delta_A)$ and $B = (Q_B, I_B, F_B, \delta_B)$ is defined provided $Q_A \cap Q_B = \emptyset$ in which case it is equal to $C = (Q_C, I_C, F_C, \delta_C)$ where

$$\begin{aligned} Q_C &= Q_A \cup Q_B \\ I_C &= I_A \cup I_B \\ F_C &= F_A \cup F_B \\ \delta_C &= \delta_A \cup \delta_B \end{aligned}$$

We write $A + B = C$. Note that the states of an acceptor can always be renamed without changing the language that it accepts (see §B–3.7 so it is always possible to find the union of two acceptors even if some of the states have the same name.

The proofs for the following two lemmas can be found elsewhere (e.g. Hopcroft et al. (2001)).

Lemma 2 $L(A \times B) = L(A) \cap L(B)$.

Lemma 3 $L(A+B) = L(A) \cup L(B)$.

B–3.5 Reverse Acceptors

The *reverse* of the transition function is denoted by δ^r is defined as

$$\delta^r(q,a) = \{q' : q \in \delta(q',a) \text{ for all } a \in \Sigma,\ q \in Q\}$$

δ^r is extended in a similar fashion as δ.

The reverse of an acceptor $A = (Q, I, F, \delta)$ is $A^r = (Q, F, I, \delta^r)$. Pictorially, the reverse of an acceptor is obtained by reversing the direction of the transition arrows and swapping the initial and final states.

Lemma 4 $L(A) = L(A^r)^r$.

Proof: Let $A = (Q, I, F, \delta)$. Since A accepts u, $F \cap \delta(I, u) \neq \emptyset$. However this means there is a final state from which δ^r transforms u^r to some initial state, i.e. $I \cap \delta^r(F, u^r) \neq \emptyset$. Therefore A^r accepts u^r, and $u^r \in L(A^r)$, which means $u \in (L(A^r))^r$. It is similarly shown that any $u \in L(A^r)^r$ belongs to $L(A)$. □

B–3.6 Forward and Backward Deterministic Acceptors

An acceptor $A = (Q, I, F, \delta)$ is *forward deterministic* iff $|I| \leq 1$ and for each state $q \in Q$ and $a \in \Sigma$, $|\delta(q,a)| \leq 1$. A is *backward deterministic* iff $|F| \leq 1$ and for each state $q \in Q$ and $a \in \Sigma$, $|\delta^r(q,a)| \leq 1$. It is easy to show that A is backward deterministic iff A^r is forward deterministic. Sometimes, instead of backward deterministic, we say *reverse deterministic.* An acceptor which is both forward and backward deterministic is called *zero-reversible* (Angluin 1982).

B–3.7 Relations Between Acceptors

This section is basically verbatim from Angluin (1982). Consider any two acceptors $A = (Q, I, F, \delta)$ and $A' = (Q', I', F', \delta')$.

A is *isomorphic* to A' iff there is a bijection h from Q to Q' such that $h(I) = I'$, $h(F) = F'$, and for all $q \in Q$ and $a \in \Sigma$ it is the case that $h(\delta(q, a) = \delta'(h(q), a)$. In other words, two acceptors are isomorphic if, modulo renaming of states, they are the same. If acceptors A and A' isomorphic, then $L(A) = L(A')$.

A' is a *subacceptor* of A iff $Q' \subseteq Q$, $I' \subseteq I$, $F' \subseteq F$, and for every $q' \in Q'$ and $a \in \Sigma$, $\delta'(q', a) \subseteq \delta(q', a)$. Pictorially, a subacceptor is obtained by removing some states and transitions from the diagram of an acceptor.

B–3.8 Stripped Acceptors

This section is basically verbatim from Angluin (1982).

Let $A = (Q, I, F, \delta)$. If Q_0 is a subset of Q, then the *subacceptor induced by* Q_0 is the acceptor $(Q_0, I_0, F_0, \delta_0)$ where $I_0 = I \cap Q_0$, $F_0 = F \cap Q_0$, and $q_0 \in \delta_0(q, a)$ iff $q, q_0 \in Q_0$ and $q_0 \in \delta(q, a)$. A state in $q \in Q$ is *useful* iff there exist strings u and v such that $q \in \delta(I, u)$ and $F \cap \delta(q, v) \neq \emptyset$. States that are not useful are called *useless*. An acceptor with no useless states is called *stripped*. The *stripped subacceptor of* A is the subacceptor of A induced by the useful states of A.

B–3.9 Cyclic Acceptors

A state q in an acceptor A is called *cyclic* iff there exists a string $u \in \Sigma^+$ such that u transforms q to itself. An acceptor $A = (Q, I, F, \delta)$ is *cyclic* iff there exists $q \in Q$ which is cyclic. If no such state exists, A is called *acyclic*. It is well known that an acceptor recognizes an infinite language iff its stripped subacceptor is cyclic.

B–3.10 Languages and the Machines which Accept Them

An acceptor relates the prefixes of a language and their tails. The following lemmas and corollarys help establish language theoretic characterizations of acceptor-based definitions of regular languages later.

Let $A = (Q, I, F, \delta)$ be any acceptor for a regular language L and let $q \in Q$. Let the *acceptable suffixes* of q are

$$S(q) = \{u : F \cap \delta(q, u) \neq \emptyset\}$$

Similarly, the *acceptable prefixes* of q are

$$P(q) = \{u : q \in \delta(I, u)\}$$

Lemma 5 Let $A = (Q, I, F, \delta)$ and denote $L(A)$ with L. Then for any $u \in Pr(L)$, $T_L(u) = \bigcup\{S(q) : u \in P(q) \text{ for all } q \in Q\}$.

Proof: Let $A = (Q, I, F, \delta)$ be any acceptor. If A is the empty acceptor then L is the empty language and the above is vacously true so assume $L(A)$ is not empty. Let L denote $L(A)$ and let $u \in Pr(L)$. Let S_u denote $\bigcup\{S(q) : u \in P(q) \text{ for all } q \in Q\}$.

Consider any $v \in S_u$. By definition, $\exists q$ such that $q \in \delta(I, u)$ and $F \cap \delta(q, v) \neq \emptyset$. Thus A accepts uv, i.e. $v \in T_L(u)$ hence $S_u \subseteq T_L(u)$. Now consider any $v \in T_L(u)$. Since A accepts L, A accepts uv. Consequently, $\exists q$ such that $q \in \delta(I, u)$ and $F \cap \delta(q, v) \neq \emptyset$. Thus $v \in S(q) \subseteq S_u$. Hence it is also true that $T_L(u) \subseteq S_u$ so $T_L(u) = S_u$. □

Corollary 4 Let $A = (Q, I, F, \delta)$ be any acceptor, let L denote $L(A)$. For all $u_1, u_2 \in Pr(L)$, $T_L(u_1) = T_L(u_2)$ iff $\{S(q) : u_1 \in P(q) \text{ for all } q \in Q\} = \{S(q) : u_2 \in P(q) \text{ for all } q \in Q\}$.

Corollary 5 Let $A = (Q, I, F, \delta)$ be a forward deterministic acceptor. Let L denote $L(A)$. For all $u_1, u_2 \in Pr(L)$, if $\delta(I, u_1) = \delta(I, u_2)$ then $T_L(u_1) = T_L(u_2)$.

B–3.11 Tail Canonical Acceptors

Let L be a regular language. The *tail-canonical acceptor* for L is $A_T(L) = (Q, I, F, \delta)$ defined as follows:

$$\begin{array}{rcl} Q & = & \{T_L(u) : u \in Pr(L)\} \\ I & = & \{T_L(\lambda)\} \\ F & = & \{T_L(w) : w \in L\} \\ xs\delta(T_L(u), a) & = & T_L(ua) \text{ iff } u, ua \in Pr(L) \end{array}$$

An acceptor isomorphic to the tail-canonical acceptor is called *tail-canonical* or simply *canonical*. For a regular language L, the tail-canonical acceptor is the forward deterministic acceptor with the fewest states. There is an efficient procedure for obtaining a tail-canonical acceptor from any forward deterministic acceptor for L (as described in Hopcroft et al. (2001)).

Lemma 6 Let L be a regular language and let $A_T(L) = (Q, I, F, \delta)$. Now consider any $u_1, u_2 \in Pr(L)$. Then $T_L(u_1) = T_L(u_2)$ iff $\delta(I, u_1) = \delta(I, u_2)$.

Proof: The right-to-left direction is immediate from Corollary 5. Then for any $u_1, u_2 \in Pr(L)$ such that $T_L(u_1) = T_L(u_2)$. From the definition of $A_T(L)$, $\delta(I, u_1) = T_L(u_1) = T_L(u_2) = \delta(I, u_2)$. □

B–3.12 The Myhill-Nerode Theorem

A right congruence is a partition of π of Σ^* with the property that $B(w_1, \pi) = B(w_2, \pi)$ iff $B(w_1u, \pi) = B(w_2u, \pi)$ for all $w_1, w_2, u \in \Sigma^*$. For any language L, $T_L(w_1) = T_L(w_2)$ iff $T_L(w_1u) = T_L(w_2u)$. Thus, L determines an associated right congruence π_L by $B(w_1, \pi_L) = B(w_2, \pi_L)$ iff $T_L(w_1) = T_L(w_2)$. This relation, called the *L-equivalence relation* and denoted $\sim_L$, forms the basis for the next theorem, which establishes a language-theoretic characterization of regular sets (see for more details (Khoussainov and Nerode 2001)).

Theorem 6 A language L is recognizable by a finite state acceptor if and only if the partition π_L induced by the L-equivalence relation $\sim_L$ has finite cardinality.

Proof: ($\Rightarrow$) Consider any finite state acceptor $A = Q, I, F, \delta$ and let $L = L(A)$. First define π_A as follows:

$$B(u, \pi_A) = B(v, \pi_A) \text{ iff } \delta(I, u) = \delta(I, v)$$

Since the co-domain of δ is 2^Q, the cardinality of π_A can be no greater than $2^{|Q|}$. Hence the cardinality of π_A is less than this, and is therefore finite. It it is easy to see that $\sim_A$ for any $u, v, w \in \Sigma^*$, $u \sim_A v$ iff $uw \sim_A vw$. Consequently if $u \sim_A v$, $uw \in L$ iff $vw \in L$, which implies u and v are L-equivalent. It follows from this that every block of the partition π_L is a union of the blocks from the π_A, i.e. π_L is coarser than π_A. Thus the number of blocks in π_L is equal to or less than the number of blocks in π_A, which we know to be finite. Thus, π_L is finite.

($\Leftarrow$) Consider any language L such that π_L has finite cardinality. Then we can construct the acceptor $A_L = (Q, I, F, \delta)$ defined as follows:

$$\begin{aligned} Q &= \{B(u, \pi_L) : u \in Pr(L)\} \\ I &= \{B(\lambda, \pi_L)\} \\ F &= \{B(w, \pi_L) : w \in L\} \\ \delta(B(u, \pi_L), a) &= B(ua, \pi_L) \end{aligned}$$

Note that A is deterministic. Since there are only finitely blocks in π_L, Q is finite. It remains to be shown that $L = L(A)$. For any w, in L, w transforms $B(\lambda, \pi_L)$ to $B(w, \pi_L)$ and since $B(w, \pi_L) \in F$ by definition, $w \in L(A)$. For any

$w \in L(A)$, $B(w, \pi_L)$ must be a final state in A and thus by definition $w \in L$. Thus the theorem is proved. □

Finally, note that the acceptor constructed in the proof is isomorphic to the tail canonical acceptor for L.

B–3.13 Head Canonical Acceptors

This section introduces some important concepts that will be utilized later, in chapters 3,4, and especially 5 and 6. I include it here primarily so that it occurs in one place, independent of the many places where the ideas are used. Also, these ideas develop naturally (actually in parallel with) the development we saw in the previous sections of this appendix, so it make sense to include them together in the same appendix.

This section introduces the head canonical acceptor for a regular language L, which is the smallest *reverse deterministic* acceptor which accepts L. It is striking that reverse determinism might play a role in natural language, or in fact in any process, due to the 'moving backward in time' aspect of reverse determinism. However, the idea here is hardly without precedent in computer science, where two-way processes are commonly employed, e.g. (Viterbi 1967, Baum 1972).[13]

B–3.13.1 Background

In the same way that we developed the notion of prefixes of a language L (see §A–1.5), we can also discuss the suffixes of L, which as shown in Lemma 7, bear an interesting relationship to $Pr(L)$.

[13] Thanks to Stott Parker for bringing up this connection.

Definition 4 The *suffixes* of a language are defined to be

$$Sf(L) = \{u : \exists v \text{ so that } vu \in L\}$$

Lemma 7 $Pr(L) = Sf(L^r)^r$.

Proof: Consider any $u \in Pr(L)$. Thus there is a string v such that $uv \in L$. Therefore, $v^r u^r \in L^r$ and $u^r \in Sf(L^r)$ and $u \in Sf(L^r)^r$. Likewise, for any $u \in S(L^r)^r$, it is the case that $u^r \in Sf(L^r)$. Thus there is some v such that $vu^r \in L^r$. It follows that $uv^r \in L$ which establishes $u \in Pr(L)$. □

Corollary 6 $Sf(L) = Pr(L^r)^r$.

In the same way that the notion of prefixes of a string given a language led naturally to a definition of the tails of that string in that language, the suffixes of a string lead naturally to a definition of what I call the *heads* of the string given that language.

Definition 5 The right-quotient of language L and string w, or the *heads* of w given L, is denoted by

$$H_L(w) = \{u : uw \in L\}$$

Thus, $H_L(w) \neq \emptyset$ iff $w \in Sf(L)$. Note also that for any $u \in Sf(L)$, $H_L^0(u) = \{\lambda\}$. Finally, note the relationship between the heads of a string for a given language and the tails of that string for that language.

Lemma 8 For any L, $w \in L$, $T_L(w) = H_{L^r}(w^r)^r$.

Proof: $\forall u, w \in \Sigma^*$, $L \subseteq \Sigma^*$,

$$u \in T_L(w) \leftrightarrow wu \in L \leftrightarrow u^r w^r \in L^r \leftrightarrow u^r \in H_{L^r}(w^r) \leftrightarrow u \in H_{L^r}(w^r)^r$$

□

Corollary 7 $H_L(w) = T_{L^r}(w^r)^r$.

In the same way that an acceptor relates the prefixes of a language with their tails (see §B–3.10), we also establish a relationship between the suffixes of a language and their heads. Recall the definitions of the acceptable suffixes and prefixes of a state q from §B–3.10 (repeated below).

$$S(q) = \{u : F \cap \delta(q, u) \neq \emptyset\}$$

$$P(q) = \{u : q \in \delta(I, u)\}$$

Lemma 9 Let $A = (Q, I, F, \delta)$ be any acceptor for a regular language L. Then for any $v \in Sf(L)$, $H_L(v) = \bigcup\{P(q) : v \in S(q) \text{ for all } q \in Q\}$.

Proof: Let $A = (Q, I, F, \delta)$ be any acceptor. If A is the empty acceptor then L is the empty language and the above is vacously true so assume $L(A)$ is not empty. Let L denote $L(A)$ and let $v \in Sf(L)$. Let P_v denote $\bigcup\{P(q) : v \in S(q) \text{ for all } q \in Q\}$.

Consider any $u \in P_v$. By definition, there exists q such that $q \in \delta(I, u)$ and $F \cap \delta(q, v) \neq \emptyset$. Hence A accepts uv, i.e. $u \in H_L(u)$ and $P_v \subseteq H_L(u)$. Now consider any $u \in H_L(v)$. Since A accepts L, A accepts uv. Consequently, there exists q such that $q \in \delta(I, u)$ and $F \cap \delta(q, v) \neq \emptyset$ which implies $v \in P(q) \subseteq P_v$. Thus it is also true that $H_L(v) \subseteq P_v$ so $H_L(v) = P_v$. □

Corollary 8 Let $A = (Q, I, F, \delta)$ be any acceptor and let L denote $L(A)$. For all $v_1, v_2 \in Sf(L)$, $H_L(v_1) = H_L(v_2)$ iff $\bigcup\{P(q) : v_1 \in S(q) \text{ for all } q \in Q\} = \bigcup\{P(q) : v_2 \in S(q) \text{ for all } q \in Q\}$.

Corollary 9 Let $A = (Q, I, F, \delta)$ be a reverse deterministic acceptor for L. For all $v_1, v_2 \in Sf(L)$, If $\delta^r(F, v_1^r) = \delta^r(F, v_2^r)$ then $H_L(v_1) = H_L(v_2)$.

As a consequence of Lemma 9 and Lemma 5, the states in an acceptor define a relation from subsets of $Pr(L)$ to subsets of $Sf(L)$.

B–3.13.2 Definition and Theorem

Now we can introduce the head canonical acceptor. Let L be a regular language. The *head-canonical acceptor* for L is $A_H(L) = (Q, I, F, \delta)$ defined as follows:

$$\begin{aligned}
Q &= \{H_L(u) : u \in Sf(L)\} && \text{if } L \neq \emptyset, \text{ otherwise } Q = \emptyset \\
I &= \{H_L(w) : w \in L\} \\
F &= \{H_L(\lambda)\} \\
\delta(H_L(au), a) &= H_L(u) && \text{if } u, au \in Sf(L)
\end{aligned}$$

The head-canonical acceptor is typically not forward deterministic. It is however, the acceptor with the fewest states for a regular language L that is backward deterministic. Acceptors isomorphic to the head-canonical acceptor are called *head-canonical*. For some regular language L there is an efficient procedure for finding a head-canonical acceptor given any other backward deterministic acceptor for L. This has not been described anywhere to my knowledge but the algorithm is analogous to the one used to obtain tail-canonical isomorphic acceptors.[14]

Lemma 10 Let L be a regular language and let $A_H(L) = (Q, I, F, \delta)$. Then for any $v_1, v_2 \in Sf(L)$, $H_L(v_1) = H_L(v_2)$ iff $\delta^r(F, v_1^r) = \delta^r(F, v_2^r)$.

Proof: The right-to-left direction is immediate from Corollary 9. Now consider any $v_1, v_2 \in Sf(L)$ such that $H_L(v_1) = H_L(v_2)$. From the definition of $A_H(L)$, $\delta^r(F, v_1^r) = H_L(v_1) = H_L(v_2) = \delta^r(F, v_2^r)$. □

Here are some examples to see the differences between the head and tail canonical acceptors.

[14] Similarly, just as there is an algorithm which can forward determinize any acceptor for some L (but not necessarily efficiently), there is a similar algorthm which can always backward determinize any acceptor for some L.

Example 1 This example gives a regular language representative of the stress pattern of Hopi (limited to strings of light syllables). Hopi generally places stress on the peninitial syllable, but in words with two syllables or less, stress falls initially. Here, 1 indicates primary stress and 0 indicates no stress. Note that if the head-canonical acceptor were reversed it would be determinstic. In fact, as is proved below in Theorem 7, the reverse of the head canonical acceptor is the tail-canonical acceptor for L^r.

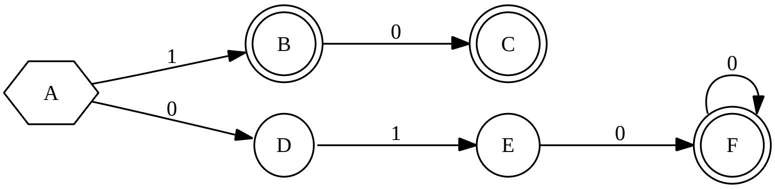

Figure 2.9: A Tail-canonical Acceptor for $L = \{1, 10, 010, 0100, 01000, \ldots\}$

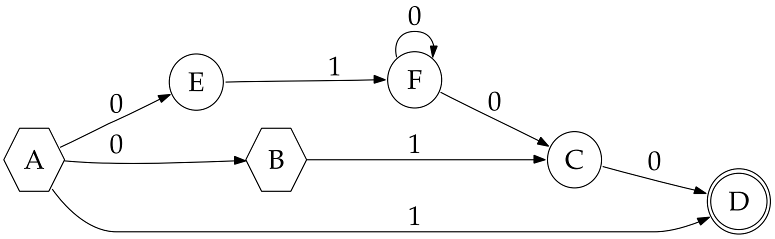

Figure 2.10: A Head-canonical Acceptor for $L = \{1, 10, 010, 0100, 01000, \ldots\}$

Example 2 The language accepted by the acceptor below accepts only words with an even number of 0s and 1s. Note that in this example, the tail-canonical acceptor is isomorphic to the head-canonical acceptor. Languages which have this property are forward and backward deterministic; i.e they are zero-reversible (Angluin 1982).

Finally, we prove a non-trivial relationship between the tail and head canonical acceptors.

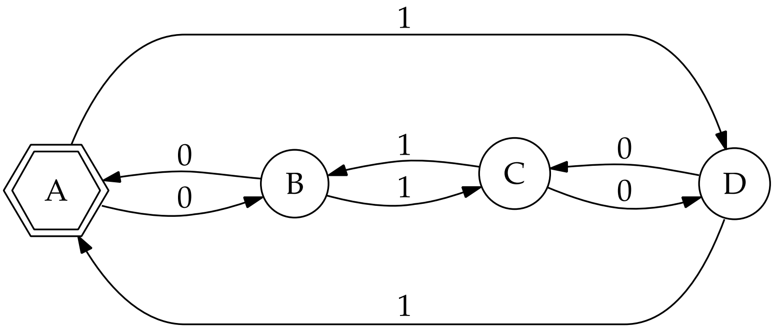

Figure 2.11: A Tail-canonical Acceptor for L={λ, 00, 11, 0011, 1100, 0101, . . .}

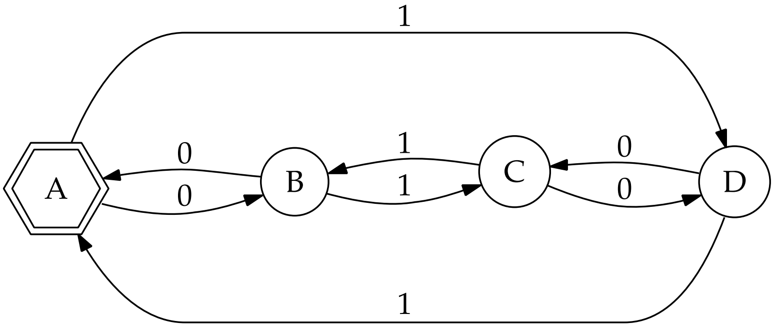

Figure 2.12: A Head-canonical Acceptor for L={λ, 00, 11, 0011, 1100, 0101, . . .}

Theorem 7 Let L be a regular language. Then $A_H(L)$ is isomorphic to $A_T(L^r)^r$.

Proof: Let L be a regular language and let $A_T(L) = (Q_T, I_T, F_T, \delta_T)$ and $A_H(L) = (Q_H, I_H, F_H, \delta_H)$. The bijection we need is $h(L) = L^r$.

First we esablish the mapping between states. Let $A_T(L^r) = (Q_{Tr}, I_{Tr}, F_{Tr}, \delta_{Tr})$ so $A_T(L^r)^r = (Q_{Tr}, F_{Tr}, I_{Tr}, \delta^r_{Tr})$. Then by definition $Q_{Tr} = \{T_{L^r}(u) : u \in Pr(L^r)\}$. For any $u \in Pr(L^r)$, $T_{L^r}(u) = H_L(u^r)^r$ by Lemma 8. So $h(T_{L^r}(u)) = h(H_L(u^r)^r) = H_L(u^r)$. Since $u \in Pr(L^r)$, $u^r \in Pr(L^r)^r$. But $Pr(L^r)^r = Sf(L)$ by Corollary 6 so $u^r \in Sf(L)$. Hence $H_L(u^r) \in Q_H$ from the definition of the head canonical acceptor.

Consider next F_{Tr}. As above, for any $u \in L^r$, $h(T_{L^r}(u)) = h(H_L(u^r)^r) = H_L(u^r)$. Since $u^r \in L$, $H_L(u^r) \in I_H$ by definition of the head canonical acceptor.

Next consider I_{Tr}. As above, $h(T_{L^r}(\lambda) = h(H_L(\lambda)^r) = H_L(\lambda)$, which belongs to F_H by definition of the head canonical acceptor.

Finally for any $u \in Pr(L^r)$, $a \in \Sigma$, $\delta_{Tr}(T_{L^r}(u), a) = T_{L^r}(ua)$ whenever $u, ua \in Pr(L^r)$ so $\delta^r_{Tr}(T_{L^r}(ua), a) = T_{L^r}(u)$. It remains to be shown that $h(\delta^r_{Tr}(T_{L^r}(ua), a) = \delta_H(h(T_{L^r}(ua)), a)$.

We show that both $h(\delta^r_{Tr}(T_{L^r}(ua), a)$ and $\delta_H(h(T_{L^r}(ua)), a)$ equal $H_L(u^r)$. Since $\delta^r_{Tr}(T_{L^r}(ua) = T_{L^r}(u)$ by definition of δ^r_{Tr}. But $T_{L^r}(u) = H_L(u^r)^r$ by Lemma 8, so $h(\delta^r_{Tr}(T_{L^r}(ua), a) = h(H_L(u^r)^r$. By definition of h, this equals $H_L(u^r)$, which is equivalent to $\delta_H(H_L(au^r), a)$ by definition of δ_H. Now $H_L(au^r) = h(H_L(au^r)^r)$ (because for any L, $(L^r)^r = L$) and $h(H_L(au^r)^r = h(T_{L^r}(ua))$ also by Lemma 8. Thus, $\delta_H(h(T_{L^r}(ua)))$ equals $H_L(u^r)$, which as proven equals $h(\delta^r_{Tr}(T_{L^r}(ua), a)$. Thus the theorem is proved. □

CHAPTER 3

Patterns over Contiguous Segments

1 Overview

In the last chapter I motivated an approach to the problem of phonotactic learning that requires identifying properties which define learnable subclasses of the regular languages which include the kinds of phonotactic patterns attested in the world's languages. In this chapter, two general schemes in which it can be understood how particular inductive principles can learn particular subsets of the regular languages are presented. One scheme is known as *state merging* (Biermann and Feldman 1972, Angluin 1982). The other I call *string extension learning.* These concepts are illustrated by showing how a categorical version of an n-gram model, a popular model in natural language processing which learns constraints over contiguous segments, is actually an instantiation of both of these more general schemes. The significance of this fact is that these more general concepts—state merging and string extension learning—are really vehicles for investigating the consequences of many possible inductive principles, some of which are explored in later chapters.

Although the n-gram based learning function studied here is easily expressible as an example of both string extension learning and state merging, it is not the case that any learner in one framework is easily expressible in the other. This is made clear by the learners in chapters 4 and 5. Thus, it really is necessary to present the two frameworks independently.

In §2, I define n-gram grammars and languages and develop a running example of a trigram grammar and language. In §3, I present a simple learner for n-gram languages which introduces the core idea behind string extension learning. In §4, I introduce the idea of state merging as a means by which generalization can occur. In §5, I make clear the inductive principle in play when learning n-gram languages and show how to instantiate it in the state merging framework. Finally, in §6, I investigate whether n-gram languages are appropriate characterizations of phonotactic patterns in general, and patterns over contiguous segments in particular. §7 summarizes the key developments of this chapter.

2 N-gram Grammars and Languages

N-gram grammars are categorical versions of n-gram models, which are a popular platform in many natural language processing scenarios (Manning and Schütze 1999, Jurafsky and Martin 2000). The n-gram model, originally conceived, is a way to predict the next element of a sequence given only the $n - 1$ previous elements. They play significant roles in many areas of natural language processing, including augmentative communication (Newell et al. 1998), spelling-error correction (Mays et al. 1991), part of speech tagging (Brill 1995), and speech recognition (Jelenik 1997).

N-gram models can also be used to compute a well-formedness value for a given word, based on the likelihood of the various contiguous sequences of segments found within the word. The idea equates transitional probabilities with well-formedness. Assuming that prohibited sequences occur with much less frequency than well-formed sequences, the model, once trained, will assign smaller well-formedness values to novel words containing those ill-formed sequences. For example, the cluster *stw* occurs very infrequently in corpi of English.[1] Consequently in a trained trigram model, the probability that a *w* follows the sequence *st* in English is very low; this low probability will bring down the well-formedness score of a possible word with an *stw* cluster. N-gram models can thus be thought of as a list of sequences of length n, each associated with some value between 0 and 1, which indicates its well-formedness (or likelihood).

I now define a categorical n-gram model, as opposed to the probabilistic variety. To distinguish these from their statistical counterparts, I call them n-gram *grammars* as opposed to models. There are two reasons for using a categorical grammar as opposed to a statistical model. First, it makes clear the hypothesis space that n-gram models operate within as well as the character of the languages within this space. This is important because many of the modifications and extensions that are made to basic n-gram models typically leave the character of the hypothesis space essentially intact. Second, it makes clear that n-gram based learning is an instantiation of two more general techniques called string extension learning (to be discussed in §3) and state merging (to be discussed in §4), both of which play a crucial role in subsequent chapters.

An n-gram grammar is simply a set of allowable sequences of length n in the language. The idea is that a word is well-formed iff every n-length subsequence in the word is licit; i.e. in the grammar. Such a language is called an n-gram language. Grammars of this kind—that is, grammars which determine whether a word is in its language by checking whether the result of some function applied to the word is a subset of the grammar—I call *string extension grammars.* In the case of the n-gram grammars, the relevant function is the one that returns the n-grams in a given word. String extension grammars are formally defined in Appendix C–1 and

[1] The cluster *stw* is not a legal onset cluster in English and is only found in compounds or across word boundaries as in *must win* (Clements and Keyser 1983).

have a number of interesting properties, some of which I make clear below.

Consider as an informal example, a language whose syllabic template is CV(C). (Yawelmani Yokuts, discussed in Chapter 2 §1.1, is such a language.) Words like those given in (1) obey the syllabic template whereas words like those in (2) do not.

(1) a. ka d. sak
b. puki e. bedko
c. kitepo f. piptapu

(2) a. *t g. *slak
b. *ak h. *partpun
c. *gast i. *manakk

To help with exposition, I will use the symbols C and V to stand for any consonant and any vowel respectively.[2] The trigram grammar G which generates the CV(C) pattern in (1) is given in (3). The symbol '#' indicates the word boundary.

(3) G = {#CV, CVC, VCV, VCC, CCV, CV#, VC#}

The well-formed words this grammar generates are all the words in which every subsequence of length three is present in the grammar. Thus words such as (a) in (1) has two subsequences of length three: #ka and ka#, which translate to #CV, CV#, respectively, and both of which are in the grammar; hence, *ka* is in the trigram language generated by G in (3). Similarly the trigram grammar G rejects a word like (a) in (2) because its one subsequence of length three #t# (translated to #C#) is not present in the grammar. Hence, *t* is not in the trigram language generated by the grammar.

[2] Alternatively we can think of the grammar only 'seeing' the consonantal feature in the segments which make up the words above (cf. Heinz (2006a)).

Later it will be shown that the grammar in (3) is equivalent to the finite state machine in Figure 3.1. This machine only accepts words in which every contiguous subsequence of length three is in the grammar in (3). It rejects all other words.

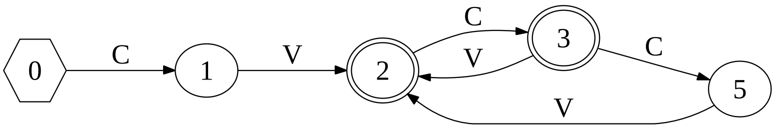

Figure 3.1: The CV(C) Syllable Structure Constraint

3 Learning N-gram Languages as String Extension Learning

First, however we show a simple way the grammar in (3) can be obtained from a list of finite examples.[3] As will be explained below, this learning procedure is an instantiation of *string extension learning.* The initial state of the learner's grammar is empty. All the learner does is record the subsequences of length three in observed words and translate them to sequences of Cs and Vs. For example, the table below shows how the grammar grows with each successive time step. New trigrams added to the grammar are given in bold.

Since the grammar G in (3) only generates words which obey the CV(C) template, no additional words will add any new sequences to the grammar in the last timestep in Table 3.1. Note that the learner generalizes tremendously on the basis of these few forms. For example, although it has not seen a CVCVC word, it knows that such a word is well-formed because each subsequence of length three which makes up a CVCVC word is in its grammar. In this way, the learner which records

[3] Because the n-gram languages (for some n) are a finite in number, there are many learner which can identify this class (Jain et al. 1999, Osherson et al. 1986). The learner I present is natural, however, in the sense that it can only learn this language class.

Table 3.1: Trigram Learning the CV(C) Syllabic Template

time	word	Subsequences of length 3	Grammar
0			∅
1	ka	{#CV, CV#}	{**#CV, CV#**}
2	puki	{#CV, CVC, VCV, CV#}	{#CV, CV#, **CVC, VCV**}
3	kitepo	{#CV, CVC, VCV, CV#}	{#CV, CV#, CVC, VCV}
4	sak	{#CV, CVC, VC#}	{#CV, CV#, CVC, VCV, **VC#**}
5	bedko	{#CV, CVC, VCC, CCV, VC#}	{#CV, CV#, CVC, VCV, VC#, **VCC, CCV**}

subsequences of length three identifies the language of the grammar G in the limit in the sense of Gold (1967). This is because it is guaranteed to converge to the correct grammar after seeing the finitely many forms which instantiate the elements which make up the grammar. In other terms, it can be shown that at each point in time, the learner hypothesizes the smallest trigram language consistent with the observations made so far. It follows that any language recongnizable by a trigram grammar is identifiable in the limit and thus the class of trigram languages is identifiable in the limit. Of course nothing here hinges on the value three, the arguments above carry through for n-gram languages for any given n.

It is also possible to identify which samples are *characteristic* for a given n-gram language. One property of characteristic samples is that there is enough information in the sample for a learner to correctly guess the target language (see also Appendix B–1.2). In the case of n-gram languages, a sample is characteristic provided, for every n-gram in the target grammar, there is some word in the sample with this n-gram. Note this does not mean that there must be as many words in the sample as there are n-grams. There could in fact just be one (perhaps very long) word in the sample.

One value of being able to identify characteristic samples is that we can investigate the extent to which such samples are present in children's linguistic environment. It is here that the concept of 'accidental gap' arises. If, for example, a particular n-gram is not present in the child's linguistic environment, yet the child learns to accept words which contain that n-gram, then there are two possible explanations.[4] The first is that the formal n-gram hypothesis space (and consequently the learner) is not right, and a different hypothesis space and learner are needed. The second is that substantitive factors are playing a role. For example, if we consider Jakobson's theory of distinctive features (Jakobson et al. 1952), we might expect that generalization also occurs along the lines of natural classes. The idea is that substantive features provides an extra layer of structure over a formal hypothesis space which a learner can use to overcome certain kinds of 'accidental gaps' (e.g. Tenenbaum 1999, Albright and Hayes 2002, Hayes and Wilson to appear). It remains an open question what kind of 'accidental gaps' exist for n-gram languages in human languages and whether they all can be handled by appeal to substantive features (also see discussion in §6). [5]

The learner exemplified in Table 3.1 is an example of string extension learning. The grammar is simply formed by collecting aspects of the grammar from each individually observed word via a function which maps words to elements of the grammar. In the case of n-gram languages, this function returns sets of n-grams. However, as shown in the appendix, key results can be generalized to different kinds of functions which determine different language classes. The utility of this is that it allows us to investigate other kinds of functions which result in language classes

[4] Here I am assuming the accidental gap is genuine; i.e. it is known (pehaps through evidence obtained in a lab) that the child actually determines that a word without the offending n-gram is more acceptable than one with the n-gram, all other things being equal.

[5] There is a third possibility that is often exploited in natural language processing. That is to engineer methods such as smoothing, etc. which are attempts to overcome these difficulties for n-gram models. See Jurafsky and Martin (2000) for more about smoothing.

that resemble natural language phonotactic patterns, a point I return to in Chapter 4.

Appendix C–1 introduces the general framework of string extension learning and proves the closure results and identifiability in the limit when the range of the function which maps a string to subsets of the grammar is finite. Appendix C–2 formalizes n-gram grammars, and shows that the function belongs to the the more general class of functions explored in Appendix C–1, thereby establishing the fundamental properties of n-gram languages.

4 Generalizing by State Merging

4.1 The Basic Idea

Here we show how the learner exemplified in Table 3.1 can be understood from a state merging perspective. The basic idea behind learning via some state merging technique is to write smaller and smaller finite state descriptions of the observed forms but keep some property invariant (Biermann and Feldman 1972, Angluin 1982). This is akin to generalizing by eliminating redundant environments in the input forms where what counts as a redundant environment is determined by the states that are chosen to be merged—that is, by the whatever a priori inductive principle is decided upon. A formal treatment of these ideas is given in §C–3.

The general scheme of learners of this type follow the two step procedure:

(4) 1.A finite state representation of the input.

2.Merge states that are equivalent (in some pre-determined sense).

Which finite state representation of the input is used and how it is decided which states to merge in this structure are the two key questions. These decisions determine everything: the kinds of generalizations that are made, and ultimately what class of languages can be learned. For now, we will use a prefix tree (defined below) to represent the input and focus on the more interesting question of how one decides whether two states are equivalent.

4.2 Prefix Trees

A *prefix tree* is a structured finite state representation of a finite sample. The idea is that each state in the tree corresponds to a unique prefix in the sample. Here 'prefix' is not used in the morphological sense of the word, but in the mathematical sense (see §A–1.5). This idea is exemplified in the Figures 3.2 - 3.3 below by naming each state with a number for reference, but also with the unique prefix the state represents.

A prefix tree is built one word at a time. As each word is added, an existing path in the machine is pursued as far as possible. When no further path exists, a new one is formed. Figure 3.2 shows the prefix tree built from the single word 'puki', translated into Cs and Vs (λ indicates the empty string). Figure 3.3 shows how the prefix tree is extended when the word 'bedko' is added to the tree. Finally,

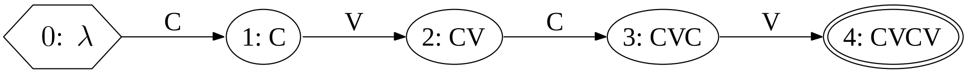

Figure 3.2: The Prefix Tree Built from a CVCV Word.

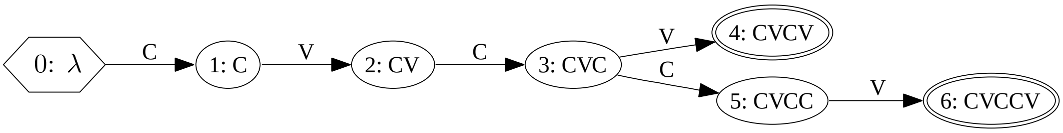

Figure 3.3: The Prefix Tree Built from Words {CVCV, CVCCV}.

Figure 3.4 shows the prefix tree given all the words in (1) (presented top down for easier page-fitting). A moment's reflection reveals that resulting prefix tree is the same even if the words that made it were presented in a different order.

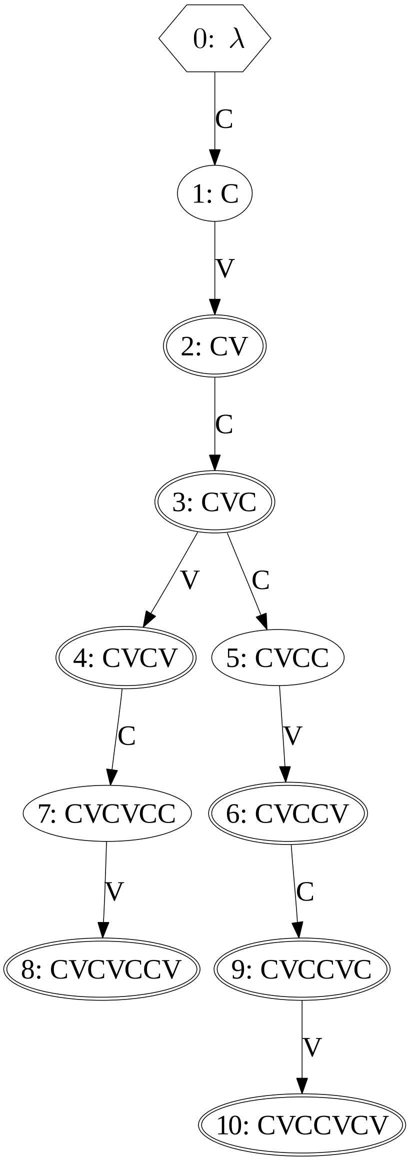

Figure 3.4: The Prefix Tree Built from Words in (1).

The prefix tree is a finite state acceptor which accepts only the finitely many forms that have been observed. No generalization has yet taken place. However, even in this simple example, it is possible to see that there is structure in the prefix tree, and that this structure repeats itself. state merging can eliminate structural redundancy, which may result in generalization.

4.3 State Merging

The next stage is to generalize by *merging states* in the prefix tree, a process where two states are identified as equivalent and then *merged* (i.e. combined). This section provides the basic ideas which are made precise in Appendix C–3.

A key concept behind state merging is that transitions are preserved (Angluin 1982, Hopcroft et al. 2001). This is one way in which generalizations may occur—because the post-merged machine accepts everything the pre-merged machine accepts, possibly more. For example in Figure 3.5, Machine B is the machine obtained by merging states 1 and 2 in Machine A. It is necessary to preserve the transitions in Machine A in Machine B. In particular, there must be a transition from state 1 to state 2 in Machine B. There is such a transition, but because states 1 and 2 are the same state in Machine B, the transition is now a loop. Whereas Machine A only accepts one word *aaa*, Machine B accepts an infinite number of words *aa, aaa, aaaa, ...*

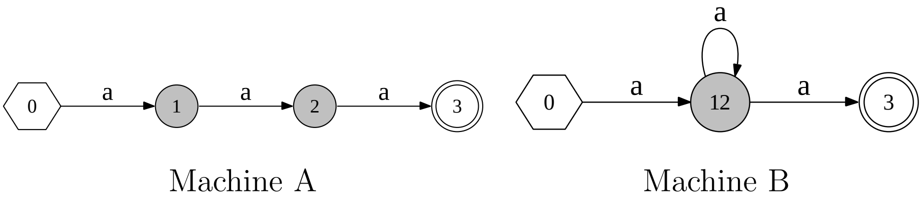

Figure 3.5: An Example of Generalization by State Merging

Note that the merging process does not specify which states should be merged. It only specifies a mechanism for determining a new machine once it has been decided which states are to be merged. Thus choosing which states are to be merged determines the kinds of generalizations that occur. A merging strategy is thus a generalization strategy. It is an inductive principle.

How can states be merged in the prefix tree in Figure 3.4 to return an acceptor which only generates words which obey the CV(C) syllabic template? As it turns out, there is more than one inductive principle which will do the trick. I review one method in §5 which can learn any language recognizable by a trigram grammar. As will be shown, this state merging strategy presented there utilizes the same basic idea as the learner represented in Table 3.1.

Appendix C–3 at the end of this chapter provides a formal treatement of prefix trees and state merging. In Appendix C–3.3, a key result is established: Given any canonical acceptor A for any regular language and a sufficient sample S of words generated by this acceptor, there is some way to merge states in the prefix tree of S which returns the acceptor A. This result does not tell us how to merge the states for a particular acceptor, it just says that such a way exists. Nonetheless, the result is important because it leaves open the possibility that there is some property of the phonotactic acceptors we write to characterize speakers' phonotactic knowledge for which there is a successful state merging strategy. In fact, the n-gram based learners exploit such a property, and later chapters demonstrate other state merging procedures which learn phonotactic patterns that are not over contiguous segments.

5 Learning N-gram Languages with State Merging

This section shows how one the simple n-gram learner presented in §3 is really a particular state merging strategy. Informally, the idea is made clear by examining the problem of learning the CV(C) syllabic template from surface forms.

As we saw, one way to learn the CV(C) syllabic template is if the learner employs a trigram grammar and records subsequences of length three in observed words as shown in Table 3.1. The state merging learner works in two stages given in (4). First, it builds a prefix tree of the observed words. Second, it merges states in the prefix tree whose corresponding prefixes share the same suffix of length two.

For example, consider the prefix tree in Figure 3.4. Because each state q in the prefix tree represents a unique prefix, we can identify suffixes of this prefix. If two states correspond to prefixes with the same suffix of length two, those states are merged, Table 3.2 shows the suffixes of length two for each state in the prefix tree in Figure 3.4. Note that in Table 3.2 states λ and C do not have suffixes of length two, so I have just listed the longest suffix for those states.[6]

	state	Suffix of Length Two
0	λ	λ
1	C	C
2	CV	CV
3	CVC	VC
4	CVCV	CV
5	CVCC	CC
6	CVCCV	CV
7	CVCCVC	VC
8	CVCCVCV	CV
9	CVCCVCVC	VC
10	CVCCVCV	CV

Table 3.2: Suffixes of Length Two for States in the Prefix Tree in Figure 3.4

From the above table, it is possible to see that states 2, 4, 6, 8 and 10 should

[6] Technically, we are supposed to list every suffix up to length two for a state. Thus the appropriate entry for state CV, for example should be $\{\lambda, V, CV\}$. Note that the strings of length less than n are predictable from strings of length n. Thus Table 3.2 really only shows the most informative elements of the set of suffixes of length up to two.

be merged, as well as states 3, 7 and 9. When these states are merged in the prefix tree in Figure 3.4, and transitions are preserved, the result is the acceptor shown in Figure 5.

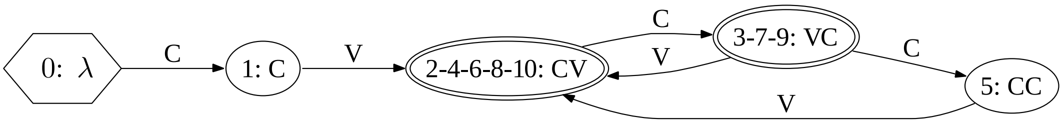

Figure 3.6: The Result of Merging States with Same Suffixes of Length Two in Figure 3.4

The machine in Figure 5 represents a generalization from the prefix tree in Figure 3.4. This machine in Figure 5 accepts an infinite number of words—only those words which obey the CV(C) syllabic template. Every word this machine rejects violates the syllabic template. Note that it is identical to the one in Figure 3.1. Thus, the learner generalizes exactly as desired.

In general, any language describable by an n-gram grammar can be learned by merging states (represented as prefixes) with the same suffixes of length $(n-1)$ in a prefix tree constructed from a (sufficient) sample. This is proven in Appendix C–4.3 below.

Although this learner is a batch learner, it is possible for a learner like the one above to be implemented in memoryless, online fashion, like the learner given in Table 3.1. In such a case state merging is interleaved with prefix tree building. A word is added, and then states are merged. Then the next word is added to the resulting structure and states are merged again (see Appendix C–4.3).

Finally, careful inspection will reveal that the online memoryless state merging learner makes the same generalizations at each point in time as the learner given Table 3.1. In other words, the iterative state merging learner is equivalent to the string extension learner. They are, in fact, two different descriptions of the same learning function.

The state merging learner does more however: it makes apparant a choice of how states are merged. It becomes natural to ask, what happens if only final states are merged? What happens if other equivalence criteria are used to merge states? Each conceivable merging strategy corresponds to some inductive principle which a learner can use to generalize from surface forms to some (regular) language and thus corresponds to some well-defined class of (regular) patterns.

Appendix C–4.1 shows how to represent n-gram grammars as finite state machines. Appendix C–4.2 provides a language theoretic characterization of languages recognizable by n-gram grammars which makes clear the inductive principle at work. Appendix C–4.3 provides a state merging learner which provably identifies the class of languages recognizable by an n-gram grammar (for fixed n).

6 Are N-grams Appropriate for Phonotactics?

It is reasonable to ask whether the grammars that we postulate to generate natural language phonotactic patterns are recognizable by n-gram grammars. In subsequent chapters, we show that long distance agreement phonotactic patterns and unbounded stress patterns do not have this property. However, even within the domain of contiguous segments, there is a sense in which n-gram grammars may not be explanatory adequate models.

6.1 N-gram Models Count to (n – 1)

The most striking thing about n-gram models is their capacity to count to $n - 1$. For example, an n-gram grammar is capable of expressing phonotactic rules such as the following:

- The next segment after a consonant must be a vowel.
- The second segment after a consonant must be a vowel.
- ...
- The $(n-1)$th segment after a consonant must be a vowel.

Our confidence that the statements describe plausible phontactic patterns decreases as we go down the list. Thus there seems to be a sense in which, if n-grams are an appropriate characterization of phonotactics, that n should be small, probably two or three. In this respect, it is useful to recall the claim that linguistic "rules do not count beyond two" (see Chapter 1 §1.1).

6.2 Resolvable Consonant Clusters

In his typological studies of word-inital onset clusters, Greenberg (1978) observes that an "overwhelming majority" (p. 250) of initial consonant clusters of length $n+1$ are *completely resolvable* in terms of initial clusters of length n. By completely resolvable, Greenberg means that there is an initial cluster of length $n+1$ iff there are two initial clusters of length n, whose (overlapping) concatenation, yields the $n+1$ length initial cluster. For example, in English, *str* is a legal initial consonant cluster. This cluster obeys Greenberg's hypothesis since *str* is resolvable by *st* and *tr*, both of which are legal initial consonant clusters in English.

Greenberg's survey of initial consonant clusters reveals languages which falsify the hypothesis that all initial consonant clusters of length $n+1$ are completely resolvable in terms of initial consonant clusters of length n. Therefore, he is careful to state this universal as a tendency as opposed to a true universal.[7] Nonethe-

[7] Greenberg's actual statement of the universal is weaker: an initial consonant cluster of length $n+1$ is partially resolvable by at least one initial consonsant cluster of length n. Greenberg observes there are exceptions even to this weaker claim.

less, given the "overwhelming majority" of languages for which this hypothesis is true, we can ask to what extent n-gram grammars are capable of reflecting this "overwhelming" tendency.

Notice that this hypothesis does not require that every possible resolvable cluster of length $n+1$ exist. Rather, the hypothesis just states that those which exist are resolvable as clusters of length n. Thus in English, although *st* and *tw* are both legal word-initial onset clusters, there are no words beginning with *stw*. Furthermore, 'blick' words such as [stwem] are considered marginal at best by native speakers of English (Clements and Keyser 1983). It is important to see that this is consistent with the above hypothesis because this hypothesis does not predict that all $n+1$ length clusters which are resolvable by n length should be acceptable. In the case of English *stw*, we conclude that there is some active phonotactic constraint prohibiting it.

On the other hand, it is a property of n-gram grammars that every sequence of length $n+1$ which is resolvable in terms of length n exists. In other words, any gap must be accidental, and not due to some active phonotactic constraint. For example, a bigram model of English initial consonant clusters predicts that *stw* is well-formed since the bigrams *st* and *tw* are attested and well-formed. In this very simple respect, we should be doubtful of the n-gram grammar's appropriateness as a phonotactic learner, even for patterns of contiguous segments.

It is true that increasing n in the n-gram grammar solves the problem above, but it creates others. Continuing the English example, if a trigram grammar is adopted instead of the bigram one, it becomes possible to develop a grammar which describes allowable English onsets exactly. However, it is also becomes possible to admit forms *stw* and exclude all other forms stX where X is any segment other than w![8] Thus there may be sequences of segments of length three which are

[8] It is even possible to exclude all grams of the form sXw, thus guaranteeing that *stw* is not

not completely resolvable in terms of segments of length two, a violation of the hypothesis.

6.3 Discussion

The main advantages of the hypothesis space employed by n-gram grammars and models is the well-structured hypothesis space which allows simple and tractable learners. The fact that the learners are simple and tractable is probably why n-gram models are widely used in many natural language processing tasks. There is little question that the n-gram learners upon which n-gram models are based are in a sense natural and sensible learners for n-gram languages.

The key issue that is often overlooked in usage of n-gram models in natural language processing tasks is whether natural language patterns are adequately describable by n-gram languages. Chomsky (1957) shows that n-gram languages are descriptively inadequate for syntactic patterns (he in fact shows no regular pattern is adequate). In this respect, it could be said that n-gram learners are poor learners for syntactic patterns because n-gram languages are poor approximations of syntactic patterns.

However, when we consider natural language phonotactic patterns over contiguous segments, it is unknown whether n-gram languages are appropriate. This really is the central issue. If they are, then the learners presented here become plausible hypotheses as to the kind of computations children make when acquiring such a pattern. The discussion above suggests that (1) n should be small and (2) that if people make use of n-gram hypothesis spaces that every completely resolvable cluster of length $n+1$ should be allowed. Although (1) seems reasonable on computational grounds (larger n requires greater computational resources), there is work

even partially resolvable, i.e. violating Greenberg's weaker stated universal.

which suggests certain n-gram languages with high values of n do not require large computational resources (Ron et al. 1996). The truth of (2) appears questionable on empirical grounds but certainly the empirical basis for Greenberg's universals need further study, both in the lab and in the field.[9]

If the n-gram languages turn out to be a poor hypothesis space for patterns over contiguous segments, the right thing to do is to look for alternative hypothesis spaces which correct the failures of the n-gram hypothesis space. String extension learning and state merging frameworks provide platforms where other inductive principles which lead to improved descriptive adequacy may be found.

7 Summary

This chapter introduced the categorical counterpart of n-gram models to make clear 1) that patterns over n or less contiguous segments are describable by n-gram languages 2) the character of n-gram languages and 3) the fundamental inductive principle learners of these language classes employ. It was shown that this inductive principle can be described in two ways: as an instantiation of string extension learning, or as a particular way of merging states of a finite state representation. The advantages of these two general learning schemes is that they provide (different) platforms in which other kinds of inductive principles which learn other language classes can also be stated. Figure 3.7 shows the trigram languages as a small subset of the regular languages which include patterns like *CCC.

[9] In this respect, recent work on initial consonant clusters in Slovakian languages promises to be revealing (Barkanyi 2007).

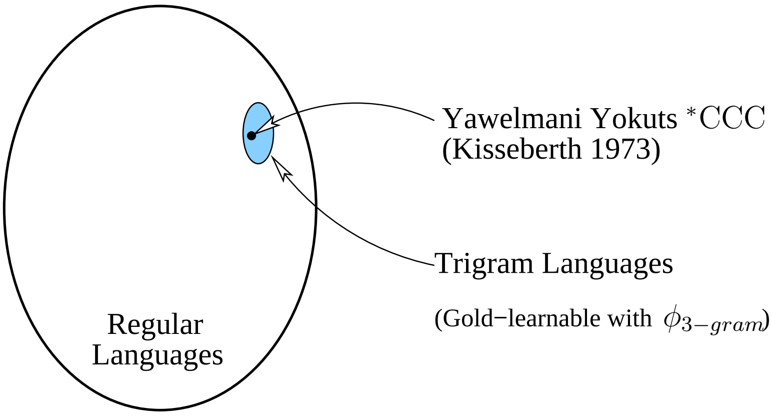

Figure 3.7: Trigram Languages

Appendices

C–1 String Extension Grammars

In this section, I describe a general class of functions $\mathcal{F}$. For each function in $\mathcal{F}$ is naturally associated with some formal class of grammars and languages. For reasons that will become apparant, these grammars are called *string extension grammars*. It follows straightforwardly from the definition of these functions and grammars that the corresponding language classes are closed under intersection, and that a learner which uses the function identifies the language class in the limit (Gold 1967). The learning procedure, given in §C–1.2, succeeds because each string in a language L is 'extended' by the function to aspects of a canonical representation of the grammar for L. This is why the learner is natural: it can only learn the class of languages with which the function is associated.

C–1.1 Definitions and Properties of $\mathcal{L}_f$

Consider some set A, and let a grammar G be a subset of A. Next consider $f : \Sigma^* \to 2^A$. Denote the class of functions which have this general form $\mathcal{F}$.

Definition 6 Define the language of grammar G_f to be

$$L(G_f) = \{w : f(w) \subseteq G\}$$

When f is understood from context, we just write G. Call the class of languages generated by grammars G_f which are subsets of A, $\mathcal{L}_f$.

As shown in more detail in §C–2 below, the function which maps a string to the n-grams (for some n) found in the string belongs to $\mathcal{F}$, and therefore defines a class of languages $\mathcal{L}_{n-gram}$ in the way described above. The following lemmas and theorems apply are stated generally, but it useful to keep in mind that the n-gram function (defined later) is an instantion of these more general results.

First, we show that the class of languages $\mathcal{L}_f$ has some structure.

Theorem 8 $\mathcal{L}_f$ is closed under intersection.

Proof: Consider any $L_1, L_2 \in \mathcal{L}_f$, and let G_1, G_2 be subsets of A which generate L_1, L_2 respectively.

(intersection) We show $L_1 \cap L_2 = L(G_1 \cap G_2)$. Consider any word w belonging to L_1 and L_2. Then $f(w)$ is a subset of G_1 and of G_2. Thus $f(w) \subseteq G_1 \cap G_2$, and therefore $w \in L(G_1 \cap G_2))$. Similarly, if we consider any $w \in L(G_1 \cap G_2)$, it means that $f(w)$ is a subset of both G_1 and G_2 and therefore $w \in L_1$ and $w \in L_2$ and so w is in their intersection. Thus $L_1 \cap L_2 = L(G_1) \cap L(G_2)$.

□

It will be useful to introduce a function γ which extends the domain of the function f from strings to languages (i.e. subsets of Σ^*) to A.

$$\gamma_f(L) = \bigcup_{w \in L} f(w)$$

When f is understood from context, we just write γ.

An element g of grammar G_f for a language L is *useful* iff $g \in \gamma_f(L)$. An element is *useless* if it is not useful. A grammar with no useless elements is called *canonical*. Clearly, there is a canonical grammar for every $L \in \mathcal{L}_f$. Now we can state another interesting property of the relation between the languages and grammars of $\mathcal{L}_f$.

Lemma 11 Let $L, L' \in \mathcal{L}_f$. $L \subseteq L'$ iff $\gamma(L) \subseteq \gamma(L')$

Proof: ($\Rightarrow$) Suppose $L \subseteq L'$ and consider any $g \in \gamma(L)$. Since g is useful, there is a $w \in L$ such that $g \in f(w)$. But $f(w) \subseteq \gamma(L')$ since $w \in L'$.

($\Leftarrow$) Suppose $\gamma(L) \subseteq \gamma(L')$ and consider any $w \in L$. Then $f(w) \subseteq \gamma(L)$ so by transitivity, $f(w) \subseteq \gamma(L')$. Therefore $w \in L'$. □

The significance of this result is that as the grammar G monotonically increases, the language of G monotonically increases too.

We can also now prove the following result, used in the next section on learning.

Theorem 9 For any $L_0 \subseteq \Sigma^*$, $L = L(\gamma(L_0))$ is the smallest language in $\mathcal{L}_f$ containing L_0.

Proof: First we show $L_0 \subseteq L$. Consider any $w \in L_0$. Since $f(w) \subseteq \gamma(L_0)$, $w \in L$ as well.

Now suppose $L_0 \subseteq L'$ for some $L' \in \mathcal{L}_f$. We show $L \subseteq L'$. Consider any $g \in \gamma(L_0)$. Thus, there is a $w \in L_0$ such that $g \in f(w)$. But $f(w) \subseteq \gamma(L')$ since $w \in L'$. Since g was arbitrary, $\gamma(L_0) \subseteq \gamma(L')$. Then by Lemma 11, it follows that $L \subseteq L'$. □

C–1.2 Natural Gold-learning of $\mathcal{L}_f$ for Finite A

We consider the case when A is finite. Note that this means the number of distinct grammars is $2^{|A|}$, which places an upper bound on $|\mathcal{L}_f|$. Therefore, there are many learners which can identify $\mathcal{L}_f$ in the limit (Osherson et al. 1986, Jain et al. 1999). What makes ϕ (below) a somewhat interesting learner for this class is that it is 'natural', in the sense that it uses f to acquire languages in $\mathcal{L}_f$. Now consider the learning function:

$$\phi(t[i]) = \begin{cases} \emptyset & \text{if } i = 0 \\ \phi(t[i-1]) & \text{if } t(i) = \epsilon \\ \phi(t[i-1]) \cup f(t(i)) & \text{otherwise} \end{cases}$$

The learner ϕ exemplifies *string extension learning*. Each individual string reveals, by extenstion with f, some aspects of the canonical grammar for L.

We now prove that this algorithm converges to the correct (canonical) grammar in the Gold framework (the key of course is the finiteness of $|A|$). The idea is that there is a point in the text in which every element of the grammar has been seen (because there are only finitely many useful elements of G and we are guaranteed to see a word for each element in $L(G)$ at some point since $|A|$ is finite). Thus at this point the learner ϕ is guaranteed to have converged to the target G (and hence L) as no additional words will add any more elements to the learner's grammar.

Lemma 12 For any text t and any $i \in \mathbb{N}$, $\phi(t[i]) = \gamma(\mathit{content}(t[i]))$.

Proof: Consider any $g \in \phi(t[i])$. By definition of ϕ, there is a $k \leq i$ such that $g \in f(t(k))$. It follows that $g \in \gamma(\mathit{content}(t[i]))$. Similarly, it follows from the definition of ϕ that for any $g \in \gamma(\mathit{content}(t[i]))$, there must be some $k < i$ such that $g \in f(t(k))$. Then, g belongs to $\phi(t[i])$ as well. □

Theorem 10 For all $L \in \mathcal{L}_f$, there is a finite sample S such that L is the smallest language in $\mathcal{L}_f$ containing S. We call S a *characteristic sample* of L in $\mathcal{L}_f$.

Proof: For $L \in \mathcal{L}_f$, construct the sample S as follows. For each $g \in \gamma(L)$, choose some word $w \in L$ such that $g \in f(w)$. Since $|\gamma(L)|$ is finite (since $|A|$ is finite), $|S|$ is finite. Clearly $\gamma(S) = \gamma(L)$ and thus $L = L(\gamma(S))$. Therefore, by Theorem 9, L is the smallest language in $\mathcal{L}_f$ containing S. $\square$

Theorem 11 If $|A|$ is finite then ϕ identifies $\mathcal{L}_f$ in the limit.

Proof: Consider any $L \in \mathcal{L}_f$. By Theorem 10, there is characteristic finite sample S for L. Thus for any text t for L, there is i such that $S \subseteq \mathit{content}(t[i])$. Thus for any $j > i$, $\phi(t(j))$ is the smallest language in $\mathcal{L}_f$ containing S, i.e. $\phi(t(j)) = \gamma(S)$ which equals $\gamma(L)$, by Lemma 12 and Theorem 10. $\square$

Also note that the learner ϕ is efficient in the length of the sample as long as f is efficiently computable in the length of a string.

C–2 A Formal Treatment of N-grams

In this section we show the function which maps a string to the set of n-grams found within it belongs to the class of functions $\mathcal{F}$ described in Appendix C–1. Consequently, it follows immediately that $\mathcal{L}_{n-gram}$ is closed under intersection, and that a very simple kind of learner identifies $\mathcal{L}_{n-gram}$ in the limit.

C–2.1 The N-Contiguous Set Function

Here I define the function CS_n which maps a string w in Σ^* to the set of n-grams which are found in the string. I call the function CS_n the *n-contiguity set* of w.

The words 'contiguity set' are meant to evoke the fact that n-grams are *contiguous* subsequences (of length n) of the string w. This will be contrasted in Chapter 4 with *precedence sets* which extract different kinds of information from a string w. (Recall that $V = \Sigma \cup \{\#\}$ and # is the word boundary symbol.)

Definition 7 For some $n \in \mathbb{N}$, define $CS_n : \Sigma^* \rightarrow 2^{V^{\leq n}}$ as follows:

$$
\begin{aligned}
CS_n(w) \quad = \quad & \{x \in V^n : \exists u, v \in V^* \text{ such that } \#w\# = uxv\} \text{ when } n \leq |w| + 2 \text{ and} \\
& \{\#w\#\} \text{ otherwise}
\end{aligned}
$$

Example 3 Consider $w = abc$. Then

$$CS_2(w) = \{\#a, ab, bc, c\#\}$$

Similarly, the 3-contiguity set induced by w is

$$CS_3(w) = \{\#ab, abc, bc\#\}$$

Finally, the 10-contiguity set of w is $\{\#abc\#\}$.

Also note that CS is an efficiently computable function in the length of a string.

C–2.2 N-Gram Grammars and N-gram Languages

N-gram grammars and languages are defined according to Appendix C–1.1. That is, for some n, a n-gram grammar G is a subset of $V^{\leq n}$. The language of a n-gram grammar is defined according Definition 6. In other words, a word w belongs to the language of G only if $CS_n(w) \subseteq G$.

Example 4 Let G be a bigram grammar equal to $\{\#a, aa, ab, b\#\}$. Then

$$L(G) = \{ab, aab, aaab, aaaab \ldots\}$$

Example 5 Let $\Sigma = \{a, b, c\}$ and consider

$$G = \left\{ \begin{array}{cccc} \#\#, & \#a, & \#b, & \\ a\#, & aa & ab, & ac, \\ & ba, & & bc, \\ & ca, & cb, & cc \end{array} \right\}$$

It is true that

1. Words in $L(G)$ begin with either a or b.
2. Words in $L(G)$ only end in a.
3. No word in $L(G)$ has a bb subsequence.
4. $\lambda \in L(G)$.

For fixed $n \in \mathbb{N}$, we denote the class of n-gram languages with $\mathcal{L}_{n-gram}$.

C–2.3 Properties of N-gram Languages

It follows from Definition 7 and Theorem 8 that $\mathcal{L}_{n-gram}$ is closed under intersection. Likewise it follows (from Lemma 11) that as a n-gram grammar monotonically increases, the corresponding n-gram language monotonically increases too.

Theorem 12 For fixed n, $\mathcal{L}_{n-gram}$ are closed under reversal but not complement.

Proof: (reversal) Consider any $L \in \mathcal{L}_{n-gram}$ and let G be the canonical grammar for L. We show that $L^r = L(G^r)$. Note that

$$(1)\ CS(w)^r = CS(w^r)$$

Consider any $w^r \in L^r$. First we show $CS(w^r) \subseteq G^r$. Consider any $g \in CS(w^r)$. It follows from (1) that $g^r \in CS(w)$. Since $w \in L$, $CS(w) \subset G$ and hence $g^r \in G$.

Therefore by definition, $g \in G^r$. Since g is arbitrary, $CS(w^r) \subseteq G^r$. Since for any $g \in G$, $g^r \in G^r$ by definition, it is the case that $CS(w^r) \subseteq G^r$. Since w^r is aribitrary, $L \subseteq L(G^r)$. Similarly, we can show $L(G^r) \subseteq L$. Since L is arbitrary it follows that $\mathcal{L}_{n-gram}$ is closed under reversal.

(not complement) Consider a n-gram grammar $G = \{(\#au, aub, ub\#)\}$ where $a, b \in \Sigma$ and $u \in \Sigma^{n-2}$. Then $L(G) = \{aub\}$ but since $\{aubaub\} \subset \Sigma^* - L(G)$, it clear that $G \subsetneq \gamma(\Sigma^* - L(G))$. Consequently aub also belongs to $L(\gamma(\Sigma^* - L(G)))$. □

At this point it is possible to prove that first n-gram specific result: that any language recognizable with an n-gram grammar is recognizable by a $(n+1)$-gram grammar, but the converse is false.

Theorem 13 Let $\mathcal{L}_{n-gram}$ denote that class of languages recognizable by n-gram grammars. Then $\mathcal{L}_{n-gram} \subset \mathcal{L}_{(n+1)-gram}$.

Proof: Let G_n denote a (canonical) n-gram grammar. It is sufficient to show (1) that for any G_n, there exists G_{n+1} such that $L(G_n) = L(G_{n+1})$ and (2) there exists a G_{n+1} such that $L(G_{n+1}) \notin \mathcal{L}_{n-gram}$.

Consider any G_n. Then construct G_{n+1} as follows:

$$G_{n+1} = \{a_1a_2 \ldots a_{n+1}) : a_1a_2 \ldots a_n \in G_n \text{ and } (a_2a_3 \ldots a_{n+1}) \in G_n$$

Consider any word $w \in L(G_n)$. Now consider any $x = x_1x_2 \ldots x_{n+1} \in CS_{n+1}(w)$. Clearly, $x_1x_2 \ldots x_n \in G_n$ and $x_2x_3 \ldots x_{n+1} \in G_n$ (since both are in $CS_n(w)$ and $CS_n(w) \subseteq G_n$). Thus by the construction above, $x \in G_{n+1}$. Since x is arbitrary, $w \in L(G_{n+1})$. And since w is arbitrary, $L(G_n) \subseteq L(G_{n+1})$. Likewise, consider any $w \in L(G_{n+1})$ and $x_1x_2 \ldots x_{n+1} \in CS_{n+1}(w)$. By the definition of G_{n+1},

$x_1x_2\dots x_n) \in G_n$ and $x_2x_3\dots x_{n+1}) \in G_n$. Since x, w arbitrary, $L(G_{n+1}) \subseteq L(G_n)$ and so $L(G_{n+1}) = L(G_n)$. Since G_n was arbitrary, $\mathcal{L}_{n-gram} \subseteq \mathcal{L}_{(n+1)-gram}$.

Finally, it is easy to find a a G_{n+1} such that $L(G_{n+1}) \notin \mathcal{L}_{n-gram}$. For example, consider the bigram grammar G in Example 4. There is no unigram grammar which recognizes $L(G)$. Therefore, $\mathcal{L}_{n-gram} \subset \mathcal{L}_{(n+1)-gram}$.

□

The consequence of this theorem is that as n increases the kinds of languages that can be described with n-gram grammars monotonically increase. It is an interesting question to ask what n is needed to adequately describe the patterns over contiguous segments in human languages. This is, as far as I know, an open question, though there is a general consensus that the number is small, probably about two or three. Another open, important question is whether there is any principled upper bound on n. A related question, whether n-gram models are appropriate for phonotactic learning and for discovery of phonotactic patterns over contiguous segments in particular, is addressed in §6.

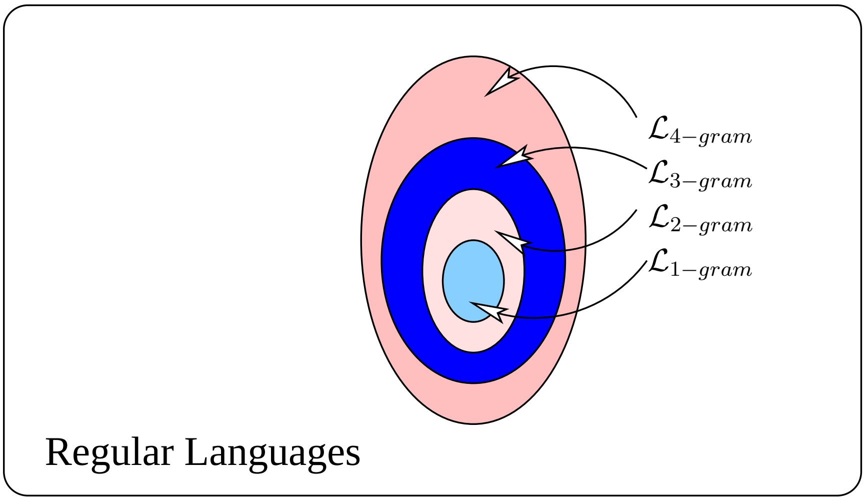

Figure 3.8: The N-gram Language Hierarchy

C–2.4 A Simple Learner

Finally, the results in Appendix C–1.2 tell us that the following learner identifies $\mathcal{L}_{n-gram}$ in the limit.

(5)

$$\phi(t[i]) = \begin{cases} \emptyset & \text{if } i = 0 \\ \phi(t[i-1]) & \text{if } t(i) = \epsilon \\ \phi(t[i-1]) \cup CS_n(t(i)) & \text{otherwise} \end{cases}$$

Since CS is an efficient function in the length of its input string, the learning function ϕ above is also efficient in the size of any given sample.

C–3 A Formal Treatment of State Merging

C–3.1 Prefix Trees

I denote the function which maps some finite sample S to a prefix tree which accept exactly S with PT.

Definition 8 $PT(S)$ is defined to be the acceptor (Q, I, F, δ) such that

$$\begin{aligned} Q &= \{Pr(S)\} \\ I &= \{\lambda\} \\ F &= \{S\} \\ \delta(u,a) &= ua \text{ whenever } u, ua \in Q \end{aligned}$$

Note that $PT(S)$ can be computed efficiently in the size of the sample S (Angluin 1982). $PT(S)$ can be computed batchwise from a sample S, or iteratively. When a word w is added to a prefix tree $PT(S)$, we speak of *extending* the prefix tree with w.

C–3.2 State Merging

This section is adapted from Angluin (1982). Let $A = (Q, I, F, \delta)$ be any acceptor. Any partition π of Q, defines another acceptor $A/\pi = (Q', I', F', \delta')$ defined as follows:

$$\begin{aligned} Q' &= \{B : B(q,\pi) \text{ such that } q \in Q\} \\ I' &= \{B : B(q,\pi) \text{ such that } q \in I\} \\ F' &= \{B : B(q,\pi) \text{ such that } q \in F\} \\ \delta'(B_0(q_0,\pi),a) &= \{B_1(q_1,\pi) : q_1 \in \delta(q_0,a)\} \end{aligned}$$

A/π is called the *quotient* of A and π.

Lemma 13 Let $A = (Q, I, F, \delta)$ be any acceptor and π any partition of Q. Then for all $p, q \in Q$, $u \in \Sigma^*$, if u transforms p to q then u transforms $B(p,\pi)$ to $B(q,\pi)$.

Proof: If A is the empty acceptor it is trivially true so assume that A is not empty. The proof is by induction. Since for any acceptor λ transforms any state to itself, it is true that if λ transforms q to q then λ transforms $B(q,\pi)$ to $B(q,\pi)$. Now assume that, for all strings u of length n, if u transforms q_0 to q_1 then u transforms $B(q_0,\pi)$ to $B(q_1,\pi)$. For the induction, assume that a string w of length $n+1$ transforms q_0 to q_1. Let $w = w_1 a$ so that $|w_1| = n$. Because $w = w_1 a$ transforms q_0 to q_1, there exists q_2 such that w_1 transform q_0 to q_2 and $q_1 \in \delta(q_2, a)$. By the inductive assumption, it is then true that w_1 transforms $B(q_0,\pi)$ to $B(q_2,\pi)$. It remains to be shown that $B(q_1,\pi) \in \delta'(B(q_2,\pi), a)$. This is so by definition of δ' since as noted $q_1 \in \delta(q_2, a)$. □

Theorem 14 Let A be any acceptor and π any partition of Q. Then $L(A) \subseteq L(A/\pi)$.

Proof: Let $A = (Q, I, F, \delta)$ be any nonempty acceptor. Let π be any partition of Q and let $A/\pi = (Q', I', F', \delta')$. Suppose A accepts u. Then u transforms some initial state q_i to some final state q_f. By Lemma 13 u transforms $B(q_i, \pi)$ to $B(q_f, \pi)$. Since $q_i \in I$ and $q_f \in F$, $B(q_i, \pi) \in I'$ and $B(q_f, \pi) \in F'$. Thus, A/π accepts u. □

The following lemma demonstrates how generalization may occur when merging states.

Lemma 14 Let $A = (Q, I, F, \delta)$ be any acceptor and π any partition of Q. For $p, q \in Q$, if u transforms p to q, and $B(p, \pi) = B(q, \pi)$ then for all $n \in \mathbb{N}$, u^n transforms $B(p, \pi)$ to itself in A/π.

Proof: This follows directly from Lemma 13 and Lemma 1. □

C–3.3 The State Merging Theorem

It has been proven that if a sample of words generated by some FSA is sufficient—that is, exercises every transition in this FSA—then there exists some way to merge states in the prefix tree to recover the generating FSA (Angluin 1982). Although we do not know which states should be merged, we are guaranteed that there is a way to merge such states to recover the original machine.

The theorem and a proof are given here (proof omitted in Angluin (1982)). First, there are some helpful definitions.

Definition 9 Let $A = (Q, I, F, \delta)$ be a tail canonical acceptor and let $w \in L(A)$. Then the *transition set* of w are those transitions in δ that make up the path of w through A (recall that there is a unique path since A is tail canonical). We denote the transition set of w in A with $TS_A(w)$.

Definition 10 Let $A = (Q, I, F, \delta)$ be a canonical finite-state acceptor. Then S is a sufficient sample of A iff $\bigcup_{w \in S} TS_A(w) = \delta$.

Pictorially, we can imagine, as A computes the path of some word w, coloring the states and transitions along this path. If a sample S is sufficient for a canonical acceptor, then every state and transition will be colored after every word in S is computed.

Theorem 15 Let $A = (Q, I, F, \delta)$ be a tail canonical finite state acceptor, S a finite sufficient sample of A, and $PT(S) = (Q_{PT}, I_{PT}, F_{PT}, \delta_{PT})$. Then there exists a partition π over Q_{PT} such that $PT(S)/\pi$ is isomorphic to A.

Proof: If $L(A)$ is empty then S is empty the result follows trivially so assume nonempty $L(A)$. The proof is in two parts. First we establish the equivalence relation which induces a partition π over $PT(S)$. Secondly, we show that A is isomorphic to $PT(S)$. For all $u \in Pr(L(A))$, denote $\delta(I, u)$ with q_u (i.e. the unique state in A that the prefix u leads to).

We say, for $p, q, \in PT(S)$, $p \sim q$ iff there exists $u, v \in Pr(S)$ such that $q_u = q_v$ and $\delta_{PT}(I_{PT}, u) = p$ and $\delta_{PT}(I_{PT}, v) = q$. Let π be the partition over Q_{PT} induced by $\sim$.

Now we show that $PT(S)/\pi$ is isomorphic to A. For any $u \in Pr(S)$, denote $B(\delta_{PT}(I_{PT}, u), \pi)$ with B_u (i.e B_u is the state one is led to in $PT(S)/\pi$ with u). Note that $u \in L(A)$ since $S \subseteq L(A)$. Thus we define h as follows: for all $u \in Pr(S)$,

$$h(B_u) = q_u$$

First we prove that $h : \pi \longrightarrow Q$ is a bijection. Consider $u, v \in Pr(S)$, $u \neq v$. If $B_u = B_v$, then $h(B_u) = h(B_v) = q_u = q_v$ since u, v are in the same block iff $q_u = q_v$

by the definition of $\sim$. If $B_u \neq B_v$, then $q_u \neq q_v$ and consequently $h(B_u) \neq h(B_v)$. So h is one-to-one. h is onto because S is a sufficient sample. To see this, consider any $q \in Q$. Since S is a sufficient, there is a word $w \in S$ and $u, v \in \Sigma^*$ such that $w = uv$ and $\delta(I, u) = q$ and $\delta(q, v) \in F$. Thus $u \in Pr(S)$ and $h(B_u) = q$.

Note that choosing aribtrary $u \in Pr(S)$ gives us an aribtrary block B_u in π and arbitrary $w \in S$ gives us an arbitrary final block B_w in π. Now, using this bijection h, we show that A is isomorphic to $PT(S)/\pi$. For ease of exposition, let $PT(S)/\pi = (Q', I', F', \delta'$.

First, recall $I = \{T_L(\lambda)\}$ and $I_{PT} = \{\lambda\}$ definitionally. Thus $I' = \{B_\lambda\}$. Now $h(B_\lambda) = q_\lambda = T_L(\lambda)$. Thus $h(I') = I$.

Second, consider any $w \in S$. Then $B_w \in F'$ and $h(B_w) = q_w \in F$ (since $S \subset L(A)$). Similarly, for any $q \in F$, there is a $w \in S$ (since S is sufficient) such that $\delta(I, w) = q$. Then $h(B_w) = q$ and so $h(F') = F$.

Finally, consider any $u \in Pr(S)$, $a \in \Sigma$. Then $h(\delta'(B_u, a)) = h(B_{ua}) = q_{ua} = \delta(q_u, a) = \delta(\delta(h(B_u), a)$. Thus the theorem is proved. □

The significance of this theorem should not be overlooked. Because the Gold framework purposefuly ignores the insufficient data problem, there is guaranteed to be a point where the learner has been exposed to a sample which exercises every transition in the target finite state grammar; i.e. which is sufficient. Thus the possibility is raised that for some restricted class of regular languages, there is some general strategy for state merging which learns that class.

C–4 Learning $\mathcal{L}_{n-gram}$ via State Merging

C–4.1 Finite State Representations

It is easy to create a finite-state acceptor for an n-gram grammar. Recall that $V = \Sigma \cup \{\#\}$.

Theorem 16 Given some $n \in \mathbb{N}$, and any $L \in \mathcal{L}_{n-gram}$, there exists a finite state acceptor A such that $L(A) = L$.

Proof: Construct $A = (Q, I, F, \delta)$ as follows, letting L denote $L(G)$. (Recall from §A–1.5 and §B–3.13.1 that $Sf^{\leq k}(L)$ are all the suffixes of length at most k of language L.)

$$\begin{aligned} Q &= Sf^{\leq n-1}(Pr(L)) \\ I &= \{\lambda\} \text{ if } L \neq \emptyset \text{ else } \emptyset \\ F &= Sf^{\leq n-1}(L) \\ \delta(xu, a) &= ua \text{ for all } a \in \Sigma, x \in \Sigma^{\leq 1} \text{ and } u \in \Sigma^* \text{ iff } xu, ua \in Q \end{aligned}$$

Note that the machine above is forward deterministic. If L is empty then $L(A)$ is empty so assume L is not empty. Consider any $w = x_1 x_2 \dots x_k \in L$. By definition of A, if $|w| \geq n-1$ then $\delta(I, w) = (x_{k-n+2} \dots x_k)$ (or $\delta(I, w) = w$, which is in $L(A)$). Now $(x_{k-n+2}, \dots x_k)$ is a final state since $(x_{k-n+2} \dots x_k) \in Sf^{n-1}(L)$ (because $w \in L$). Thus $w \in L(A)$. Similarly, if $w \in L(A)$, then $w \in L$ by the above construction so $L(A) = L$. □

The construction above essentially creates a state for every gram in $\gamma(L)$. These states are identified by suffixes of length $n-1$ (or less) of the prefixes of L.

Example 6 Here is how an FSA is constructed for a bigram grammar G, letting L stand for $L(G)$. Recall $\delta : Q \times \Sigma \rightarrow 2^Q$.

$$\begin{aligned} Q &= Sf^{\leq 1}(Pr(L)) \\ I &= \{\lambda\} \text{ if } L \neq \emptyset \text{ else } \emptyset \\ F &= Sf^{\leq 1}(L) \\ \delta(xu, a) &= v \text{ for all } a \in \Sigma, x \in \Sigma^{\leq 1} \text{ and } xu, v, \in \Sigma^* \text{ iff } u, v \in Q \text{ and } ua = v \end{aligned}$$

Consider the bigram grammar G below.

$$G = \{(\#, a), (\#, b), (a, \#), (a, a), (a, b), (b, a)\}$$

Then $A =$

$$\begin{aligned} Q &= \{\lambda, a, b\} \\ I &= \{\lambda\} \text{ iff } L \neq \emptyset \text{ otherwise } \emptyset \\ F &= \{a\} \\ \delta &= \{(\lambda, a, \{a\}), (\lambda, b, \{b\}), (a, a, \{a\}), (a, b, \{b\}), (b, a, \{a\})\} \end{aligned}$$

A drawing of A is shown in Figure 3.9.

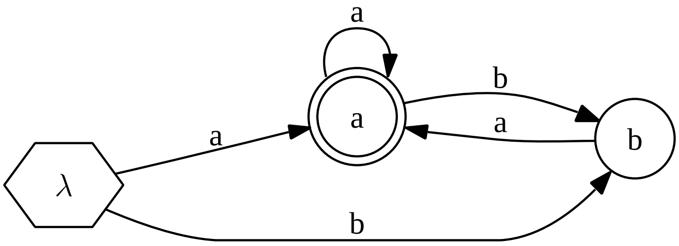

Figure 3.9: The FSA for the Grammar in Example 6.

As a consequence of this definition, it should be clear that most states a machine will have is $|\Sigma|^n - 1$. Note that the n-gram grammar which recognizes Σ^* (a one state canonical acceptor) uses all of these states! It is worth pointing out, however, that since the construction in Theorem 16 yields a deterministic machine, standard machine minimization algorithms can be applied, which are efficient (Hopcroft et al. 2001). However, below is a theorem which provides another way of obtaining the canonical finite state representation of an n-gram model.

C–4.2 Towards a Language-theoretic Characterization

The construction above illuminates another way to describe languages recognizable by n-gram grammars.

Theorem 17 For fixed n, consider $L \in \mathcal{L}_{n-gram}$. $\forall u, v \in Pr(L)$, if $\exists x \in \Sigma^{n-1}$ such that x is a suffix of u and x is a suffix of v, then $T_L(u) = T_L(v)$.

Proof: This follows directly from the construction in Theorem 16 and Corollary 4 in §B–3.10. □

Consider an example with a bigram model.

Example 7 Consider $L \in \mathcal{L}_{2-gram}$. $\forall u, v \in Pr(L)$, $T_L(u) = T_L(v)$ iff $\exists a \in \Sigma, u_1, v_1 \in \Sigma^*$ so that $u = u_1 a, v = v_1 a$.

This characterization of an n-gram languages makes clear the inductive principle used by the learners described earlier as illustrated in the next example. In the example below, we see the generalization on the basis of even a single word in a bigram grammar.

Example 8 Suppose $abcad \in L \in \mathcal{L}_{2-gram}$. Note $a, abca \in Pr(L)$. Also note that $bcad \in T_L(a)$ and $d \in T_L(abca)$. Let $u_1 = \lambda$ and $v_1 = abc$. Then $a = u_1 a$ and $abca = v_1 a$. By Theorem 17 $T_L(a) = T_L(abca)$. Consequently $bcad$ is also a good tail of $abca$ so $abcabcad \in L$. Likewise, $ad \in L$.

These inferences can be stated directly in the state merging model below.

Note that the converse of Theorem 17 is not true. Two prefixes with different suffixes could have the same tails. For example consider a bigram language $L = \{abc, adc\}$. Prefixes ab and ad have same tails, but not the same suffixes of length one. Further below we provide a complete language theoretic characterization of n-gram grammars.

C–4.3 Learning N-gram Languages by State Merging

Here we present a description of the learner ϕ in §C–2.4 in terms of state merging. Given some acceptor $A = (Q, i, F, \delta)$, consider the function which maps states q in Q to strings of length at most n by which state q could be reached; i.e. the function $I_n : Q \rightarrow \Sigma^{\leq n}$ defined below.

$$I_n(q) = \{w \in \Sigma^{\leq n} : \exists p \in Q \text{ such that } w \text{ transforms } p \text{ to } q\} \tag{3.1}$$

$I_n(q)$ can be thought of as the set of incoming paths to q. As mentioned in §A–1.3, this function induces an equivalence relation over the states Q (i.e. $p \sim q$ iff $I_n(p) = I_n(q)$). This relation, denoted $\sim_{I_n}$ is called the *incoming-n* equivalence relation. The incoming-n equivalence relation induces a partition π_{I_n} over Q. The blocks of this partition are merged to yield a new acceptor.

It is now possible to state the learner precisely. We give the learner in two versions, a batch learner and an iterative learner.

Algorithm 1 The N-gram State Merging Learner (batch version)

```
**Input:** a positive sample $S$ and a positive integer $n$.
**Ouput:** an acceptor $A$.
*Initialization*
Let $A_0 = (Q_0, I_0, F_0, \delta_0) = PT(S)$.
*Merging*
Compute $\pi_{I_{n-1}}$ over $Q_0$.
*Termination*
Let $A = A_0/\pi_{I_{n-1}}$ and output acceptor $A$.
```

Algorithm 2 is the iterative version of this algorithm.

Algrorithms 1 and 2 are guaranteed to converge to grammars which recognize

Algorithm 2 The N-gram State Merging Learner (iterative version)

```
Input: a positive sample $S$ and a positive integer $n$.
Ouput: an acceptor $A$.
Initialization
Let $A_0 = (\{q_0\}, \{q_0\}, \emptyset, \emptyset)$.
Let $i = 1$.
for all $w \in S$ do
    Let $A'_{i-1}$ be the extension of $A_{i-1}$ with $w$.
    Compute $\pi_{I_{n-1}}$ over $Q'_{i-1}$.
    Let $A_i = A'_{i-1}/\pi_{I_{n-1}}$.
    Increase $i$ by 1.
end for
Termination
Output acceptor $A_i$.
```

the target language, provided S is a sufficient sample.

Lemma 15 or any $q \in Q_{PT}$, $I_n(q) = Sf^{\leq n}(\{u\})$ where u transforms I_{PT} to q in $PT(S)$.

Theorem 18 Given any sample S, $PT(S)/\pi_{I_{n-1}}$ is isomoprhic to the acceptor A constructed to Theorem 16 for $L(\gamma_n(S))$.

Proof: Let $PT(S) = (Q_{PT}, I_{PT}, F_{PT}, \delta_{PT})$ and let A, the acceptor constructed according to Theorem 16 for $\gamma(S)$, equal (Q, I, F, δ). For any $u \in Pr(S)$, denote $B(\delta_{PT}(I_{PT}, u), \pi)$ with B_u. Let $sf^n(u)$ equal u if $|u| \leq n$, otherwise $sf^n(u) = v$ such that $|v| = n$ and $xv = u$ for some x in Σ^*. In other words $sf(u)$ returns the longest suffix of u up to length n.

Then the bijection we need is:

$$h(B_u) = sf^{n-1}(u)$$

I omit the rest of the proof. □

Corollary 10 $PT(S)/\pi_{I_{n-1}}$ is the smallest n-gram grammar containing S.

Corollary 11 Algorithms 1 and 2 identify $\mathcal{L}_{n-gram}$ in the limit.

Note that $\sim_{I_n}$ is actually a *stronger* equivalence relation than needed because prefix tree construction guarantees that for every state p, $|I_n(p)| = 1$.

C–4.4 Obtaining the Canonical FSA for a N-gram Grammar

Above, we saw that the finite state representation for a n-gram grammar grows quite large with respect to n. This section compiles a few notes about how to obtain the smallest forward determinstic acceptor which accepts the same language as the n-gram grammar.

The following lemma follows from Theorem 17.

Lemma 16 Consider $L \in \mathcal{L}_{n-gram}$. $\forall u, v \in Pr(L)$, $T_L(u) = T_L(v)$ iff $Pr^{\leq k}(T_L(u)) = Pr^{\leq k}(T_L(v))$.

Proof: Consider $L \in \mathcal{L}_{n-gram}$. The ($\Rightarrow$) direction is trivial so for any $u, v \in Pr(L)$, suppose $Pr^{\leq n-1}(T_L(u)) = Pr^{\leq n-1}(T_L(v))$. We show $T_L(u) = T_L(v)$ by showing that for any $x \in Pr^{\leq n-1}(T_L(u))$, $T_L(ux) = T_L(vx)$.

Consider any $x \in Pr^{\leq n-1}(T_L(u))$ such that $|x| = n - 1$. Since x is also in $Pr^{\leq n-1}(T_L(v))$, both ux and vx belong to $Pr(L)$. Then by Theorem 17, $T_L(ux) = T_L(vx)$.

Now consider any $x \in Pr^{\leq n-1}(T_L(u))$ such that $|x| < n-1$ and $ux \in L$. (If $ux \notin L$, then there must be some $y \in \Sigma^*$ such that $|xy| = n-1$ because $ux \in Pr(L)$. This case was handled above.) But x is also in $Pr^{\leq n-1}(T_L(v))$ and so vx also belongs L. In other words, whenever λ belongs to $T_L(ux)$, it also belongs to $T_L(uv)$.

Since x is arbitrary, $T_L(u) = T_L(v)$. Since u, v are arbitrary, the theorem is proved. □

Given some acceptor $A = (Q, i, F, \delta)$, consider the function $O_n : Q \to \Sigma^{\leq n}$ defined below.

$$O_n(q) = \{w \in \Sigma^{\leq n} : \exists p \in Q \text{ such that } w \text{ transforms } q \text{ to } p\} \tag{3.2}$$

$O_n(q)$ can be thought of as the set of outgoing paths to q. As mentioned in §A–1.3, this function induces an equivalence relation over the states Q (i.e. $p \sim q$ iff $O_n(p) = O_n(q)$). This relation, denoted $\sim_{O_n}$ is called the *outgoing-n* equivalence relation. The outgoing-n equivalence relation induces a partition π_{O_n} over Q.

The idea is that the finite state representation of a bigram grammar can be made canonical by merging the blocks of this partition π_{O_n}. Before we can prove this, we will need the following lemma, which holds for any forward deterministic acceptor.

Lemma 17 Denote $L(A)$ with L and suppose A is forward determinstic. For any $u \in Pr(L)$, denote with q the unique state in $\delta(I, u)$. Then $O_n(q) = Pr^{\leq n}(T_L(u))$.

Proof: If L is empty then this follows vacuously so assume L is not empty. Consider any $x \in O_n(q)$. Thus, there is a (unique) $p \in \delta(q, x)$ and since $\delta(I, u) = q$, $\delta(I, ux) = p$. Consequently $ux \in Pr(L)$ and $x \in T_L(u)$. Since $|x| \leq n$, $x \in Pr^{\leq n}(T_L(u))$ by definition. Hence, $O_n(q) \subseteq Pr^{\leq n}(T_L(u))$.

Now consider any $x \in Pr^{\leq n}(T_L(u))$. Therefore, there is a $v \in \Sigma^*$ such that $xv \in T_L(u)$ and $uxv \in L$. Given that $\delta(I, u) = q$ and A is forward deterministic, there is a unique p in $\delta(q, x)$. Since $|x| \leq n$, $x \in O_n(q)$ by definition. Hence $Pr^{\leq n}(T_L(u)) \subseteq O_n(q)$, and consequently $O_n(q) = Pr^{\leq n}(T_L(u))$. □

Theorem 19 Let $L \in \mathcal{L}_{n-gram}$ and A be the acceptor for L, constructed according to Theorem 16. Then the tail canonical acceptor for L, denoted $A_T(L)$, is isomorphic to $A/\pi_{O_{n-1}}$.

Proof: Omitted. □

Consequently, we have the following language theoretic characterization of n-gram languages.

Corollary 12 For fixed n, consider $L \in \mathcal{L}_{n-gram}$. $\forall u, v \in Pr(L)$, $T_L(u) = T_L(v)$ iff $\exists x \in \Sigma^{n-1}$ such that x is a suffix of u and x is a suffix of v or $Pr^{n-1}(T_L(u)) = Pr^{n-1}(T_L(v))$.

CHAPTER 4

Patterns Over Non-contiguous Segments

1 Overview

This chapter presents an inductive principle that demonstrates how Long Distance Agreement (LDA) patterns can be learned from limited experience. Long Distance Agreement patterns are patterns in which segments which are noncontiguous within a word, agree or disagree in some phonological feature (to be defined more carefully below). LDA patterns have been thought to be difficult to learn because of the observation that arbitrarily many segments may intervene between agree-ers (see below). For example, Albright and Hayes (2003a) observe that "the number of logically possible environments. . . rises exponentially with the length of the string." There are thus potentially too many environments for a learner to consider when trying to discover LDA patterns. However, the idea put forward here is that "arbitrarily many" does not require a learner to consider every logically possible nonlocal environment. This chapter presents a learnable hypothesis space for LDA patterns where "arbitrarily many" is interpreted to mean "no sense of distance at all."

The inductive principles introduced in this chapter operate on the notion of precedence. Precedence here means *precedes at any distance* and is not be confused with immediate precedence, for which I use the term *contiguous with.* Learners with this notion of precedence cannot distinguish different degrees of distance because they cannot count at all. These learners only distinguish which segments may precede other segments and are thus able to learn long distance agreement patterns from positive evidence easily because the resulting hypothesis space is sufficiently small and well-structured.

§2.1 gives a brief typological survey of LDA patterns and explains why n-gram models are inadequate for learning patterns of this type. §3 defines precedence grammars which recognize LDA patterns. Since precedence grammars, like n-gram grammars, are string extension grammars (see §C–1), string extension learning guarantees identifiability in the limit. §4 shows how the string extension learner can be instantiated as a state merging model. §5 discusses to what extent these results explain other typological observations about LDA patterns, and shows how the learner can be combined with n-gram learning to learn phonotactic both LDA constraints and constraints over contiguous segments. §7 summarizes the chapter.

2 Long Distance Agreement

In their seminal typological studies of consonant harmony, Hansson (2001) and Rose and Walker (2004) define Long Distance Agreement as follows.

(1) Long Distance Agreement (LDA) patterns are those within which particular segments, separated by at least one other segment, must agree or disagree in some phonological feature.

Hansson (2001) adds to this definition the following:

(2) The intervening segments between the agreeing segments are not audibly affected by the agreeing feature.

(2) is necessary in order to clearly distinguish LDA from languages which exhibit patterns known as 'feature spreading'. Feature spreading describes patterns where arbitrarily long sequences of contiguous segments agree in some phonological feature. The classic example is nasal spreading. For example, in the Johore dialect of Malay oral vowels and glides may not contiguously follow a nasal consonant, nasalized vowel, or nasalized glide. (Onn 1980, Walker 1998, 2003). Consequently, there are words like [peŋãw̃ãsan] 'supervision', but none like *[peŋawasan] nor *[peŋãwasan]. Although it is true in [peŋãw̃ãsan] that [ŋ] and the second [ã] agree in the feature nasal and are separated by two intervening segments, the intervening segments are not arbitrary since they participate in the agreement as well. Hence, this is not a case of LDA. I review the issues surrounding LDA patterns and spreading patterns for phonotactic learning below in §2.1.

2.1 Kinds of LDA

2.1.1 Consonantal Harmony

The extensive surveys by Hansson (2001) and Rose and Walker (2004) establish many different kinds of consonantal long distance agreement. They provide examples of sibilant harmony (see below), liquid harmony, dorsal harmony, nasal harmony, and voicing harmony, among others. Although some cases appear to have a grammatical constraint limiting the distance at which the agreement may apply (see §5.2 below), many cases have no such limitations. In other words, it typically does not matter how many segments intervene between two segments which are subject to agreement.

As an example, recall the classic case of Navajo Sibilant Harmony (Sapir and Hojier 1967, Fountain 1998) described in Chapter 2 (and repeated here), in which sibilants agree in the feature [anterior]. At the segmental level, no words exist which contain two segments with different values of anteriority. That is, no word contains a sound from the set of [+anterior] sibilants in Navajo [s,z,ts,ts',dz] and a sound from the [-anterior] sibilant set [ʃ,ʒ,tʃ,tʃ',dʒ]. (3) shows well-formed words with sibilants obeying the agreement and (4) shows ill-formed words which disobey the agreement.

(3) a. **ʃ**iːteː**ʒ** 'we (dual) are lying'
b. da**s**doːli**s** 'he (4th) has his foot raised'

(4) a. ***ʃ**iːteː**z** (hypothetical)
b. *da**s**doːli**ʃ** (hypothetical)

There are two key observations that the definition of LDA captures. The first is that the agreement holds at arbitrary distances. The second is that the intervening segments are unaffected by the feature [anterior]. (This second fact is not uncontroversial, I return to it below.) These facts are captured in the statements in (5) which summarize the LDA pattern in Navajo.

(5) 1.[-anterior] sibilants are never preceded by [+anterior] sibilants.
2.[+anterior] sibilants are never preceded by [-anterior] sibilants.

Other consonantal harmony patterns can be described similarly. For example, in Yaka (Bantu) (Hyman 1995), voiced consonants are never preceded by nasals in stems. Thus [míːtuk-ini] 'sulk' is well formed but not *[**m**íːtuk-i**d**i].[1] Statements like the one in (5) provide the key to the learner presented later in this chapter.

There is debate whether intervening segments are affected in long distance agreement patterns. The 'strict locality' hypothesis that they are affected, articulated

[1] See Hyman (1995) for arguments against an allomorphic analysis.

clearly by Gafos (1999), is essentially the hypothesis that all apparent cases of LDA are actually spreading. Indeed, at least one case originally claimed to be LDA does appear to be an instance of spreading upon closer inspection, Kinyarwanda (Mpiranya and Walker 2005, Byrd et al. 2006, Walker 2007). Whether all cases of LDA are actually spreading remains an open question. The case of nasal consonantal harmony in Yaka appears particularly difficult for a theory equating LDA with spreading because vowels in Yaka are nasalized and voiceless consonants can occur between the agreeing nasals (as in the example above) (Hyman 1995). This issue recurs in the next section and I review the consequences the outcome of the debate has for a theory of phonotactic learning in §2.3.

2.1.2 Vowel Harmony

The definition of LDA above in (1) is not inherently restricted to consonants. Languages which require vowels to agree in some feature can also be plausibly treated as long distance agreement. Consider the the following definitions of vowel harmony:

- "...we can define a vowel harmony language as any language containing at least two sets of vowels which cannot co-occur within the same ... word..." (Ringen 1988:1)
- "Vowel harmony is the phenomenon observed in some languages by which all the vowels in a word ... must bear the same value of some vocalic feature."(Baković 2000:1)
- "I regard vowel harmony as the phenomenon ... where potentially all vowels ... within a domain like the phonological or morphological word ... systematically agree with each other with regard to one or more articulatory features." (Krämer 2003:3)

Nothing in the definition of LDA in (1) excludes vowel harmony from consideration. Hansson (2001) and Rose and Walker (2004) focus exclusively on consonantal harmony, and leave open the possibility that their analyses may extend to the domain of vowel harmony (see also Hansson (2006)).

There are two acknowledged problems when analyzing vowel harmony as long distance agreement. The first is that vowel harmony can be analyzed as spreading provided that consonants participate. This assumption is standard; e.g. Baković (2000:6) adopts the assumption that "consonants fully participate in vowel harmony." The second is that in most languages, the distribution of vowels require that they occur frequently. In other words, for independent reasons, vowels typically do not occur arbitrarily far apart as appears to be the case for sibilants in Navajo. Thus, because there is some independently motivated bound on the distance that separates vowels, it becomes more difficult to defend the statement that there is no principled bound on how far apart agreeing vowels may be. For these two reasons, it is less clear whether vowel harmony constitutes LDA.

There are some cases of vowel harmony which suggest that the two problems above do not, at least, hold for all cases of vowel harmony. For example, vowel harmony cases with so-called transparent or neutral vowels are claimed to exist. Hansson (2007) provides a review of the most relevant cases, (see also Baković (2000), Krämer (2003) and references therein). In these cases, it has been claimed that the agreeing vowels are separated by vowels which do not participate in the harmony. If arbitrarily many neutral vowels may occur between agreeing vowels, then the resulting pattern meets the definition of LDA. Nonetheless, the debate is ongoing as recent work calls into question whether neutral vowels are really neutral (Gordon 1999, Chiosáin and Padgett 2001, Recasens et al. 2003, Gick et al. 2006).

2.2 Inadequacy of N-gram Models

Because there is no principled upper bound on how many segments may intervene between agreeing segments in a LDA pattern, the n-gram hypothesis space is clearly inadequate to describe patterns of this type. There is no value for n which adequately describes any LDA pattern because it is always possible to imagine a word in which the agreeing segments are separated by $n + 1$ segments and consequently whatever n-grams are present in such a grammar cannot enforce the necessary agreement at this longer distance. Consequently, some other hypothesis space will have to be employed which at least contains these patterns.

2.3 Long Distance Agreement or Spreading?

The debate about whether LDA patterns can be analyzed as spreading has consequences for a theory of phonotactic learning. This is because spreading patterns can be described with a simple bigram grammar and thus learned by a bigram learner. If it is shown that in the Navajo sibilant harmony pattern, for example, the intervening segments are audibly different between agreeing segments than otherwise, then n-gram based learners become viable. Whether the spreading hypothesis is correct and cases of true LDA exist is an issue that will be empirically resolved with careful examination of every case of attested LDA.

The work here does not weigh in on this debate one way or the other. I only wish to show that true LDA patterns can be learned simply and effectively in the manner described below. I do this because I do not think it has been established beyond reasonable doubt that all cases of LDA are actually spreading (see also Hansson (2007)). Indeed, cases like Yaka (Hyman 1995) suggest not all cases can be explained by spreading.

3 Precedence Grammars

3.1 The Basic Idea

As can be seen from the definition of LDA patterns given above, the distance between the segments subject to agreement does not matter. This crucial observation—that the learner need not distinguish how many intervening segments there are at all—is the key to developing a learnable hypothesis space which contains LDA patterns.

Recall the case of Navajo sibilant harmony. We can state the Navajo sibilant harmony rule in the following manner, without reference to features (cf. (5)):

(6)
1. No element of {s,z,ts,ts',dz} is ever preceded by an element of {ʃ,ʒ,tʃ,tʃ',dʒ}.
2. No element of {ʃ,ʒ,tʃ,tʃ',dʒ} is ever preceded by an element of {s,z,ts,ts',dz}.

A precedence grammar is simply a set of the allowable pairs (a, b) in the language such that a is allowed to precede b in a word.[2] Here precedence does not mean immediate precedence, but precedence at any distance. Words are well-formed in the language of the precedence grammar iff every precedence relation in the word exists in the grammar. In this way, a precedence grammar is the similar to a bigram grammar (a list of pairs of alphabetic symbols), but the pairs in these sets have different interpretations.

Thus precedence grammars, like n-gram grammars, are string extension gram-

[2] A similar idea has been proposed to explain human performance in certain orthographic processing tasks (Schoonbaert and Grainger 2004, Grainger and Whitney 2004, Whitney 2001, Whitney and Berndt 1999) and carries the term 'open bigram' (see also Dehaene et al. (2005)). Precedence grammars may be considered a set of open bigrams. The ideas in these pages developed independently and I choose to use the word 'precedence' because I find the properties of these formal languages more transparently related to the notion of precedence than openness.

mars (formally defined in Appendix C–1, see also Chapter 3 §2). The only difference is that *n*-gram grammars make use of a function which returns the *n*-grams of a word, whereas precedence grammars make use of a function which returns the precedence relations of a word.

An example makes the idea clear. To simplify the discussion of Navajo, consider a fragment of Navajo—that is, the Navajo pattern restricted to an alphabet of four symbols: {s,ʃ,t,o}. Then consider the grammar in (7).

(7)

$$G = \left\{ \begin{array}{cccc} \text{(s,s)} & & \text{(s,t)} & \text{(s,o)} \\ & \text{(ʃ,ʃ)} & \text{(ʃ,t)} & \text{(ʃ,o)} \\ \text{(t,s)} & \text{(t,ʃ)} & \text{(t,t)} & \text{(t,o)} \\ \text{(o,s)} & \text{(o,ʃ)} & \text{(o,t)} & \text{(o,o)} \end{array} \right\}$$

The language of G includes a word like [sotos] because every precedence relation which exists in [sotos] is present in the grammar. For example, [s] precedes [t] in this word and (s,t) is in the grammar. In fact, every precedence relation in [sotos] is in the grammar G above. On the other hand, the language of G excludes a word like [sotoʃ]. The reason is that [s] precedes [ʃ] in [sotoʃ] but crucially (s,ʃ) is not in the grammar G. Since neither (s,ʃ) nor (ʃ,s) belong to the grammar G, the language of this grammar rejects any word in which [s] precedes (at any distance) [ʃ] and vice versa. Since every other possible precedence relation for this fragment is present in the grammar, however, every other possible word belongs to the language of G. In this way, the language of this grammar faithfully reproduces the phonotactic constraint: every word in the language obeys the constraint, and every word not in the language disobeys it.

Thus the grammar G above recognizes exactly the same language as the finite-state acceptor in Figure 4.1 (repeated from Figure 2.3). Every word the machine accepts obeys the sibilant harmony rules, and every word the machine rejects violates it. Note again that this machine, like the grammar G above, isolates the sibilant harmony phonotactic from other phonotactic constraints such as ones governing syllable structure. Thus the machine in Figure 4.1, like the grammar G above, accepts ill-formed Navajo words like those with several adjacent consonants or several adjacent vowels (e.g. [soooooooooooos]). Such words are ill-formed, however, because of other phonotactic constraints, and not the sibilant harmony constraint. There is an efficient procedure for writing a precedence grammar like the one in (7) above as a finite state acceptor, as shown in Appendix D–2.1.

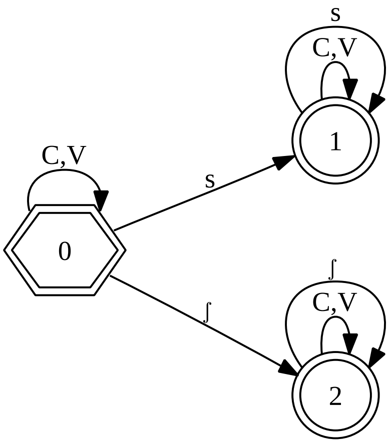

Figure 4.1: Navajo Sibilant Harmony

3.2 Learning Precedence Languages

It is also easy to see how languages of precedence grammars can be obtained from a list of finite examples. Like the n-gram learner, the procedure here is an example of string extension learning. The initial state of the learner's grammar is empty. All the learner does is record the precedence relations in observed words. For example, Table 4.1 below shows how the grammar grows upon observing the sequence *tosos*, *ʃotoʃ*, *stot*. New precedence relations added to the grammar are given in bold.

time	word	Precedence Relations	Grammar
0			$\emptyset$
1	tosos	(t,o) (t,s) (o,s) (o,o) (s,s)	$\left\{\begin{array}{cc} \textbf{(s,s)} & \textbf{(s,o)} \\ & \\ \textbf{(t,s)} & \textbf{(t,o)} \\ \textbf{(o,s)} & \textbf{(o,o)} \end{array}\right\}$
2	ʃotoʃ	(ʃ,o) (ʃ,t) (ʃ,ʃ) (o,t) (o,o) (o,ʃ) (t,o) (t,ʃ)	$\left\{\begin{array}{cccc} \text{(s,s)} & & & \text{(s,o)} \\ & \textbf{(ʃ,ʃ)} & \textbf{(ʃ,t)} & \textbf{(ʃ,o)} \\ \text{(t,s)} & \textbf{(t,ʃ)} & & \text{(t,o)} \\ \text{(o,s)} & \textbf{(o,ʃ)} & \textbf{(o,t)} & \text{(o,o)} \end{array}\right\}$
3	stot	(s,t) (s,o) (t,o) (t,t) (o,t)	$\left\{\begin{array}{cccc} \text{(s,s)} & & \textbf{(s,t)} & \text{(s,o)} \\ & \text{(ʃ,ʃ)} & \text{(ʃ,t)} & \text{(ʃ,o)} \\ \text{(t,s)} & \text{(t,ʃ)} & \textbf{(t,t)} & \text{(t,o)} \\ \text{(o,s)} & \text{(o,ʃ)} & \text{(o,t)} & \text{(o,o)} \end{array}\right\}$

Table 4.1: Precedence Learning Navajo Sibilant Harmony

On the basis of these few forms, the learner already generalizes tremendously. It accepts words like [ʃoʃ], [ʃtot], and [sototos] but not words like [ʃtos] or [sosoʃ]. Since the grammar G in (7) only generates words which obey the sibilant harmony constraint, no additional words add any precedence relations to the grammar in the last time step in Table 4.1. In this way, the learner which records precedence relations identifies the language of the grammar G in the limit because it is guaranteed to converge after seeing finitely many forms (since it is guaranteed to see a word which instantiates every precedence relation at some point).

It is also possible to characterize which samples are sufficient for successful learning precedence languages. A sample is sufficient provided, for every precedence relation in the target grammar, there is some word in the sample with this precedence relation. Consequently, we can ask whether there are any 'accidental gaps' for precedence relations in the linguistic environments of children (see related discussion in §5.1). If there are, then again the options are to find some alternative formal hypothesis space, or see whether all such cases of 'accidental gaps' can be resolved upon appeal to features.[3]

The learning procedure outlined above, which I call the precedence learner, is tractable. This is because the number of precedence relations in a word is given by a quadratic function in the length of the word. Furthermore, the value of this function is bounded from above by the square of the number of alphabetic symbols (see Appendix D–1.1).

[3] The situation is analgous to the problem for n-gram languages. So another possibility is to develop smoothing methods for the precedence language hypothesis space. See Jurafsky and Martin (2000) for more about smoothing.

3.3 Local Summary

Any LDA pattern like sibilant harmony in Navajo can be described with a precedence grammar. This is because it is possible to describe any list of LDA constraints of the form: segment b cannot (anywhere) follow segment a. Note that if $b = a$ then long distance Obligatory Contour Principle (OCP) type patterns can also be described (Leben 1973, Goldsmith 1976, McCarthy 1979, 1981, Frisch et al. 1995). Also, any LDA with no blocking pattern can be learned efficiently in the manner described above (see Appendix D–1).

4 Learning Precedence Languages by State Merging

Like the n-gram learner in the last chapter, the learner in §3.2 above can be described as a state merging strategy. However, unlike the n-gram learner in which both the state merging and string extension versions were relatively simple, the state merging learner which is equivalent to the string extension precedence learner is fairly complex. This section illustrates this learner and explains the equivalence relation used to merge states. Additional steps are necessary to make the state merging learner equivalent to the string extension precedence learner. A formal treatment is given in Appendix D–1.4.

The basic idea of the state merging learner for precedence grammars is to (1) construct a prefix tree, and (2) merge states whose corresponding prefixes have the same range. The range of a string is simply the set of symbols present in the string. In a sense, every segment is adjacent to every preceding segment.

For example, Figure 4.2 shows the prefix tree for the words {tosos,ʃotoʃ, stot}. The table below shows the range for each prefix (and its associated state). From Table 4.2, is clear that the state merging learner merges states 3, 4, 5, 13, and 14

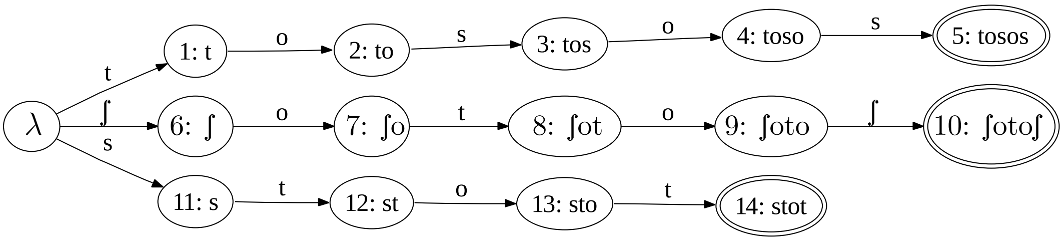

Figure 4.2: The Prefix Tree for Words {tosos, ʃotoʃ, stot}

	state	range
0	λ	$\emptyset$
1	t	{t}
2	to	{o,t}
3	tos	{o,s,t}
4	toso	{o,s,t}
5	tosos	{o,s,t}
6	ʃ	{ʃ}
7	ʃo	{o,ʃ}
8	ʃot	{o,ʃ,t}
9	ʃoto	{o,ʃ,t}
10	ʃotoʃ	{o,ʃ,t}
11	s	{s}
12	st	{s,t}
13	sto	{o,s,t}
14	stot	{o,s,t}

Table 4.2: The Range for States in the Prefix Tree in Figure 4.2

into a single state, as well as states 8, 9, and 10. The result is in Figure 4.3.

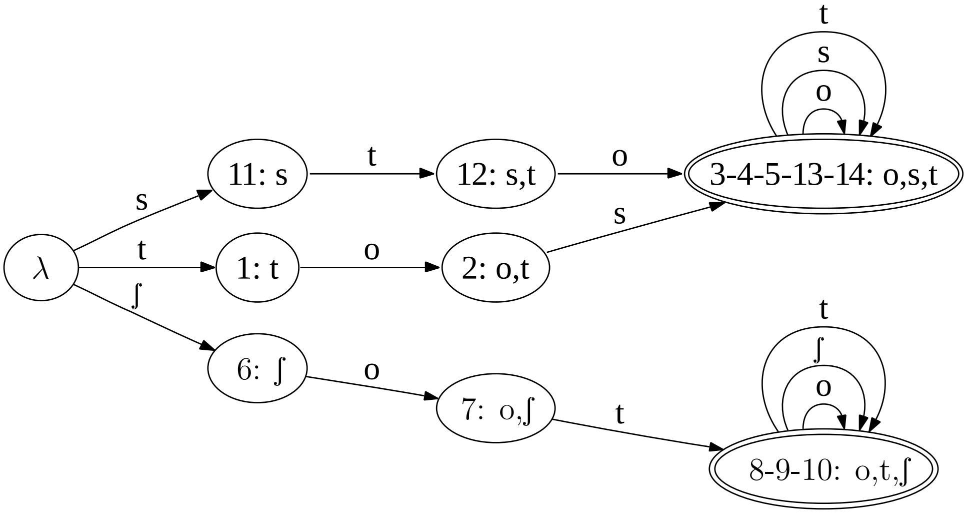

Figure 4.3: The Result of Merging Same-range States in Figure 4.2

This state merging learner as described is a batch learner, but as was the case with the n-gram learner, it can also be implemented in a memoryless online fashion by simply interleaving the prefix tree building and state merging steps.

Careful readers will note that the generalizations obtained at each stage of the state merging online memoryless learner are not the same as the precedence learner presented in §3.2. The state merging learner is guaranteed to converge in the limit, but is in general much slower.[4] More precisely, the sample the state merging learner needs to converge to the correct grammar is much larger than the sample needed by the precedence learner.

The reason the state merging learner requires a larger, richer sample is because it does not take advantage of all the properties a precedence language has. There are simple modifications that can be made to the state merging learner that do take advantage of these properties. For example, it can be shown that every state in finite

[4] This is because the class of languages obtained by merging same-range states in prefix trees built from subsets of Σ^* is strictly larger than the precedence languages.

state representations of precedence grammars is final (see Appendix D–1.3). Making every state final in the above machine increases the amount of generalizations made at each time step (because fewer distinctions are made).

Also, in the appendix it is shown that 'later' states inherit only those outgoing transitions that occur at 'earlier' states (see Appendix D–1.3). Consequently, because state '3-4-5-13-14' has three outgoing transitions (with labels [t,s,o]) it can be inferred that each state represented by some subset of the set of symbols representing this state also has those same outgoing transitions. Thus, for example, we can add two outgoing transitions with labels [t,s] to state 12 because this state represents the set {s,t}, a subset of {o,s,t}. Since state 12 represents the state in which only segments of the set {s,t} have been observed, both transitions will loop back to state 12. By the same principle, state 11 must also have outgoing transitions labeled with [s,o]. The transition labeled [s] will loop back to state 11, but the transition labeled [o] will have to go to a new state which represents the set {s,o}.

If we make these modifications to the state merging learner, then the machine obtained after observing {tosos, ʃotoʃ, stot} is the one in Figure 4.4. At each time step, this modified learner makes the same generalizations as the Precedence Learner. The language accepted by the machine in Figure 4.4 is the same as the one accepted by the machine shown in Figure 4.1. It is possible to efficiently obtain the smallest forward deterministic machine by an additional (precedence-language specific) merging procedure described in Appendix D–2.4.

5 Properties of LDA

Both Hansson (2001) and Rose and Walker (2004) describe additional properties of LDA patterns. In this section, I evaluate to what extent those properties follow

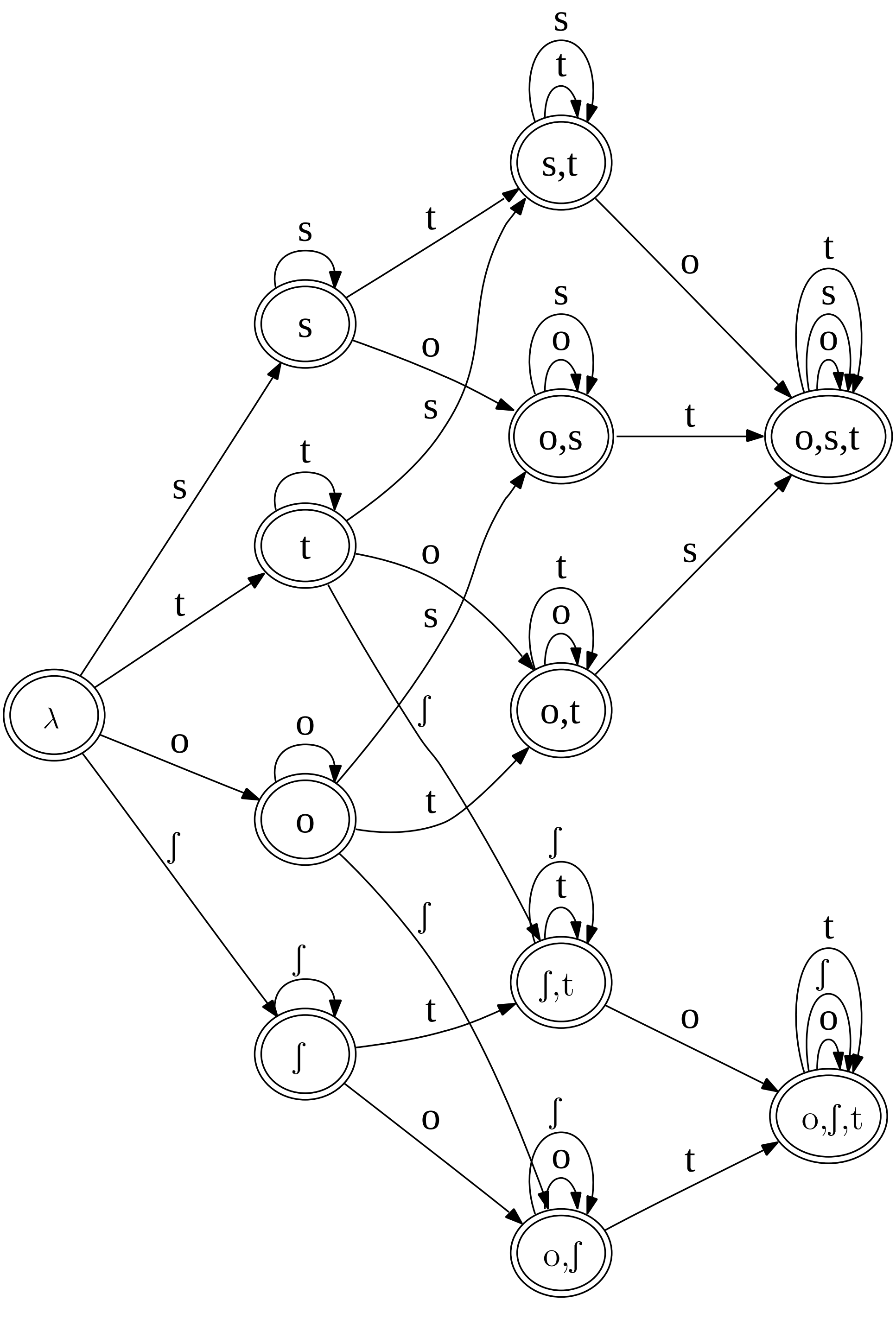

Figure 4.4: FSA Representation of Additional Inferences for Words {tosos, ʃotoʃ, stot}

from the precedence learner.

There are some properties not reflected in precedence learning. First, the model admits phonetically unnatural long distance patterns whereas the agreeing segments in LDA patterns are highly similar. Second, the model is unable to model distance effects observed in some long distance patterns. In §5.1 and §5.2 below, I explain why neither of these is problematic for the result obtained here.

On the other hand, precedence learning does predict one major facet of LDA patterns. Both Hansson (2001) and Rose and Walker (2004) conclude that LDA patterns are notable for the absence of blocking effects. Cases of blocking do exist, but are rarely attested. The precedence learner is unable to model long distance agreement which exhibit blocking effects. I discuss the relevant issues in §5.3.

5.1 Phonetically Unnatural LDA Patterns

It has been observed, in both the domains of consonantal harmony and vowel harmony, that the elements which stand in the agreement relationship are similar segments (Hansson 2001, Rose and Walker 2004, Baković 2000). However, the formalism adopted here admits description of long-distance agreement patterns that are not attested, e.g. not including (b,ʒ) in a grammar G effectively describes the LDA constraint '[b] never precedes [ʒ]'.

There are two responses to this observation. The first is that perhaps the precedence model is correct. Do we really know whether adults, who learned a phonotactic grammar in a linguistic environment containing no words in which [b] precedes [ʒ], find a word with a [b] preceding [ʒ] as less acceptable as one which does not, all other things being equal? An answer to this question can in principle be discovered in the lab.

The other response is to acknowledge that other substantive constraints further restrict the space of possible grammars. In other words, we do not expect the proposed formalization of locality in terms of precedence to explain why LDA patterns hold over similar segments, and instead hold that such restrictions will follow from other considerations, e.g. substantive bias (Wilson 2006a). The properties of precedence languages do not depend in any particular way on the symbol being interpreted as a segment as opposed to some more richly articulated phonological entity. Thus one promising area of future research would be to explore how the addition of features and a notion of similarity to a learner might be combined with precedence grammars such that the resulting hypothesis space more closely resembles the typological observations.

5.2 Distance Effects

Some cases of LDA exhibit distance effects. Some of these effects are 'subword', i.e. the agreement is only obligatory in roots or stems (Hansson 2001). Because the current model has no concept of distance, precedence grammars cannot describe these facts at all. This problem, however, is the problem of identifying the domain a phonotactic constraint or rule applies. This is one of the many complicating factors this dissertation has abstracted away from. How the child, or any computing device, can simultaneously discover a pattern and its domain of application is a research program beyond the scope of this dissertation.

There is also variability within, and exceptionality to, LDA patterns which increases the further apart the agreeing segments are. It is not uncommon, for example, for agreement to be obligatory for adjacent consonants or those in transvocalic contexts, but optional at greater distances (Hansson 2001). It is not clear whether these observations should affect our descriptions of the grammar (though see Rose and Walker (2004), Martin (2004)). Certainly they matter for a theory of learning because we would like to develop learners that succeed (in some sense) in the presence of less-than-perfect input. However this goal, as explained in Chapter 2, is also beyond the scope of this dissertation.

5.3 Blocking Effects

One of the other properties of LDA patterns is the absence of blocking effects (Hansson 2001, Rose and Walker 2004). That is, there do not seem to be any LDA patterns which admit regular exceptions if certain segments intervene between the agree-ers (but see below). This is different from languages with feature-spreading patterns, which often have blocking elements.

One exception to this claim comes from Ineseño Chumash. Ineseño Chumash has a sibilant harmony pattern similar to Navajo. Stridents never precede stridents with the opposite value of anteriority *except* [ʃ] may precede [s] as long as the nearest preceding [ʃ] to the [s] is immediately followed by [n,t,l] (Applegate 1972, Poser 1982). The examples in (8) indicate the aspect of the Ineseño Chumash pattern that resembles Navajo, and (9) the blocking effect.

(8)	a.	k**s**unonu**s**	‘I obey him’
	b.	*k**s**unonu**ʃ**	(hypothetical)
	c.	k**ʃ**unot**ʃ**	‘I am obedient’
	d.	*k**ʃ**unot**s**	(hypothetical)

(9)	a.	**ʃt**ijepu**s**	‘he tells him’
	b.	**ʃ**i**ʃl**u**s**i**s**in	‘they (dual) are gone awry’
	c.	***ʃ**i**ʃk**u**s**i**s**in	(hypothetical)

(10) summarizes the phonotactic generalizations that are drawn from the above data.

(10) 1.[ʃ] is never preceded by [s].

2.[s] is never preceded by [ʃ] unless the nearest preceding [ʃ] to the [s] is immediately followed by [n,t,l].

This constitutes a local blocking pattern because any one of [n,t,l], if immediately contiguous after a [ʃ], blocks further [-anterior] agreement.

The precedence learner is unable to learn any blocking effect like the one in Ineseño Chumash. This is because upon seeing a word like [**ʃt**ijepu**s**] 'he tells him', the learner concludes incorrectly that [ʃ] may precede [s] regardless of intervening material.

It is interesting that the precedence learner fails to learn the LDA pattern present in Ineseño Chumash because such blocking patterns are rare. In fact, Rose and Walker (2004) and Hansson (2001) argue that the absence of blocking is a distinguishing property of LDA patterns. If human learners use an inductive principle like the one the Precedence Learner uses, we have an explanation for the fact that most LDA patterns do not exhibit blocking effects: LDA patterns with blocking patterns are difficulty to learn (this is made more precise below).

How rare is blocking in LDA patterns? There are only three cases discussed in the literature: Ineseño Chumash, Sanskrit Vedic, and Kinyarwanda. Both Hansson (2001) and Rose and Walker (2004) diagnose the Sanskrit Vedic pattern as a feature spreading (see also Schein and Steriade (1986), Gafos (1999), Chiosáin and Padgett (2001)). Evidence presented by Mpiranya and Walker (2005), Byrd et al. (2006) and Walker (2007) likewise suggests that Kinyarwanda exhibits feature-spreading as opposed to LDA. This leaves Ineseño Chumash as the only case of LDA with blocking discussed in the literature.[5]

In the case of Ineseño Chumash, Hansson (2001) points out that the local block-

[5] Another potential case is voicing agreement in Ngizim (Schuh 1978, 1997).

ing is probably related to the fact that there are no surface [ns, ts, ls] clusters. Evidence indicates a rule changes underlying /ns, ts, ls/ to [nʃ, tʃ, lʃ], respectively (Poser 1982). In a full phonological grammar, this rule may interact with rules enforcing agreement resulting in the phonotactic pattern (see also McCarthy (to appear)). On these grounds, Hansson (2001) dismisses the case of Chumash. Still, it remains to be explained how a learner can acquire the LDA pattern from surface forms which exhibit a regular exception.

If LDA patterns admit local blocking like the kind found in Ineseño Chumash, then the hypothesis space given by precedence learners will have to be expanded. In Appendix D–3, I sketch one way of elaborating the hypothesis space of precedence grammars so that a learner can identify LDA patterns with local blocking like the one found in Ineseño Chumash.[6] This elaborated hypothesis space properly contains the hypothesis space learnable by the precedence learner. Consequently, the learner for this larger hypothesis space takes longer to succeed (i.e requires more evidence to converge) because it is able to make more distinctions. This suggests the hypothesis spaces discovered here are on the right track: the simpler precedence learner makes fewer distinctions and is able to learn common kinds of patterns, whereas the elaborated learner, which necessarily takes longer due to the larger hypothesis space, learns more complex, rarer patterns like the one found in Ineseño Chumash.

[6] This learner cannot learn, however, unattested non-local blocking LDA patterns (where the blocking element can appear anywhere between the two agreeing segments which superficially appears to be the case in Sanskrit Vedic and Kinyarwanda).

6 Learning LDA Patterns and Patterns over Contiguous Segments

The learners presented in this chapter are unable to learn patterns over contiguous segments. They can only learn LDA patterns. How can the Precedence Learner be combined with the n-gram learner to learn phonotactic patterns which contains both patterns over contiguous and non-contiguous segments? The answer is simple. Given some input sample, both the n-gram learner and the precedence learner go to work, making their respective generalizations. The result is two regular languages: a n-gram language obtained by the n-gram learner and a precedence language obtained by the precedence learner. Well-formed words in the language are those words which exist in both n-gram and precedence languages. In other words, the phonotactic grammar which respects both kinds of patterns found in the sample is simply the intersection of the precedence and n-gram languages (i.e. the intersection of the finite state machines obtained by the two learners).

7 Summary

This chapter described the kinds of long distance agreement patterns, which are the patterns found over non-contiguous sounds in natural language. Precedence grammars and precedence languages can represent the phonotactic knowledge speakers have of long distance agreement patterns. It was shown how speakers can acquire these representations from limited experience using an inductive principle which straightforwardly relates to properties of LDA patterns. It follows that if people make generalizations like the precedence learner, we explain why LDA patterns exist. Also, because the precedence learner fails to learn LDA patterns with blocking, we explain their absence (or extreme rarity) in the known LDA typology. Finally, it was shown how these grammars and learners can be encoded into finite state terms. Figure 4.5 shows the precedence languages as a small subset of the regular languages which include patterns Navajo sibilant harmony.

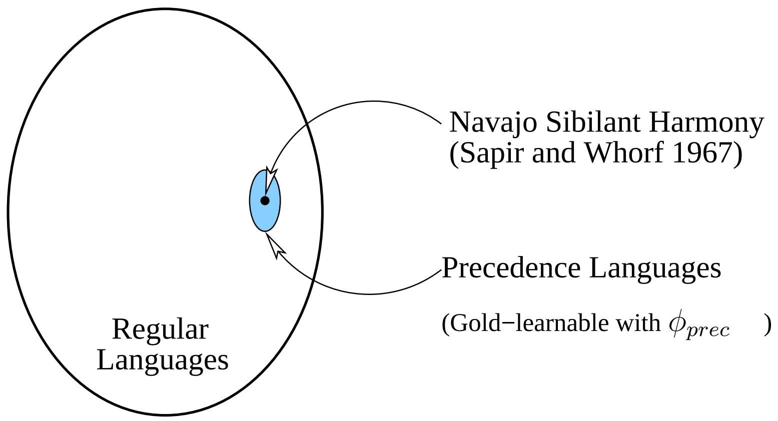

Figure 4.5: Precedence Languages

Appendices

D–1 A Formal Treatment of Precedence

In this section, we formally define precedence relations, sets, grammars and languages. It follows definitionally that the function PS (defined below), which computes the precedence relations in a given string, belongs to $\mathcal{F}$ (defined in Appendix C–1.1). Thus, the function PS, just like the function CS which computes the n-grams of a string, naturally defines a class of string extension grammars and languages. I call these languages precedence languages and denote them with $\mathcal{L}_{prec}$. Because $PS \in \mathcal{F}$, $\mathcal{L}_{prec}$ is closed under intersection and is identifiable in the limit by a learner which records precedence relations in observed words (by way of PS), as shown in Appendix C–1. Precedence languages have additional structure; these properties are also given below.

D–1.1 Precedence Relations and Sets

The symbols in the alphabet are augmented with the word boundary symbol #. We write $V = \Sigma \cup \{\#\}$.

Definition 11 Let $w \in V^*$ and $w = x_1 x_2 \ldots x_n$ for some $n \in \mathbb{N}$. For i, j where $1 \leq i, j \leq n$, we say x_i *precedes* x_j in w iff $i < j$

We write $x_i <_w x_j$. $<_w$ is also called the *precedence relation* induced by w. When w is understood, we just write $<$.

Example 9 Let w = *abcd*. It will be useful to write w with indices, i.e.as $a_1 b_2 c_3 d_4$. Then the following are true:

1. $a_1 < b_2$ (since $1 < 2$).

2. $b_2 < d_4$ (since $2 < 4$).

Remark 1 *If two symbols stand in a precedence relation in a subsequence of a string, then that precedence relation holds for the string itself. I.e. for all $u, v, w \in \Sigma^*$, if $a <_w b$, then it is also the case that $a <_{uwv} b$. This turns out to have interesting consequences, as shown below in Theorem 21 and Theorem 22 below.*

Every string $w \in V^*$ induces a *precedence set*, that is the set of precedence relations induced by w.

Definition 12 Let $w \in \Sigma^*$. The n-precedence set of w is:

$$PS(w) = \{(a, b) : a <_{\#w\#} b\}$$

PS is a function that, like the n-gram function CS, belongs to the class of functions $\mathcal{F}$ described in Appendix C–1.

Example 10 Consider $w = abc$. Then

$$PS(w) = \left\{ \begin{array}{l} (\#, a), (\#, b), (\#, c), \\ (a, b), (a, c), \\ (b, c), \\ (a, \#), (b, \#), (c, \#), \end{array} \right\}$$

Note that there is no w such that $PS(w) = \emptyset$ since even when $|w| \leq 1$, word boundaries are added to either side of w when computing PS.

Note also that computing $PS(w)$ is tractable in the length of w. This is because the most precedence relations in w is $(|w|^2 - |w|)/2$. Furthermore, the size of V provides an upper bound on this value because no precedence set contains more than $|V|^2$ elements.

D–1.2 Precedence Grammars and Precedence Languages

Precedence grammars are defined according to Appendix C–1.1 with the function PS. In other words, a precedence grammar G is a subset of V^2. Likewise, the language of a precedence grammar is defined according Definition 6. In other words, a word w belongs to $L(G)$ only if $PS(w) \subseteq G$.

Example 11 Let $\Sigma = \{a, b, c\}$. Let

$$G = \left\{ \begin{array}{llll} (\#, a), & (\#, b), & (\#, c), & (\#, \#), \\ & (a, b), & (a, c), & (a, \#), \\ (b, a), & (b, b), & & (b, \#), \\ (c, a), & & (c, c), & (c, \#) \end{array} \right\}$$

Since G is a subset $V \times V$, it is a precedence grammar. Note that $L(G)$

1. does not include words with two as because (a, a) is not an allowable precedence relation (i.e not in G).
2. does not include words which have a b following a c since (c, b) is not an allowable precedence relation.
3. does not include words which have a c following a b since (b, c) is not an allowable precedence relation.
4. includes all other words.

Remark 2 *Note that if $(\#, \#)$ is not in a precedence grammar G, then $L(G)$ is empty since for any $w \in \Sigma^*$, deciding whether $w \in L(G)$ means determining whether $PS(\#w\#)$ is a subset of G and $(\#, \#)$ is always an element of $PS(\#w\#)$ for any $w \in \Sigma^*$. It follows that λ is an element of every nonempty precedence language and* *similarly that $(\#, \#)$ is an element of the canonical grammar for any nonempty precedence language.*

D–1.3 Properties of Precedence Languages

The next theorem illustrates that, like $\mathcal{L}_{n-gram}$ (for some n), $\mathcal{L}_{prec}$ is closed under reversal but not closed under complement.

Theorem 20 $\mathcal{L}_{prec}$ is not closed under reversal but not complement.

Proof: (reversal) Consider any $L \in \mathcal{L}_{prec}$ and let G be the canonical grammar for L. We show that $L^r = L(G^r)$. Note that

$$(1)\ PS(w)^r = PS(w^r)$$

Consider any $w^r \in L^r$. First we show $PS(w^r) \subseteq G^r$. Consider any $g \in PS(w^r)$. It follows from (1) that $g^r \in PS(w)$. Since $w \in L$, $PS(w) \subset G$ and hence $g^r \in G$. Therefore by definition, $g \in G^r$. Since g is arbitrary, $PS(w^r) \subseteq G^r$. Since for any $g \in G$, $g^r \in G^r$ by definition, it is the case that $PS(w^r) \subseteq G^r$. Since w^r is aribitrary, $L \subseteq L(G^r)$. Similarly, we can show $L(G^r) \subseteq L$. Since L is arbitrary it follows that $\mathcal{L}_{prec}$ is closed under reversal.

(not complement) The only precedence language which does not contain λ is the empty language (see Remark 2). Any nonempty precedence language other than Σ^* then has a nonempty complement which does not contain λ, and is therefore (by Remark 2) not describable by any precedence grammar. □

Precedence languages have some unusual properties, to which the following theorems speak. First, we show that the precedence set of a subsequence of a string w is a subset of the precedence set of w.

Theorem 21 Let $L \in \mathcal{L}_{prec}$. Then for all $w, u, v \in \Sigma^*$ such that $uwv \in L$, $PS(w) \subseteq PS(uwv)$.

Proof: Consider any $uwv \in L$ and any $(a, b) \in PS(w)$. Thus $a <_w b$ by definition and by Remark 1, $a <_{uwv} b$. Consequently, $(a, b) \in PS(w)$ and the theorem is proved. □

Corollary 13 Let $L \in \mathcal{L}_{prec}$. Then $Pr(L) = Sf(L) = L$.

Proof: Let $uwv \in L \in \mathcal{L}_{prec}$ and G generate L. By considering the case when $u = \lambda$, Theorem 21 tells us that $PS(\#w\#) \subseteq PS(\#wv\#)$, which we know to be a subset of G (since $wv \in L$). Thus every prefix of the language belongs to the language. Similarly when we consider the case when $v = \lambda$, we conclude every suffix of L belongs to L. □

Corollary 14 Let $L \in \mathcal{L}_{prec}$. For all $u, v \in \Sigma^*$, if $uv \in L$ then $u, v \in L$.

Proof: Follows directly from Corollary 13. □

Incidentally, this is another way to see that λ belongs to every nonempty precedence language since λ is a prefix (and suffix) of every string.

It is useful to recall the concept of the *range* of a string (see Appendix A–1.5) which are the set of symbols present in the string. I denote the range of a string w with $range(w)$. Also recall that γ_{PS} is extends the domain of PS from strings to subsets of Σ^*.

Lemma 18 Let $L \in \mathcal{L}_{prec}$ and $u, v \in L$. Then $uv \in L$ iff for all $b \in range(v)$ and $a \in range(u)$, $(a, b) \in \gamma(L)$.

Proof: ($\Rightarrow$) Suppose $uv \in L$. Thus $PS(\#uv\#) \subseteq \gamma(L)$. Consider any $b \in range(v)$ and any $a \in range(u)$. Clearly $a <_{\#uv\#} b$ and therefore $(a, b) \in PS(\#uv\#) \subseteq \gamma(L)$.

($\Leftarrow$) Suppose for all $b \in range(v)$ and $a \in range(a)$, $(a, b) \in G$. Since $u, v \in L$, $PS(\#u\#) \subseteq \gamma(L)$ and $PS(\#v\#) \subseteq \gamma(L)$. Consequently, along with the ($\Leftarrow$) assumption, $PS(\#uv\#) \subseteq \gamma(L)$ and $uv \in L$. □

Lemma 19 Let $L \in \mathcal{L}_{prec}$ and suppose $uv \in L$. Then for all $b \in range(v)$, $ub \in L$.

Proof: Consider any $b \in range(v)$ and any $a \in range(u)$. Since $uv \in L$, $(a, b) \in \gamma(L)$. Therefore $PS(\#ub\#) \subseteq \gamma(L)$ and $ub \in L$. □

We can now prove the following theorem which makes clear the structure of languages in $\mathcal{L}_{prec}$.

Theorem 22 Let $L \in \mathcal{L}_{prec}$. Then for all $uv \in Pr(L)$, $T_L(uv) = T_L(u) \cap T_L(v)$.

Proof: Consider any $uv \in Pr(L)$ and any $w \in T_L(uv)$. Thus $uvw \in Pr(L) = L$ and by Corollary 13, $vw \in L$. Thus $w \in T_L(v)$. To see that $w \in T_L(u)$, note that since $uvw \in L$, by Lemma 19, $uw \in L$. Thus, $w \in T_L(u)$. Therefore, $w \in T_L(u) \cap T_L(v)$ and since w was arbitrary, $T_L(uv) \subseteq T_L(u) \cap T_L(v)$.

G be a precedence grammar generating L and consider any $w \in T_L(u) \cap T_L(v)$. Thus $uw \in L$ and $vw \in L$. Thus, for all $a \in range(u)$, $b \in range(v)$, and $c \in range(w)$, both (a, c) and (b, c) belong to G. Since $uv \in L$, (a, b) is also in G. Thus by Lemma 18, $uvw \in L$ and hence $w \in T_L(uv)$. Therefore $T_L(u) \cap T_L(v) \subseteq T_L(uv)$ and we conclude $T_L(uv) = T_L(u) \cap T_L(v)$. □

D–1.4 Learning Precedence Languages by String Extension

Finally, the results in Appendix C–1.2 tell us that the following learner identifies $\mathcal{L}_{prec}$ in the limit.

(11)

$$\phi(t_i) = \begin{cases} \emptyset & \text{if } i = 0 \\ \phi(t_{i-1}) & \text{if } t_i = \epsilon \\ \phi(t_{i-1}) \cup PS(t_i) & \text{otherwise} \end{cases}$$

It is worthwhile to reiterate the important results in Appendix C–1 that lead to this result. Because because $PS \in \mathcal{F}$, for every individual word w the learner observes, $PS(w)$ instantiates aspects of G. This is what is meant by string extension. It follows that at each point in the text, the learner ϕ guesses the smallest precedence language which contains every observation it has seen so far. Since every precedence language has a finite characteristic sample, there is some point in the text where the learner converges to the correct precedence language.

Also note that since PS is an efficient function in the length of its input string, the learning function ϕ above is also efficient in the size of any given sample.

D–2 Learning $\mathcal{L}_{prec}$ via State Merging

D–2.1 Finite State Representation

Theorem 23 For any $L \in \mathcal{L}_{prec}$, there is a finite state acceptor which accepts exactly L.

Proof: Let L be a precedence language. Then there is an acceptor A such that $L(A) = L$ where A is defined as follows:

$$\begin{aligned} Q &= \{range(u) : u \in Pr(L)\} \\ I &= \{\emptyset\} \text{ iff } L(G) \neq \emptyset \text{ , otherwise } \emptyset \\ F &= \{range(u) : u \in L\} \\ \delta(S, a) &= S \cup \{a\} \text{ iff } S, S \cup \{a\} \in Pr(L) \end{aligned}$$

If L is empty then $L(A)$ is empty so assume L is not empty. Consider any $w \in L$. Since $w \in L$, $range(w)$ is a final state. By definition of Q, every prefix of w has a state and it follows from the definition that $\delta(I, w) = range(w)$. Thus, $w \in L(A)$ and so $L \subseteq L(A)$. Similarly, if $w \in L(A)$ then $\delta(I, w) = range(w)$, which also must be a final state. Thus $w \in L$, i.e. $L(A) \subseteq L$, which now implies $L = L(A)$. □

Note also that, as a consequence of Corollary 13, every state in the above construction is a final state.

Figure 4.6 illustrates the construction used in Theorem 23 with the grammar in Example 11.

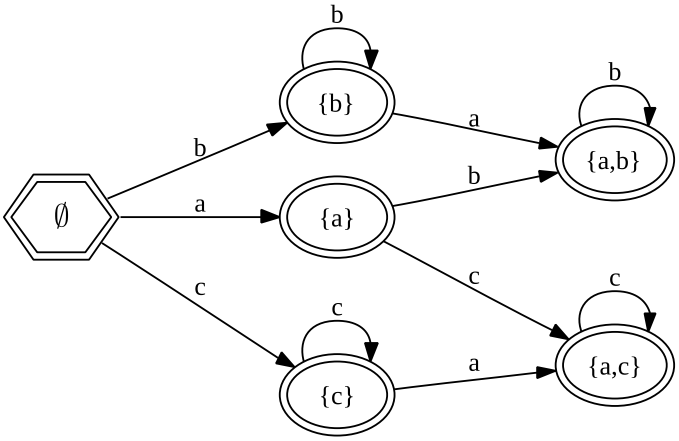

Figure 4.6: FSA for the Grammar in Example 11

Note that the machines have more structure than what is revealed in the construction. In particular, the following lemma holds as a direct consequence of

Theorem 22.

Lemma 20 For any $L \in \mathcal{L}_{prec}$, let $A = (Q, I, F, \delta)$ be the acceptor which recognize L, constructed according to Theorem 23. Then for all $S \in Q$, $a \in \Sigma$, $\delta(S, a)$ is defined iff for all $s \in S$, $\delta(\{s\}, a)$ is defined.

Proof: This follows directly from Theorem 22 and the construction above. □

As an example, notice that the labels on the outgoing transitions of state $\{a, b\}$ is the intersection of the outgoing transitions of states $\{a\}$ and $\{b\}$.

Also, note that in the construction given above A is forward deterministic. As a consequence of this definition, it should be clear that most states a machine will have is $2^{|\Sigma|}$. Note that Σ^* (a one state canonical acceptor) uses all of these states! Although standard efficient minimization techniques (e.g. Hopcroft et. al. (2001)) allow one to obtain the smallest determinstic acceptor (a canonical acceptor for $L(G)$), we show below another (simpler) way to obtain the canonical acceptor of a precedence grammar. But first we show the construction above makes apparant the following language-theoretic characerization of precedence languages.

D–2.2 Towards a Language-theoretic Characterization

Theorem 24 Let $L \in \mathcal{L}_{prec}$, and consider $u, v \in Pr(L)$. If $range(u) = range(v)$ then $T_L(u) = T_L(v)$.

Proof: Suppose $range(u) = range(v)$. Then by construction of forward deterministic $A(L)$ above, $\delta(I, u) = \delta(I, v)$. Thus $T_L(u) = T_L(v)$. □

D–2.3 Learning Precedence Languages by State Merging

Here we present a description of the learner ϕ in Appendix D–2.1 in terms of state-merging. Given some acceptor $A = (Q, I, F, \delta)$, consider the function which maps states q in Q to sets of symbols which make up strings by which state q can be reached from I; i.e. the function $R : Q \rightarrow 2^{\Sigma}$ defined below.

$$R(q) = \{a \in \Sigma : \exists w \in \Sigma^* \text{ such that } \delta(I, w) = q \text{ and } a \in range(w)\} \tag{4.1}$$

$R(q)$ can be thought of as the union of the range of strings which form the incoming paths to q. As mentioned in Appendix A–1.3, this function induces an equivalence relation over the states Q (i.e. $p \sim q$ iff $R(p) = R(q)$). This relation, denoted $\sim_R$ is called the *range* equivalence relation. The range equivalence relation induces a partition π_R over Q. The blocks of this partition are merged to yield a new acceptor.

It is now possible to state the learner precisely. We give the learner in two versions, a batch learner and an iterative learner.

Algorithm 3 The Precedence State Merging Learner (batch version)

```
  Input: a positive sample $S$ and a positive integer $n$.
  Ouput: an acceptor $A$.
  Initialization
  Let $A_0 = (Q_0, I_0, F_0, \delta_0) = PT(S)$.
  Merging
  Compute $\pi_R$ over $Q_0$.
  Termination
  Let $A = A_0/\pi_R$ and output acceptor $A$.
```

Algorithm 4 is the iterative version of this algorithm.

Algorithm 4 The Precedence State Merging Learner (iterative version)

```
**Input:** a positive sample $S$ and a positive integer $n$.
**Ouput:** an acceptor $A$.
*Initialization*
Let $A_0 = (\{q_0\}, \{q_0\}, \emptyset, \emptyset)$.
Let $i = 1$.
**for all** $w \in S$ **do**
    Let $A'_{i-1}$ be the extension of $A_{i-1}$ with $w$.
    Compute $\pi_R$ over $Q'_{i-1}$.
    Let $A_i = A'_{i-1}/\pi_R$.
    Increase $i$ by 1.
**end for**
*Termination*
Output acceptor $A_i$.
```

Algrorithms 3 and 4 are guaranteed to converge to grammars which recognize the target language, provided S is sufficient. However, as noted in §4, these learners do not converge as quickly they could. In particular, they do not take advantage of the properties of precedence languages given in Corollary 13 and Theorem 22. Thus it is not the case that $PT(S)/\pi_R$ is the same as the acceptor obtained by the construction in Theorem 23 for $L(\gamma(S))$. However, the machine obtained by adding making every state final and making adding transitions and states in accordance with Lemma 20 is the same acceptor, but I omit the statement of the result and formal proof here.

D–2.4 Obtaining the Canonical FSA for a Precedence Grammar

The finite state represnetation for a given precedence grammar was forward deterministic, but not canonical. This section compiles a few notes about how to obtain the smallest forward determinstic acceptor which accepts the same language as a precedence grammar.

Lemma 21 Consider $L \in \mathcal{L}_{prec}$. $\forall u, v \in Pr(L)$, $T_L(u) = T_L(v)$ iff $range(u) = range(v)$ or $Pr^1(T_L(u)) = Pr^1(T_L(v))$.

Proof: Consider $L \in \mathcal{L}_{prec}$ and let $G = \gamma(L)$. The ($\Rightarrow$) direction follows trivially so consider any $u, v \in Pr(L)$. If $range(u) = range(v)$ then by Theorem 24, $T_L(u) = T_L(v)$. Now suppose $range(u) = range(v)$ but $Pr^1(T_L(u)) = Pr^1(T_L(v))$. Consider any $x \in T_L(u)$. For all $x_0 \in range(x)$, $u_0 \in range(u)$, $(u_0, x_0) \in G$. Therefore $ux_0 \in L$ (by Corollary 13) and by assumption $vx_0 \in Pr(L)$ too. Consequently for all $v_0 \in range(v)$, $(v_0, x_0) \in G$. Since x_0 is arbitrary in x, it follows that $vxinL$ by Lemma 18. Thus $x \in T_L(v)$ and since x is arbitrary, $T_L(u) \subseteq T_L(v)$. The same argumentation shows that $T_L(v) \subseteq T_L(u)$ and therefore $T_L(u) = T_L(v)$. Thus the lemma is proved. □

The idea is that the finite state representation of a precedence grammar can be made canonical by merging the blocks of this partition π_{O_1} (note that bigram grammars have the same property). As was the case with n-gram grammars the proof relies partly on Lemma 17.

Theorem 25 Let $L \in \mathcal{L}_{prec}$ and A be the acceptor for L, constructed according to Theorem 23. Then the tail canonical acceptor for L, denoted $A_T(L)$, is isomorphic to A/π_{O_1}.

Proof: Omitted. □

D–3 Extending Precedence Grammars to Handle Local Blocking

This section defines *relativized bigram precedence grammars*, which begin to generalize the notion of precedence grammars. It is shown that this type of grammar can describe the LDA pattern with local blocking (as found in Ineseño Chumash) as well as the more common patterns with no blocking. It is also shown that the languages recognizable by these grammars are identifiable in the limit.

This is accomplished with another function belonging to $\mathcal{F}$. This function extracts what I call the *relativized bigram precedence relations* in a given string. This function which naturally defines a class of languages which is identifiable in the limit by recording such relations.

D–3.1 Definitions and Examples

Definition 13 Let $w \in V^*$ and $w = x_1x_2 \ldots x_n$ for some $n \in \mathbb{N}$. For i, j where $1 \leq i, j \leq n$, we say x_ix_{i+1} *relatively bigram precedes* x_j in w iff

1. $i + 1 < j$ and
2. for all k such that $i + 1 < k < j$, $x_i \neq x_k$.

We write $x_ix_{i+1} <_w x_j$. $<_w$ is also called the *relativized bigram precedence relation* induced by w. When w is understood, we just write $<$.

It should be clear from the definition where the notions of *bigram* and *relative precedence* appear. The *bigram* refers to the fact that the relation defines precedence between a contiguous subsequence of length two and another segment. The second condition in the definition provides the notion of *relativized* precedence.

Example 12 Let w $=$ *babc*. We can write w as $b_1a_2b_3c_4$. Although $b_1a_2 < b_3$, $b_1a_2 \not< c_4$ since although $2 < 4$, there is a b (b_3) which intervenes between b_1 and c_4; i.e. condition (2) of Definition 13 is not met.

Every word $w \in V^*$ induces a *relativized bigram precedence set*, that is the set of relativized bigram precedence relations induced by w.

Definition 14 Let $w \in V^*$. The relativized bigram precedence set of w is:

$$PS_{(2,1)}(w) = \{(ab, c) : ab <_{\#\#w\#} c\}$$

I adopt the notation $PS_{(2,1)}$ because I suspect there is a family of precedence sets that can be defined in this way (cf. the family of n-gram relations).[7]

Example 13 Consider $w = babcd$. Then

$$PS_{(2,1)}(w) = \left\{ \begin{array}{lllll} (\#\#, b), & (\#\#, a), & (\#\#, c), & (\#\#, d), & (\#\#, \#), \\ & (\#b, a), & (\#b, c), & (\#b, d), & (\#b, \#), \\ (ba, b), & & & & \\ & & (ab, c), & (ab, d), & (ab, \#), \\ & & & (bc, d) & (bc, \#) \\ & & & & (cd, \#) \end{array} \right\}$$

Note that writing w as $b_1a_2b_3c_4d_5$ it is clear that $(ba, c) \notin PS_{(2,1)}(w)$ since by definition $b_1a_2 \not< c_4$ since b_3 intervenes. Similarly for (ba, d).

As was the case with precedence languages, we define a *bigram-precedence grammar G* to be a subset of V^3, and the *bigram precedence languages* are defined according to the whether PS(w) is a subset of the grammar (see Appendix C–1). We denote this class of languages $\mathcal{L}_{(2,1)prec}$.

[7] Intuitively, the function PS defined in Appendix D–1 would be called $PS_{(1,1)}$ in this family.

The LDA pattern of Ineseño Chumash, with the local blocking, belongs to $\mathcal{L}_{(2,1)prec}$. This can be seen by considering a grammar which includes (ʃt,s) but excludes (ʃx,s) where [t] stands for all coronal segments and [x] stands for all non-cornal segments.

D–3.2 Properties of Relativized Bigram Precedence Languages

There are a number of properties of this language class which are like those of $\mathcal{L}_{prec}$ that I omit here for the sake of brevity. It is also easy to show that the bigram precedence languages properly contain the precedence languages, though I omit the proof here.

Theorem 26 $\mathcal{L}_{prec} \subset \mathcal{L}_{(2,1)prec}$.

Proof: Omitted. □

D–3.3 Learning with String Extension

(12)

$$\phi(t_i) = \begin{cases} \emptyset & \text{if } i = 0 \\ \phi(t_{i-1}) & \text{if } t_i = \epsilon \\ \phi(t_{i-1}) \cup PS_{(2,1)}(t_i) & \text{otherwise} \end{cases}$$

The learner is efficient in the length of the input. It should be noted that this learner is 'slower' to learn a precedence language (without local blocking) than the precedence learner in the sense that it needs to see a larger sample in order to succeed. This follows from the fact that it makes (strictly) more distinctions than the precedence learner.

CHAPTER 5

Stress Patterns

> Metrical theory forms part of a general research program to define the ways in which phonological rules may apply non-locally by characterizing such rules as local with respect to a particular representation.(Hayes 1995:34)

1 Overview

In this chapter, I consider the rhythmic patterns found in the world's languages. I introduce a (near) universal property of these patterns called *neighborhood-distinctness*. This property, which relates directly to phonologist's notions of locality, naturally provides an inductive principle which the learner below uses to successfully identify the target stress pattern from finite samples.

The choice to study stress systems was made for three reasons. First, they are a well-studied part of phonological theory and the attested typology is well-established (Hyman 1977, Hayes 1981, Prince 1983, Halle and Vergnaud 1987, Idsardi 1992, Bailey 1995, Hayes 1995, Hyde 2002, Gordon 2002). Secondly, learning of stress systems has been approached before making it possible to compare learners and results (Dresher and Kaye 1990, Gupta and Touretzky 1991, Tesar 1998, Tesar and Smolensky 2000).

The rest of this chapter is organized as follows. §2 describes the stress typology used in this study and explains why n-gram and precedence learning models are inadequate to describe these patterns. §3 introduces the property neighborhood-distinctness, which is shown to be a (near) universal property of these attested patterns. In §4, I present a simple learner, which I call the Forward Neighborhood Learner, which learns many of the neighborhood-distinct stress patterns but not all of them. Then I motivate an elaboration of this learner called the Forward Backward Neighborhood Learner, which succeeds on 100 of the 109 stress patterns (414 out of 422 languages) in the typology. In §5, I show another way in which the learner approximates the stress patterns found in the world's languages: Many logically possible but unattested stress systems cannot be learned by the Forward Backward Neighborhood Learner. In other words, the range of the learning function approximates the stress patterns in the world's languages in an interesting way.

2 Stress Patterns in the World's Languages

2.1 The Stress Typology

The collection of languages and stress patterns discussed here are primarily due to two typological surveys: Bailey's (1995) database of primary stress patterns and Gordon's (2002) typology of quantity-insensitive stress patterns.[1] Both of these researchers culled languages from first source material and report dominate stress patterns, though it should be recognized that many languages exhibit subordinate stress patterns (not part of the typology here). The dominant stress patterns are also discussed in standard texts of metrical theory such as Halle and Vergnaud (1987) and Hayes (1995). Note that 'language' here is domain-specific: a language

[1] The stress database *StressTyp* currently maintained by Harry van der Hulst and Rob Goedemans did not become available online until after this project was underway. Many of the same languages in Bailey (1995) and Gordon (2002) are included in StressTyp, but StressTyp includes more languages than the ones in the Bailey (1995) and Gordon (2002) combined. Their database is available here `http://stresstyp.leidenuniv.nl/`. Also see Goedemans et al. (1996).

which assigns stress differently in nouns and verbs (e.g. Lenakel) is counted as two languages. Because Bailey (1995) only catalogs primary stress patterns, I consulted every primary source possible given in Bailey (1995) and recorded a description of the dominant secondary stress pattern (if any) with the aid of two undergraduate assistants Rachel Schwartz and Stephen Tran. The resulting typology is summarized as an appendix to the dissertation. References for the languages that are mentioned in this chapter can be found in this appendix.

The resulting stress typology used in this chapter is available electronically as a free open-source MYSQL database. In addition to the stress information, the database includes the Ethnologue Language Code for the languages in the typology, as well as source information. There is a website that accesses the database; this site is linked from the author's website.

One reason for making such a database available is that it provides other researchers the opportunity to find and correct errors unwittingly made by researchers whose work contributes to the typology in some manner, including of course my own. Although I have done my best to faithfully transfer the information in Bailey (1995) and Gordon (2002) and have checked the accuracy repeatedly, I think it is inevitable that some errors occur along the way. Understanding the secondary stress descriptions for languages in Bailey (1995) also requires interpretations and the usual warnings apply (see Hayes (1995:1)). Finally, it is also possible that the sources upon which these typologies are based contain mistaken analyses of the stress patterns or unintentionally omitted relevant information. Despite all of this, I maintain that the database is one accurate representation (cf. StressTyp in footnote 1) of the field's state of knowledge regarding known stress patterns in human languages.

There are many ways one can categorize the kinds of stress patterns found in the world's languages. Following previous researchers, two useful distinctions that are employed here are (1) whether or not stating the primary stress rule requires reference to the type of syllable (often in terms of its quantity, also called weight) and (2) whether or not the distance between a primary stressed syllable and the word edge is bounded or not. It turns out the bounded-unbounded distinction only matters for systems which are quantity-sensitive. Thus, there are three categories: quantity-insensitive, quantity-sensitive bounded, and quantity-sensitive unbounded. I introduce these types (and their subtypes) in turn in order to demonstrate the extent of the variation that occurs. After reviewing the variation in the known typology, I introduce a previously unknown (near) universal property of stress patterns: neighborhood-distinctness.

2.1.1 Phonotactic Restrictions

In addition to the different stress assignment rules described below, there is additional variation that is relevant to a learner of stress patterns. Some languages place additional restrictions on what strings of syllables are well-formed. Many languages prohibit monosyllabic words, or words consisting of a single light syllable. Some languages only have superheavy syllables occurring finally. Other languages require every word to have at least (or at most) one heavy syllable. These phonotactic constraints matter for a learner because words which violate these phonotactic constraints are not present in the learner's linguistic environment. Therefore, whenever such a restriction was mentioned in a source, it was noted. These restrictions are included in the typology and contribute to the total number of distinct patterns. For example, Alawa and Mohawk both assign stress to the penultimate syllable, but words in Mohawk are minimally disyllabic, which is not the case as far as I know in Alawa, and so these patterns differ minimally in this respect.

2.1.2 Quantity-Insensitive Stress Systems

Quantity-insensitive (QI) stress patterns are those in which stating the stress rule need not refer to the quantity, or weight, of the syllables. A review of the typology reveals 319 languages to be quantity-insensitive. These 319 languages exhibit 39 distinct stress patterns. These patterns can be divided into four kinds: single, dual, binary and ternary systems. Single stress systems have a single stressed syllable in each word. Dual stress systems have at most two stressed syllables in each word. Binary and ternary systems have no fixed upper bound on the number of stressed syllables in a word and place stress rhythmically on every second or third syllable, respectively. No other kind of QI stress system is attested.

Of the 39 single QI systems, some languages place a single stress initially (e.g. Chitimacha), finally (e.g. Atayl), on the penultimate syllable (e.g. Nauatl), on the peninitial syllable (e.g. Lakota), and on the antepenultimate syllable (e.g. Macedonian). Note that there is no language which place a single stress on the fourth or fifth syllable from the word edge. Nor is there any language which places stress on the middle syllable (or on the left or rightmost middle syllable in words with an even number of syllables). The attested QI single stress patterns are bounded in the sense that primary stress occurs within some bound of the word edge. Because there are some languages like Macedonian which can assign antepenultimate stress, this bound appears to be three.

There are fifteen dual stress patterns in the typology. The most common dual system places primary stress on the penultimate syllable and secondary stress on the initial syllable (e.g. Sanuma). The pattern where primary stress is placed on the initial syllable and secondary stress is placed on the penultimate syllable also exists (e.g. Lower Sorbian). Languages like Quebec French place primary stress on the final syllable and secondary stress initially. Another dual pattern places primary stress on the initial syllable and secondary stress on the antepenultimate syllable (e.g. Mingrelian). Finally, one language places primary on the antepenultimate syllable and secondary on the initial syllable (Georgian).

In the eighteen binary systems, alternating syllables are stressed. One kind of binary pattern places stress on even syllables counting from the right word edge (e.g. Malakmalak), and another places stress on odd syllables counting from the left word edge (e.g. Maranungku). Another places stress on odd syllables counting from the right word edge (e.g. Urubú Kaapor). Other languages place stress on even syllables counting from the left word (e.g. Araucanian).

Some binary patterns allow the alternating pattern to skip a beat (a lapse) or to pick up an extra beat (a clash) near word edges. For example, one pattern places stress on odd syllables counting from the left word edge but never on the final syllable (e.g. Pintupi), resulting in a lapse word-finally in words with an odd number of syllables. Garawa stresses the even syllables counting from the right and the first syllable, but never the second syllable, resulting in a lapse after the first syllable in words with an odd number of syllables. Piro stresses odd syllables from the left, the penultimate syllable, never the antepenultimate, resulting in a lapse before the penult in words with an odd number of syllables. On the other hand, Gosiute (Shoshone) stresses odd syllables counting from the left and the final syllable, resulting in a clash word-finally in words with an even number of syllables. Similarly, Tauya stresses even syllables from the right and the initial syllable, resulting in a clash in words with an even number of syllables. Southern Paiute stresses even syllables from the left and penult, but never the final syllable, which results in a clash on the penult and antepenult in words with an even number of syllables.

There are only two QI languages with ternary patterns in the typology. Cayuvava places syllable on every third syllable, counting from the right. Words with fewer than three syllables place stress initially. Ioway-Oto places stress on the peninitial syllable and every third syllable afterwards.

2.1.3 Quantity-Sensitive Bounded Stress Systems

Quantity-sensitive (QS) stress systems are unlike QI stress systems in that stress placement is predictable only if reference is made to syllable types. Because syllable distinctions are usually describable in terms of the quantity, or weight, of a syllable, these such patterns are called quantity-sensitive. The typology includes44 patterns which have quantity-sensitive bounded patterns.

Consider the classic case of Latin (Jacobs 1989, Mester 1992, Hayes 1995). Latin distinguishes between syllables which are 'light' and those which are 'heavy'. Light syllables are those with a short vowel and no coda consonant, i.e. (C)V. Heavy syllables are all other syllable types in Latin, e.g. CVː, CVC, CVCC. The stress rule is now given below:

(1) In words at least three syllables in length, stress the penult if it is heavy, otherwise stress the antepenult. In shorter words, stress the initial syllable.

The distinction between light and heavy syllables is necessary in order to understand that stress placement is rule-governed. The examples in (2) exemplify the stress rule (Jacobs 1989, Mester 1992, Hayes 1995).

(2)	a.	amíːkus	L H́ H	‘friend, kind’
	b.	gubernáːbunt	L H H́ H	‘they will reign’
	c.	inimiːkìtia	L L H Ĺ L L	‘hostility’
	d.	doméstikus	L H́ L H	‘belonging to the house’
	e.	mándaː	H́ H	‘entrust (2sg.imp)’
	f.	kánis	Ĺ H	‘dog’
	g.	héri	Ĺ L	‘yesterday’

Also note that the examples (e-f) show that stress falls initially in disyllabic words.

Stress patterns like the one found in Latin are not only quantity-sensitive, they are bounded systems; i.e. primary stress falls within a certain distance of the word edge (three syllables from the right edge).

Like the QI patterns, QS patterns can be subdivided in single, dual, binary and ternary types. There is also a type I term ‘multiple’ for reasons given below. Because of the weight distinction, each of these subtypes shows extensive variation.

For example, some single systems are relatively simple. Maidu stresses the first syllable of a word if it is heavy, and the second syllable otherwise. On the other hand, Shoshone Tumpisa stresses the second syllable if it is heavy, but the first syllable otherwise. Kawaiisu is like the Maidu pattern, but at the right edge, whereas Javanese is like the Shoshone Tumpisa pattern, but at the right edge. A more complicated QS single pattern is exemplified by Pirahã. Pirahã makes a distinction between five syllable types, which, for expositional purposes, can be placed on a single scale from ‘lightest’ to ‘heaviest’.[2] Stress falls on either the final, penult, or antepenult: whichever is the rightmost heaviest syllable receives the stress.

[2] These distinctions are usually analyzed along a dimension distinct from weight called *prominence* (Everett 1988, Hayes 1995).

There is one QS dual system in the typology, Maithili, which assigns primary stress to the penult if it is heavy, otherwise to the final if it is heavy, otherwise to the antepenult if it is heavy. If none of those syllables are heavy, stress falls on the penult. Secondary stress falls on the initial syllable.

The QS bounded binary patterns exhibit dramatic variation. For example, Inga assigns secondary stresses iteratively from the right word edge. If the first syllable to the left of the word edge or a stressed syllable is heavy it receives stress, otherwise the second syllable to the left receives stress. Nyawaygi assigns secondary stress like Inga with one difference: Inga places primary stress on the rightmost stressed syllable, Nyawaygi places primary stress on the leftmost one. Manam assigns primary stress to the the rightmost superheavy in the final, penult, or antepenult syllable if there is one, otherwise to the rightmost heavy in the penult or antepenult if there is one, otherwise to the antepenult. Then using the main stress as the starting point, it assigns secondary stresses iteratively like Inga. Seminole Creek assigns stress from the left word edge iteratively: if the first syllable to the right of the word edge or a stressed syllable is heavy it receives stress, otherwise the second syllable to the right receives stress. There are many more binary QS bounded patterns, this is just a sample.

There are fewer ternary patterns. Sentani places final primary stress on final syllable if it heavy otherwise on the penult. Secondary stresses are assigned iteratively to the left of main stress: if the second syllable to the left of a stress is heavy, it receives stress, otherwise the third syllable to the left of a stress receives stress. Estonian optionally assigns secondary stress according to a ternary rule as in Sentani, or according to a binary rule like the one in Palestinian Arabic.

Some languages assign primary stress like the single systems described above, and place secondary stress only on heavy syllables, e.g. Cambodian. These patterns I call ‘multiple’ QS patterns. This because they are similar to binary and ternary patterns, in that there is no upper limit on how many syllables in a word can receive stress. These differ from binary and ternary patterns however, because they are not bounded, as any number of unstressed syllables can occur between stresses. They are included here with QS bounded systems because the location of primary stress is bounded.

Finally, note that the languages make the heavy/light distinction in different ways. For example in Munsee, CVC and CVV syllables count as heavy and CV syllables count as light, but in Malecite-Passamoquody only CVV syllables count as heavy and CVC and CV syllables count as light. Interested readers should consult Gordon (2006) to learn more about the role of syllables weight in prosodic systems. This study abstracts away from how a language makes the distinction between heavy and light syllables for the purpose of assigning stress.

2.1.4 Quantity-Sensitive Unbounded Stress Systems

Quantity-sensitive unbounded stress systems place no limits on the distances between primary stress and word edges. In this way they differ from other QS systems which require stress to fall within a certain window of the word edge or of another stress.

For example, Selkup assigns primary stress according to the 'Rightmost Heavy Otherwise Leftmost' stress rule, stated below in (3).

(3) Place stress on the rightmost heavy syllable in the word. If there are no heavy syllables, stress the leftmost syllable.

In Selkup syllables with long vowels count as heavy, otherwise they count as light. The Selkup words in (4) exemplify the stress rule. Stressed syllables can occur in any position—the first, last or a middle syllable—as long as the rule in (3) is obeyed

(Kuznecova et al. 1980, Idsardi 1992, Halle and Clements 1983, Walker 2000).

(4)	a.	[pynakɨsə́ː]	L L L H́	‘giant!’
	b.	[ilɨsɔ́ːmɨt]	L L H́ L	‘we lived’
	c.	[qóːkɨtɨlʲ]	H́ L L	‘deaf’
	d.	[qumoːqlɪlíː]	L H L H́	‘your two friends’
	e.	[uːcɔ́ːmɨt]	H H́ L	‘we work’
	f.	[uːcɨkkóːqɪ]	H L H́ L	‘they two are working’
	g.	[qúmmɨn]	Ĺ L	‘human being’ (gen.)
	h.	[ámɨrna]	Ĺ L L	‘eats’
	i.	[qólʲcɨmpatɨ]	Ĺ L L L	‘found’

The typological survey includes 45 which assign stress in an unbounded manner like the ones above. There are 26 distinct patterns found among these 45 languages. Two commons sources of variation is whether the left or rightmost heavy is stressed and which syllable is stressed in words without heavy syllables. For example, Murik stresses the leftmost heavy syllable and in words with no heavy syllables, the initial syllable is stressed. Kwakwala is like Murik except stress falls finally in words with no heavy syllables. Cheremis Mountain stresses the rightmost nonfinal heavy syllable, and in words with no heavy syllables, the final syllable is stressed. Other sources of variation include whether there is secondary stress (e.g. alternating from primary stress in ʏidiɲ, heavy syllables in Nubian Dongolese), whether there are any restrictions on the distribution of syllables (e.g. Murik has at most one heavy syllable per word), and whether whether there is a three-way distinction among syllable types (light, heavy, and superheavy as in Klamath).

2.2 Summary of the Typology

Combining Bailey's (1995) and Gordon's (2002) stress typologies yields a typology of 422 languages, exhibiting 109 distinct stress patterns. These patterns are broadly categorized into three groups by primary stress placement: quantity-insensitive, quantity-sensitive bounded, and quantity-sensitive unbounded. Within each of these types there is extensive variation.

2.3 Inadequacy of N-gram and Precedence Learners

N-gram based learners and precedence based learners are inadequate to learn the kinds of patterns here. This is because the hypothesis space to which these learners are intrinsically bound cannot adequately describe the stress patterns above.

Consider first n-gram languages. As was the case with the precedence languages, there is no value n available such that an n-gram language can describe all stress patterns. This is due to the nature of QS unbounded stress systems. For example, in the 'Leftmost Heavy otherwise Leftmost' pattern of Murik, a heavy syllable cannot follow at any distance a stressed light syllable. As demonstrated with long distance agreement, this kind of constraint is simply outside the class of patterns describable with n-gram languages.

Interestingly, precedence-based learners do not fare much better when it comes to unbounded stress patterns. This is because precedence languages have the peculiar property (see Appendix D–1.3) that if a can precede a then every word in a^* is in the language. Thus since an unstressed syllable may follow another unstressed syllable in an unbounded pattern (in fact, in all patterns), there are well-formed words which consist solely of unstressed syllables! Consequently, a precedence language cannot describe exactly the QS unbounded stress patterns either.

Finally, we can ask whether combining precedence and n-gram learners as suggested in Chapter 4 §6 is a successful strategy. However, it is easy to see that the answer in this case must be that it does not. The problem is the same: since the precedence learner accepts strings of unstressed syllables as legal and the n-gram accepts strings of $n-1$ unstressed syllables as legal, intersecting the results keeps words which are strings of unstressed syllables. Therefore the combination offers no improvement over the n-gram learners.[3]

2.4 Unattested Stress Patterns

Despite the extensive variation recounted above, stress patterns are not arbitrary. There are many, many logically possible ways to assign stress which are unattested. No language places a stress on the fourth syllable from the right (or left) in words four syllables or longer, and on the first (or final) syllable in words three syllables or less. No stress pattern places stress on every fourth or every fifth syllable (cf. binary and ternary patterns above which place stress on every second or third syllable). Moving further afield, languages do not place stress on every nth syllable, where n is a prime number, nor on every nth syllable where n is equal to some prime number minus one. When we consider the myriad of logically possible ways stress can be assigned, the attested variation appears quite constrained.

3 Neighborhood-Distinctness

This section introduces a (near) universal property of the 109 stress patterns which provides an inductive principle learners can use to generalize correctly from limited experience.

For each of the distinct patterns in the typology, I constructed a finite state

[3] Bigram, trigram, 4-gram, and precedence learners could learn 11, 54, 75, and 0 of the 109 patterns respectively. Simulations which intersect the n-gram learner's grammar with the grammar obtained by the precedence learner make no improvement as expected.

acceptor such that only those words which obey the language's stress rules are recognized by the acceptor. The words these machines generate are strings of syllables, not segments. Thus this study abstracts away from the different ways languages determine the relevant quantity of a syllable. These machines are available electronically as part of the stress typology database (see §2.2).

3.1 Definition

The key idea is that each state in a finite state acceptor represents some phonological environment (for some related discussion see Riggle (2004)). Given the idea that phonological environments are 'local', we can identify each state with its local characteristics. Thus I define the neighborhood of a state as

(5)
1. the set of incoming symbols to the state
2. the set of outgoing symbols to the state
3. whether it is a final state or not
4. whether it is a start state or not

Thus the neighborhood of state can be determined by looking solely at whether or not it is final, whether or not it is a start state, what the set of symbols labeling the transitions which reach that state is, and what the set of symbols labeling the transitions which depart that state is. Pictorially, all the information about the neighborhood of a state is found within the state itself, as well as the transitions going into and out of that state. For example, suppose states p and q in Figure 5.1 belong to some larger acceptor. We can decide that states p and q have the same neighborhood because they are both nonfinal, nonstart states, and both can be reached by some element of $\{a, b\}$, and both can only be exited by observing a member of $\{c, d\}$.

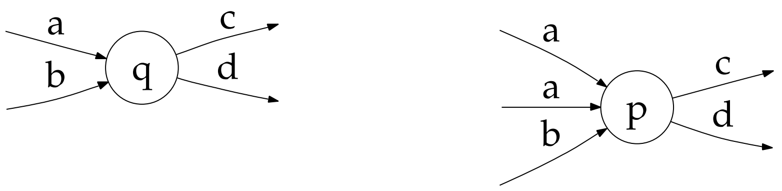

Figure 5.1: Two States with the Same Neighborhood

It is now possible to define acceptors that are neighborhood-distinct.

(6) An acceptor is said to be **neighborhood-distinct** iff no two states have the same neighborhood.[4]

The class of neighborhood-distinct languages is defined in (7).

(7) **The neighborhood-distinct languages** are those for which there is an acceptor which is neighborhood-distinct.

Neighborhood-distinctness is highly restrictive. The neighborhood-distinct languages are a (finite) proper subset of the regular languages over some alphabet Σ: all regular languages whose smallest acceptors have more than $2^{2|\Sigma|+1}$ states cannot be neighborhood-distinct (since at least two states would have the same neighborhood). Thus most regular languages are *not* neighborhood-distinct.

Recall from Chapter 2 Appendix B–3 that although many different acceptors can recognize the same language or pattern, certain ones are more useful than others. For example, a forward deterministic acceptor with the fewest states for a language is called the language's tail canonical acceptor and typically finite state patterns are represented with this acceptor. I will refer to such acceptors which are neighborhood-distinct (and the languages they recognize) as *tail-canonically*

[4] Also, the acceptor must be stripped; i.e. every state must be useful (see Appendix B–3.8).

neighborhood-distinct. However, another algebraically equivalent (though arguably less useful) choice is a backward deterministic acceptor with the fewest states for a language, which is called the language's head canonical acceptor. It will be useful to refer to head-canonical acceptors which are neighborhood-distinct (and the languages they recognize) as *head-canonically neighborhood-distinct.* These two acceptors can be computed from any acceptor which recognize a given pattern. Finally, I will refer to patterns which are either tail or head canonically neighborhood distinct simply as *canonically neighborhood-distinct.*

3.2 Universality

When we consider the typology of stress patterns, 97 are tail canonically distinct and 105 are head-canonically neighborhood distinct. Only two languages are neither tail nor head canonically distinct. In other words, 107 of the 109 types of languages in the stress typology are canonically neighborhood-distinct (documented in Appendix E–2). One of these two non-canonically neighborhood-distinct stress patterns is provably not neighborhood distinct, the pattern of Içuã Tupi (Abrahamson 1968). It remains an open question whether there is some neighborhood-distinct acceptor which recognizes the other one, which is the pattern of Hindi as described by Kelkar (1968). Nevertheless canonical neighborhood-distinctness is a (near) universal property of attested stress patterns.

3.3 Discussion

There is a question as to how serious a challenge the two languages which are not canonically neighborhood-distinct are serious obstacles to the hypothesis that all phonotactic patterns are canonically neighborhood-distinct. If the the two stress patterns in question were common, or from languages whose phonology was well-studied and uncontroversial, the challenge to the hypothesis would obviously be more serious. As it is however, we would like to know more about the patterns in the languages themselves.

Unfortunately, in the case of Içuã Tupi, this seems impossible as Abrahamson (1968:6) notes that the tribe is "almost extinct" with only two families alive at the time of his studies. According to his paper, Içuã Tupi places stress on the penult in words four syllables or fewer and on the antepenult in longer words. In metrical theory, one would say that final syllable extrametricality is invoked in words with five or more syllables, but not invoked in words with four or less syllables. Although his paper devotes only a few lines to the topic of word stress, there are no obvious errors and the description of the pattern is clear as are the illustrative examples. I see little alternative but to accept the pattern as genuine.

Nonetheless, there are other plausible possibilities which would render the Içuã Tupi pattern canonically neighborhood-distinct. For example, Abrahamson (1968) makes no mention of secondary stress. The presence of secondary stress can distinguish states (see discussion of Klamath and Seneca in §5.6 below). For example, if Içuã Tupi also exhibits secondary stress word-finally, then the pattern becomes neighborhood-distinct. Another possibility is that words of five syllables or longer may optionally place stress on the penult or on the antepenult. Although it may be unfair to assume this (as we can expect Abrahamson to have noted it), this alteration also makes the pattern neighborhood-distinct.

On the other hand, the stress pattern of Hindi has been the subject of many different proposals (as described in Hayes (1995)). There is little consensus as to what the stress pattern of Hindi actually is. However, even if each different description of Hindi were correct (perhaps because speakers belong to different dialectal groups), a small change to Kelkar's description renders it neighborhood-distinct. According to Kelkar, Hindi is a QS unbounded system with a threeway quantity distinction with main stress falling on the rightmost (nonfinal) heavy syllable, or in words with all light syllables, the penult. Secondary stresses fall on heavy syllables and alternate light syllables to the left of the main stress.[5] Kelkar's description of the stress patterns, however, rests on words that are only a few syllables in length. In other words, although his description makes clear predictions about how stress falls in longer words, it is far less clear that these predictions are actually correct. If, in some longer words, primary stress optionally occurs as secondary stress, then this pattern also becomes neighborhood-distinct.

To sum up, it is premature to reject the hypothesis that all patterns are canonically neighborhood-distinct because of the counter examples of Içuã Tupi and Hindi (per Kelkar). The proposed descriptions of these patterns ought to be investigated further if possible to see if they hold up as counterexamples. A small change in the description of the pattern may render it neighborhood-distinct. Finally, note that the hypothesis that all stress patterns are canonically neighborhood-distinct is supported by the many established stress patterns in the typology which fall into this class.

4 The Learner

Because the neighborhood-distinct languages form a finite class, there are actually many, many learners which can identify this class of patterns in the limit (Jain et al. 1999, Osherson et al. 1986). Consequently, there is much to explore.

In this section, I introduce a simple learner which only uses the concept of neighborhood to generalize. The idea is to merge same-neighborhood states in the finite state representation of the input. Note that to the extent this learner

[5] This may also well be the most complicated pattern in the typology, as measured by the number of states in its tail and head canonical acceptors: 32 and 29 respectively (cf. Pirahã which has 33 and 18, respectively).

succeeds, it explains why stress patterns are neighborhood-distinct. As it turns out, this learner does not identify this class of patterns in the limit, though it does succeed for many of the attested stress patterns.

I introduce the learner in two steps for easier exposition. The first step introduces a learner called the Forward Neighborhood Learner first, which succeeds on many, but not all, of the attested patterns. I argue that analysis of the languages which the Forward Neighborhood Learner fails to learn reveals that it is handicapped by the prefix tree representation of the input. I propose an additional alternative representation of the input—suffix trees—and a revised learner called the Forward Backward Neighborhood Learner, which succeeds on many more (but not all) of the attested patterns. Results and predictions are discussed in §5.

Note that unlike the learners in the previous chapters, the only known description of the learning functions presented here use the state merging scheme.

The learners are evaluated in the following mannar. For each acceptor representing some stress pattern, the learners below were provided samples generated by this acceptor. If the acceptor output by the learner recognizes the same language as the original acceptor, then that was counted as successful learning. This criterion establishes identification in the limit because the learners below have the property that there is some sample with which the learners succeed, and the learners continue to succeed with any strictly larger sample.

4.1 The Forward Neighborhood Learner

The Forward Neighborhood Learner merges states in the prefix tree which have the same neighborhood. I denote by M_{nd} the function which maps a prefix tree PT to the neighborhood-distinct acceptor obtained by merging all states in PT with the same neighborhood. Note that computing M_{nd} is efficient in the size of PT. This is because (1) merging two states is efficient (Hopcroft et al. 2001), (2) an algorithm need at most check every pair of distinct states for neighborhood-equivalence to determine if they should be merged, and (3) determining the neighborhood-equivalence of two states is efficient.[6] It is now possible to state precisely the Forward Neighborhood Learner (FNL). The Forward Learner successfully identifies 85 of the 109 pattern types as shown in Appendix E–2.

Algorithm 5 The Forward Neighborhood Learner

```
**Input:** a positive sample $S$
**Ouput:** an acceptor $A$.
Let $A = M_{nd}(PT(S))$ and output acceptor $A$.
```

These results also make clear that the languages in the range of the learning function are not the same as the neighborhood-distinct languages. The two classes of languages clearly overlap, but the Forward Learner does not identify the class of neighborhood-distinct languages in the limit. The Forward Learner does not even identify the tail-canonically neighborhood-distinct languages, falsifying the conjecture made in Heinz (2006b) that it does. Nonetheless, the results are promising because the languages for which the Forward learner succeeds cross-cuts the QI, QS bounded, and QS unbounded stress patterns, suggesting the learner is on the right track.

When we examine the languages for which the Forward Learner fails to learn we find that the error is always one of *overgeneralization*. This happens because states are merged which should be kept distinct. Consequently, the grammar returned by the learner accepts a language strictly larger than the target language. This means that there is some word for which the learner's grammar accepts different stress assignments. This can be construed as optionality—a particular string of syllables

[6] How efficient depends on the representation of the acceptors (i.e. as matrices or as tuples of sets).

can be stressed this way or that way.

Another characteristic that all stress patterns for which the Forward Learner fails share (except Kashmiri) is that they are typically analyzed with a metrical unit at the right word edge. Why would such languages be problematic for the Forward Learner? One idea is that the prefix tree's inherent left-right bias fails to distinguish the necessary states, and this occurs more commonly in languages analyzable with a metrical unit at the right word edge. If this were the case, the problem is not with the generalization procedure per se, but rather with the inherent left-right bias of the prefix tree. Below I propose another way the input to the learner can be represented as a finite state acceptor: suffix trees.

4.2 Suffix Trees

If the input were represented with a *suffix tree*, then the structure obtained has the reverse bias, a right-to-left bias. Like a prefix tree, a suffix tree is a finite state representation of the input: it accepts exactly the words from which it was built and nothing else. A suffix tree is structured differently from a prefix tree, however, because each state now represents a unique suffix in the sample instead of a prefix. Whereas a prefix tree is forward deterministic, a suffix tree is reverse deterministic. A formal definition is provided in Appendix E–1, where it is also proven that a suffix tree can be constructed in terms of a prefix tree given some sample. This procedure runs as follows: Given a sample of words, build a prefix tree reading each word *in reverse*. Since the resulting prefix tree accepts exactly the reverse of each word in the sample, *reverse* this tree by changing all final states to start states, all start states to final states, and changing the direction of each transition (see Appendix B–3.5). The resulting acceptor is a suffix tree and accepts exactly the words in the sample.

Figure 5.2 shows a suffix tree constructed from all words which obey the 'Rightmost Heavy Otherwise Leftmost' stress pattern of Selkup up to four syllables in length. Compare the structure of the suffix tree of this representation to the prefix tree shown in Figure 5.3. Though both representations accept exactly the same finite input, they are structurally different and not mirror images of each other. The two trees have different structures—though both accept exactly the same (finite) set of words. Because they have different structures, the states in a suffix tree may have different neighborhoods than the states in a prefix tree. Consequently, the generalizations acquired by merging states with the same neighborhoods may be different.

4.3 The Forward Backward Neighborhood Learner

The Forward Backward Neighborhood Learner is very simple. Let M_{nd} be the function which maps an acceptor to the acceptor obtained by merging same-neighborhood states. Let PT and ST denote functions which map a finite sample to the prefix tree and suffix tree, respectively, which accepts exactly the given sample. The learner simply applies the N_{nd} to the prefix and suffix tree representations of the samples and intersect the results. This learner succeeds on 100 of the 109 patterns (414 of 422 languages), a considerable improvement over the Forward Learner. The

Algorithm 6 The Forward Backward Neighborhood Learner

```
**Input:** a positive sample $S$
**Ouput:** an acceptor $A$.
Let $A_1 = M_{nd}(PT(S))$.
Let $A_2 = M_{nd}(ST(S))$.
Let $A = A_1 \cap A_2$ and output the acceptor $A$.[7]
```

[7] Intersection ($\cap$) of two acceptors A and B results in an acceptor which only accepts words accepted by both A and B (See §B–3.4).

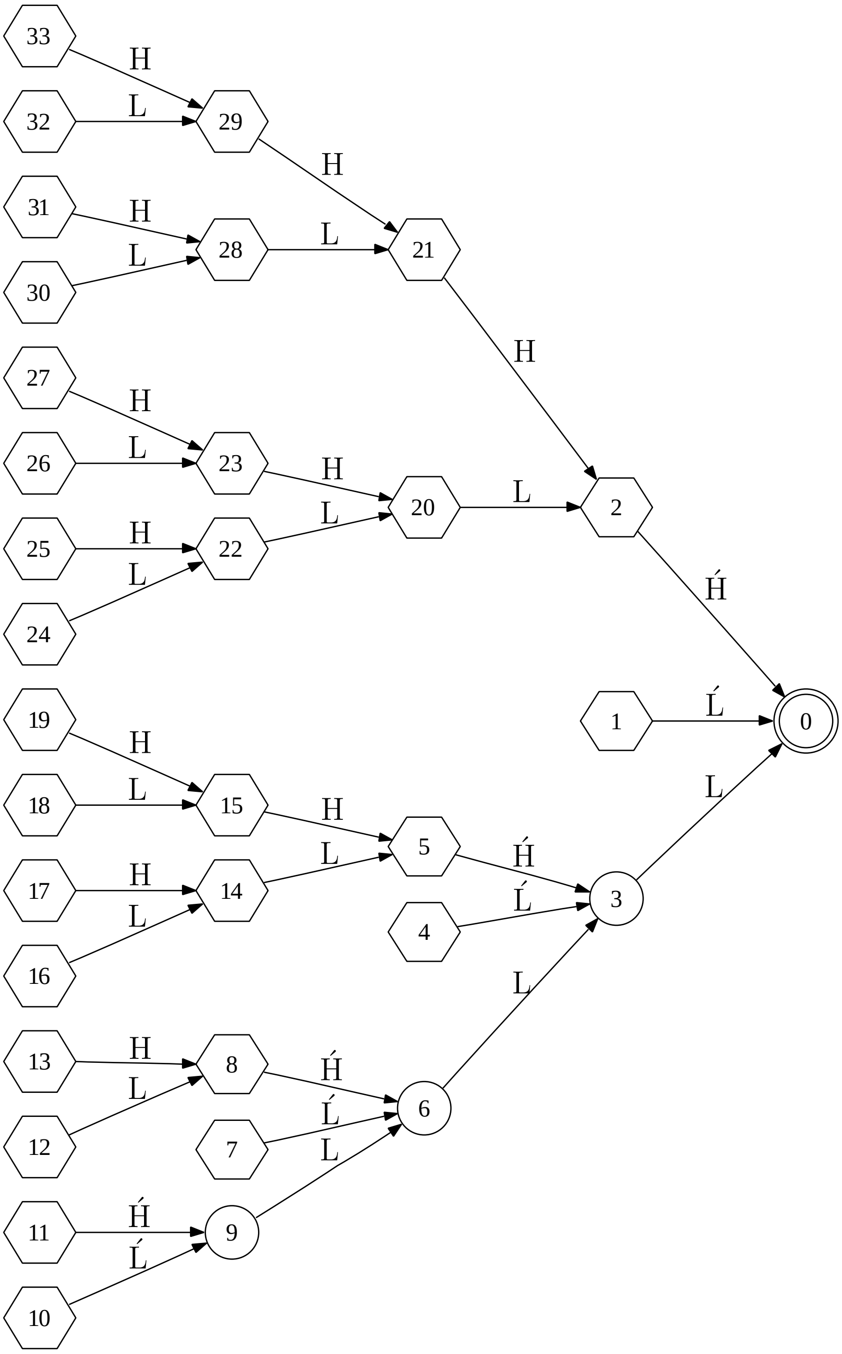

Figure 5.2: Suffix Tree for Stress Pattern of Selkup for All Words Four Syllables or Less

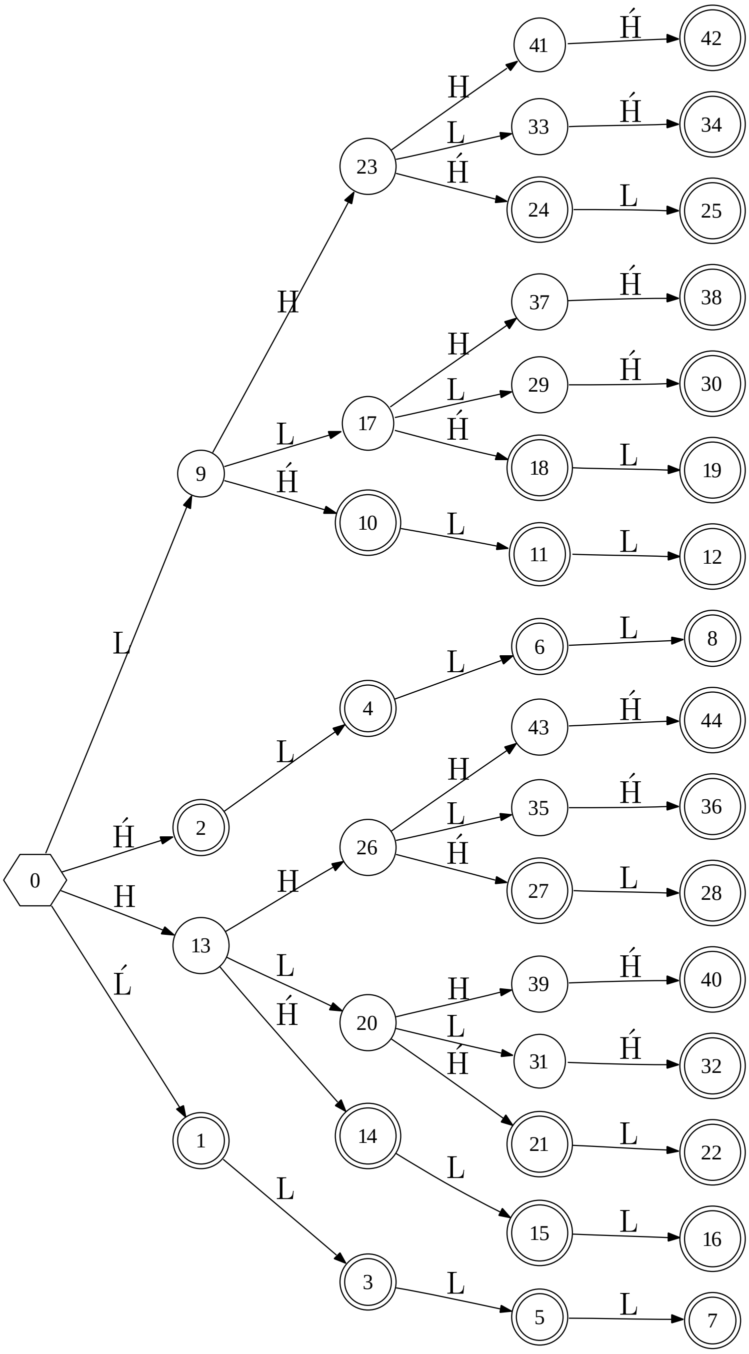

Figure 5.3: Prefix Tree for Stress Pattern of Selkup for All Words Four Syllables or Less

appendix in §E–2 provides these results, along with those of the Backward Neighborhood Learner (which generalizes only by merging same-neighborhood states in the suffix tree).

5 Discussion

In this section I explain why the Forward Backward Neighborhood Learner works as well as it does, demonstrate that significant classes of logically possible stress patterns cannot be learned by this learner, discuss the attested patterns it fails to learn as well as additional predictions made by the learner.

5.1 Basic Reasons Why the Forward Backward Learner Works

The reason the Forward Backward Learner succeeds in more cases than the Forward Learner is simple: intersection keeps the robust generalizations. The robust generalizations are the ones made in *both* the prefix and suffix trees. Overgeneralizations that are made by the Forward Learner are not always made by merging same-neighborhood states in the suffix tree. Consequently, those that are not do not survive the intersection process. Likewise, it is also true that overgeneralizations made by merging same-neighborhood states in the suffix tree are not always made in the prefix tree.

However, the generalization strategy itself—the merging of same-neighborhood states—is the real reason for the algorithm's success. Consider again the Forward Leaner. By merging states with the same neighborhood, the algorithm guarantees that its output is neighborhood-distinct. Similarly, when the same-neighborhood states are merged in the suffix tree, the resulting acceptor is neighborhood distinct. The learner—by merging same-neighborhood states—generalizes to neighborhood-distinct patterns. Thus if people generalize similarly, it explains why nearly all stress patterns are neighborhood-distinct.

There is one caveat, however. As explained in Chapter 6, the class of neighborhood-distinct languages is not closed under intersection. Thus when the Forward Backward Neighborhood Learner intersects the two acceptors obtained by merging same-neighborhood states in the prefix and suffix trees, the resulting language is not guaranteed to be neighborhood distinct. Little is understood about what additional properties are necessary to ensure that neighborhood-distinctness survives the intersection process. Whatever those properties are, they appear to be in play here. The patterns obtained via the intersection process in the current study produced a tail or head canonically neighborhood-distinct pattern for every pattern in the study save Ashéninca.

5.2 Input Samples

As with the n-gram and precedence learners, we can ask what kind of input sample is necessary for the learner to succeed when it does. Unlike those other learners however, the characteristics of a sufficient sample—that is, a sample which guarantees the learner converges to the target grammar—are not yet known for neighborhood-based learners. Consequently, it is difficult to determine whether the kinds of samples the learner requires to succeed are present in the linguistic environment of children. This section argues that there is reason to be optimistic, despite the lack of a characterization.

First, let me make clear the samples that were given to the learner in the simulations. They consisted of all words form one to n syllables which obeyed the stress pattern (recall that stress pattern here includes restrictions on what syllable sequences are allowed, see §2.1.1). If the learner failed for some given n, then n was increased by one and the learner tried again. If the learning did not occur by the time n equaled nine, or in some cases eight, I concluded the learner failed.

For example, Table 5.3 in Appendix E–2 shows that the Forward Backward Learner succeeded in learning the stress pattern of Mam when provided a sample which consisted of all words with one to five syllables. Because this languages makes a threeway distinction between syllable types, this means there $3^5 = 343$ words that made up the sample.[8]

This method of sample construction provides a very crude idea of what constitutes a sufficient sample. We can only conclude from the above demonstration that a sufficient sample for the stress pattern exemplified by Mam must include at least one word of length five syllables. In fact, when simple QI patterns are considered, it is easy to see that a single word of sufficient length is all that is needed to generalize correctly to infinitely many longer words. Thus in all likelihood, the size of a characteristic sample is much smaller than the size of the samples given in the simulations suggest, which increases our confidence that such a sample is plausibly found in the linguistic environment of children.[9]

5.3 Unlearnable Unattested Patterns

It is also interesting to note that most unattested patterns cannot be learned by the Forward Backward Neighborhood Learner. Intuitively, this follows from the fact that neither the Forward Learner nor Backward Learner can ever learn a non-neighborhood-distinct pattern (of which there are infinitely many).

[8] Learners were only given words which made the necessary syllable distinctions. It is easy to see, however, that if QI learners were given words which made some (superficial) light-heavy distinction that the learner would still converge to the correct grammar. This is because additional (i.e. unnecessary) syllable distinctions do not change the character of the neighborhoods in the target grammars. It does mean that the sample size becomes larger, which makes the question of what constitutes a characteristic sample more pressing.

[9] Our confidence should also increase when we recall that there are additional properties of stress patterns not considered by the neighborhood-learner (which only makes use of a particular formulation of locality when generalizing) that learners likely make use of. See §5.6.

For example, logically possible unattested stress patterns such as those which place stress on every fourth, fifth, sixth, or nth syllable cannot be learned. To see why, consider the acceptor in Figure 5.4 which generates the logically possible stress pattern which assigns stress to the initial syllable and then every fourth syllable. The reason is that this pattern cannot be learned by the Forward Backward Neighborhood Learner because states 2 and 3 have the same neighborhood. It is not possible to write some other acceptor for this language that would not have two states like states 2 and 3 above with the same neighborhood (because the pattern requires exactly three unstressed syllables between stresses). Thus this pattern is not neighborhood-distinct. Consequently neither the Forward Learner nor the Backward Learner could ever arrive at this pattern by merging same-neighborhood states since states 2 and 3 (or more precisely, their corresponding states in the prefix and suffix trees) would always be merged. Furthermore, since this overgeneralization is made by both learners, it survives the intersection process. Thus the result obtained by the Forward Backward Learner is that secondary stresses must occur *at least* two syllables apart. In a sense, the learner fails because it cannot distinguish 'exactly three' from 'at least two.' In this way, the Forward Backward Learner cannot makes concrete the idea that "linguistic rules cannot count past two" (Kenstowicz 1994:597). Whether children or adults behave similarly in artificial language learning experiments is an open question.

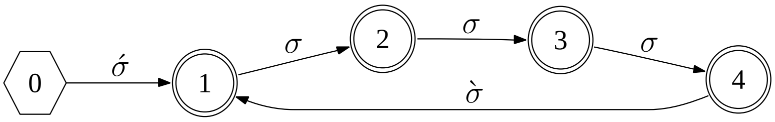

Figure 5.4: The FSA for a Quaternary Stress Pattern

5.4 Unlearnable Attested Patterns

In this section, I discuss the eight languages which the Forward Backward Neighborhood Learner failed to learn. Again, in every case the learner failed because it overgeneralized. Thus for certain words, although the grammar obtained by the learner places stress in the correct positions, it can also place stress in other positions. In other words, the learner allows a certain degree of optionality. I address these failures in the next section below.

The concrete reason why all of these patterns fail is because there are two states which are merged which should not be. In other words, the learner does not distinguish phonological environments where it should have. To make it more concrete than that requires careful examination of the canonical acceptors and the prefix and suffix trees, and space and time prohibit such an extended discussion. Therefore in what follows, I only make a few observations. Undoubtedly, more questions are raised than can be answered.

Two of the languages for which it fails, Içuã Tupi and Hindi (per Kelkar), are not canonically neighborhood distinct and are discussed in §3.3.

Mingrelian is a neighborhood-distinct pattern which places primary stress initially and secondary stress on the antepenult. The Forward Backward Neighborhood Learner fails because it cannot distinguish the sequence of two unstressed syllables at the end of the word from similar sequences in the middle of the word.

The stress patterns of Palestinian Arabic, Cyranaican Bedouin Arabic, Hindi per Fairbanks are all not learnable by this learner, though they are neighborhood-distinct. It is striking that these are precisely the patterns in the typology that have been analyzed with extrametrical feet (Hayes 1995). It appears that patterns describable with extrametrical feet are beyond the range of the learning function.

Ashéninca and Pirahã are two other patterns which are neighborhood-distinct but beyond the the range of the learning function. These patterns are well-known prominence systems. However, I suspect the reason the Forward Backward Learner fails has less to do with this, than with the fact that both of these languages, like the ones above, can place stress on the third syllable (or the fourth in the case of Ashéninca) from the right edge in particular circumstances. It seems that the Forward Backward Neighborhood learner can learn only some patterns like this (e.g. Walmatjari).

5.5 Addressing the Unlearnable Attested Patterns

Given the hypothesis that the stress patterns are in the range of the FBL learning function, but eight of the stress patterns are not learned by the algorithm, there are two possibilities: the stress patterns as described are incorrect, or the hypothesis is false. I consider both possibilities here.

First, I consider the possibility that the stress patterns have not been accurately described. With the exception of Kelkar's description of Hindi, all of the patterns that fail regularly place stress in certain words on the third syllable from the right word edge. There are many stress patterns which place stress on the antepenultimate syllable and are in the range of the FBL. What accounts for the difference?

One instructive case comes from Mingrelian, which recall places primary stress on the initial and secondary stress on the antepenult. A similar pattern is found in Walmatjari, which *optionally* places stress on the penult or antepenult in longer words. The pattern of Walmatjari is learnable because the states in the acceptor which generate the pattern are made distinct in the suffix tree by the optional penult pattern that occurs in longer words. Interestingly, these are the only two QI dual languages in the typology which place primary stress close to the left word edge and secondary stress on the antepenult. Furthermore, if Mingrelian places secondary stress on the penult in trisyllabic words, even optionally, the stress pattern is now learnable (as the relevant states are now distinct in the suffix tree). Given that the data from Mingrelian comes from a single source (Klimov 2001), it may very well be the case that there is optionality in Mingrelian.[10]

Similarly, if the FBL is correct, the prediction is that the stress patterns in the other languages are different from what has been described. In this respect, it is worth pointing out that the patterns for which the learner fails are ones where consensus has formed over a somewhat small data set. This does not mean that that the actual patterns are completely different from what previous researchers described. In fact, the patterns can differ minimally in interesting ways and even include the same set of words that earlier researchers used to develop their own hypotheses. The two ways that I am suggesting here are (1) in certain words, there will be optionality and (2) in languages currently described as lacking secondary stress, there may in fact be secondary stress. Because theory helps direct the course of investigation, it is plausible that these might be overlooked (or in the case of secondary stress, difficult to detect) in earlier hypothesis formation.

Turning to the second possibility, it may be the case that hypothesis that the FBL is correct is wrong and that the descriptions are 100% accurate. At this point it is useful to recall that because the canonically neighborhood-distinct class is finite, there are many learners for this class which identify it in the limit, of which the FBL is not one.

The fact that the FBL fails for stress patterns that are describable with a rule of foot extrametricality (Palestinian Arabic, Cyranaican Bedouin Arabic, Hindi per Fairbanks, see Hayes (1995)) shows that not all patterns describable in standard

[10] I am currently in the process of obtaining this source.

metrical theory (Hayes 1995) can be learned by the FBL.[11] The source of this conflict is not well understood except at the most superficial level: the locality conditions imposed by the FBL learner are not met in patterns describable with extrametrical feet.

However, if the locality conditions were adjusted by extending the notion of neighborhood to include incoming and outgoing paths of length two then in fact all the languages in the typology can be identified in the limit by a similarly modified FBL. In other words, the neighborhood discussed so far may be considered a '1-1 neighborhood' where '1-1' means 'incoming paths of length 1 and outgoing paths of length 1', and in fact all patterns are '2-2 neighborhood-distinct' and learnable by a '2-2 FBL'.[12]

Chapter 6 investigates this more general concept of the neighborhood. This parameterization of the 'j-k' neighborhood-distinct languages turns the hypothesis space into a infinitely large space, where the most common patterns are the ones found for small values of j and k, and rarer patterns require larger values of j and k (though still no larger than 2).[13] Although this infinite hypothesis space is neither Gold- nor PAC- learnable (since for any regular language there is some j and k for which it becomes j-k neighborhood-distinct), formal learning theorists are

[11] It is worth asking if there are other stress patterns that are 'natural' in some sense, yet either non-neighborhood-distinct or non-learnable by the FBL. This is project beyond the current scope of this dissertation. One way to proceed might be to see whether the stress patterns generated in a factorial typology of OT constraints are learnable by the FBL. Proposals include Eisner (1998), Tesar (1998), and Kager (1999).

One potentially problematic pattern is one where alternating stress occurs on both sides of a primary stress. Different states for the alternating pattern are required to keep track of whether the primary stress has been seen, but the states themselves may have the same neighborhoods. This is like the Yidiɲ pattern, except Yidiɲ is neighborhood-distinct and FBL learnable because of the distinction between heavy and light syllables.

[12] Since all trigram and precedence languages are 1-1 neighborhood-distinct, it is easy to see that they are also 2-2 neighborhood-distinct.

[13] Of course, extending the size of the neighborhood in this way also allows quaternary stress patterns to be learned by the learner. Thus it would be claimed that such patterns are learnable though not as easily learned as the more common patterns, which are neighborhood-distinct for smaller values of j and k.

interested in precisely this kind of problem and are seeking to develop a framework of learning which investigates how learning can proceed in these kinds of hypothesis spaces (Pitt 1989, Angluin 1992, de la Higuera 1997).

5.6 Other Predictions

Other predictions follow from the proposal that stress patterns are in the range of the Forward Backward Learner. As mentioned, the FBL predicts that the presence or absence of secondary stress matters for learnability. It was mentioned above that adding secondary stress can make unlearnable patterns learnable. Removing it can also make learnable patterns unlearnable. For example, It was discovered that if secondary stress is excluded from the grammars of Klamath and Seneca, then the Forward Backward Neighborhood Learner fails to learn these grammars. It fails because, in the actual grammars of Klamath and Seneca, the presence of secondary stress *distinguishes* the neighborhoods of certain states of the prefix and/or suffix trees.

The Forward Backward Neighborhood Learner can also learn unattested patterns that are unnatural. In such cases, it is important to remember that learner developed here only examines the contribution that locality can make to learning. For example, consider the logically possible stress pattern 'Leftmost Light Otherwise Rightmost' (LLOR). Whether or not humans can learn such a pattern is an open question. However, even if it were shown that LLOR is more difficult to learn than the more natural 'Leftmost Heavy Otherwise Rightmost' pattern, the fact is plausibly due to considerations separate from locality (e.g. the Weight-to-Stress Principle (Prince 1992, Gordon 2006)).

5.7 Comparison to other Learning Models

Here I compare the Forward Backward Learner to the ordered cue-based learner in the Principles and Parameters framework (Dresher and Kaye 1990, Gillis et al. 1995), a perceptron-based learner (Gupta and Touretzky 1994), and an OT-based learner (recursive constraint demotion with robust interpretive parsing) (Tesar 1998, Tesar and Smolensky 2000). Like the Forward Backward Learner, each of these learning models was evaluated with respect to stress patterns. However, exact comparisons are not possible because each learner was tested on a different set of stress patterns with different kinds of input samples.

Gillis et al. (1995) implement the cue-based model presented in Dresher and Kaye (1990). The ten parameters yield a language space consisting of 216 languages. The language space is based on actual stress patterns but does not include all attested stress types. The learner discovers parameter settings compatible with 75% to 80% of these languages when provided a sample of all possible words from one to four syllables. As Dresher (1999) notes, it is possible (though unknown) that accuracy increases if longer words are admitted into the sample.

Gupta and Touretzky (1994) present a perceptron with nineteen stress patterns, of which it successfully learns seventeen. The training input consists of a sample of all words of length seven syllables and less, and is presented to the perceptron at least seventeen times. This is the smallest number of times that resulted in successful learning any of the nineteen patterns (e.g. the perceptron learned Latvian, a QI single system with word-initial stress, when presented with such training input). The largest number of presentations of the sample is 255 (for Lakota, a QI single system which places stress on the peninitial syllable). If the perceptron is given a training sample of shorter words, it is able to learn the two patterns which it otherwise fails to learn.

Tesar and Smolensky (2000) report 12 constraints which yields a space with 124 languages. Like the language space in the P&P model above, this is an artificial language space based on actual languages (but not all attested patterns are included). If the initial state of the learner is monostratal—that is, no a priori ranking—then the learner succeeds on about 60% of the languages. When a particular initial constraint hierarchy is adopted, the learner achieves ∼97% success.

The FBL is certainly simpler than the P&P and OT learners in the sense that it uses fewer a priori parameters. It is not exactly clear how to count the number of a priori parameters of each model since that requires placing them all on a level playing field. But certainly the FBL, which has no a priori P&P parameters or OT constraints, is much simpler. The speed at which the FBL converges (measured by sample size) appears slower than both of these models; this is almost certainly related to the fact that the hypothesis space of the FBL is so much larger.

When the FBL is compared to the perceptron learner, it is less clear which is the simpler model. However, the perceptron learner is much, much slower than the FBL as it requires repeated presentations of words.

However, the main advantage the FBL has over the other models is that the locus of explanation now resides in the learning process. In fact (with the one caveat mentioned earlier) we can say that the reason stress patterns are neighborhood-distinct is because learners generalize from their experience in the way predicted by the FBL. In this way, the FBL is more explanatory than the other models, where the locus of explanation lies in the parameters, the constraints, or is obfuscated.

Consider the OT learner. There, the learner works because of additional structure imposed by the nature of an OT grammar over the hypothesis space. If the constraints were different—say they allowed patterns describable with feet of size four or five syllables—then learning would proceed as before with the same success over this (false) hypothesis space for stress patterns, despite the fact that such patterns are unattested. The locus of explanation in OT is the content of the constraints themselves, i.e. the content of Con.

One possibility for OT is to restrict the kinds of constraints allowed in Con so that only neighborhood-distinct constraints are allowed (cf. Eisner (1997a), McCarthy (2003)). This is a serious restriction and has the effect of eliminating patterns from the subsequent factorial typology that can be described with feet of size four or five syllables. It also has the effect of allowing OT practitioners to continue to use most of the standard kinds of constraints (alignment constraints of course the notable exception).

In the same way, that one can ask of OT, why these constraints and not some others, one can also ask the question: Why this learning function instead of some other learning function? For example, stress patterns, and phonotactic patterns in general are not found in the full range of the zero-reversible languages (see Angluin (1982)). Is there a difference between stipulating constraints in OT and stipulating inductive principles that learners can use? The answer to this last question is Yes, there is a difference. The fact is that moving the discussion from why this constraint, to why this learning algorithm is an advance because it is a simpler platform from which we can explain aspects of the nonarbitrary character of the observed patterns. In this respect, this goal is no different from the one which seeks to explain the nature of constraints in terms of phonetic—or perhaps more generally articulatory or perceptual—naturalness (Myers 1997, Hayes 1999, Steriade 2001, Hayes et al. 2004).

Finally, let me be clear that the FBL does not compare to OT, as a theory of phonology. This is simply because OT can do many things the FBL cannot, e.g. describe patterns of alternation. Nonetheless, in the domain of learning stress patterns (and phonotactic patterns in general, see Chapter 6), the FBL, due to its explanatory power, offers insight into the kinds of stress patterns that OT currently stipulates in an a priori constraint set.

6 Summary

Two recent surveys (Bailey 1995, Gordon 2002), when put together, yield a typological survey of 422 stress languages and 109 distinct stress patterns, representing over 70 language families. For each of these stress patterns, I constructed a finite state acceptor representing it. 107 of these patterns are tail or head canonically neighborhood-distinct—that is, are made up of phonological environments (states) that are uniquely defined by local properties. Thus one hypothesis put forward in this chapter are that all stress patterns are canonically neighborhood distinct.

Neighborhood-distinctness is not only interesting because it is a novel formulation of locality in phonology and a (near) universal of attested stress patterns, but also because it naturally provides an inductive principle learners can use to generalize. The Forward Backward Neighborhood Learner, which generalizes by merging same-state neighborhoods in prefix and suffix trees, correctly learns 100 of the stress patterns. Another hypothesis put forward in this chapter is that stress patterns fall within the range of the Forward Backward Neighborhood learning function.

These two hypothesis are expressed in Figure 5.5. Note that it is not known which is the stronger or weaker hypothesis because the exact 'size' of both of these language classes is unknown. Certainly in terms of raw numbers given the attested typology, the former hypothesis fares better. Only two stress patterns in the typology lie outside the domain of the canonically neighborhood-distinct languages. The range of the Forward Backward Learner, which again does not line up exactly with the canonically neighborhood-distinct class, excludes more of the attested stress patterns.

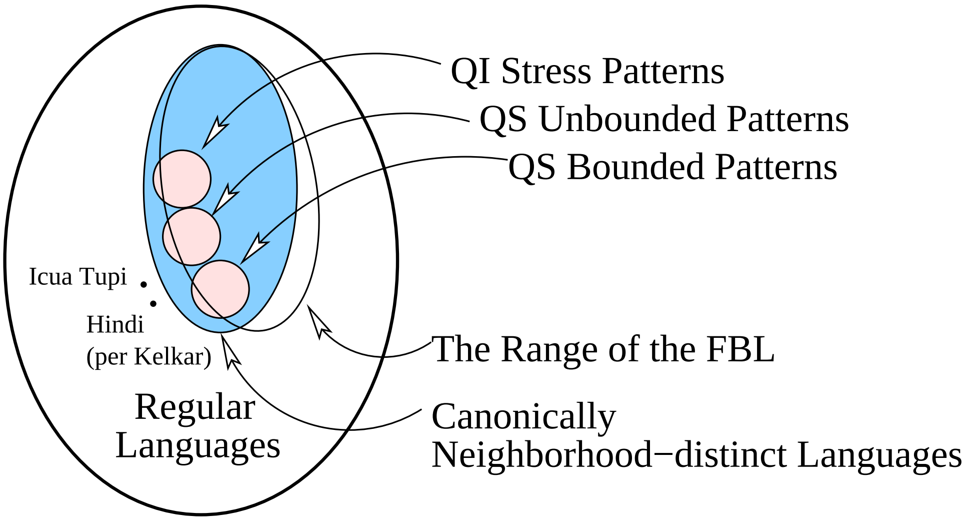

Figure 5.5: The Stress Typology

Although the learner does not learn all of the patterns, it is striking that such a simple procedure learns so many. Additionally, most unattested logically possible stress patterns cannot be learned by this learner. Therefore, the the class of languages in the range of the proposed learning function approximate the attested patterns in a nontrivial, interesting way.

Appendices

E–1 A Formal Treatment of Suffix Trees

A *suffix tree* is an acceptor constructed from a finite sample like a prefix tree. Define $ST(S) = (Q, I, F, \delta)$ as follows:

$$\begin{aligned} Q &= \{Sf(S)\} \\ I &= \{S\} \\ F &= \{\lambda\} \\ \delta(au, a) &= u \text{ whenever } u, ua \in Q \end{aligned}$$

For any S, $ST(S)$ is a backward deterministic acceptor that accepts exactly S.

The following lemma and corollary establish a nontrivial relationship between suffix trees and prefix trees.

Lemma 22 For any positive sample S, $PT(S^r)^r$ is isomorphic to $ST(S)$.

Proof: For some positive sample S, let $PT(S^r) = (Q, I, F, \delta)$ and $ST = (Q', I', F', \delta')$. By definition $PT(S^r)^r = (Q, F, I, \delta^r)$. $h(u) = u^r$ is the bijection we need. $Q = \{Pr(S^r)\} = \{Sf(S)^r\}$ by Lemma 7. For all $u \in Q$, $h(u) = u^r \in Sf(S)$. By the definition of suffix trees, $u^r \in Q_\prime$. Thus by the nature of the reverse operation, then h is one to one and onto. Since $h(\lambda) = \lambda$, $h(I) = F'$. Similarly $h(F) = I'$. Finally for any $u \in Pr(S^r)$, $a \in \Sigma$, $\delta(u, a) = ua$ whenever $u, ua \in Pr(S^r)$ implies $\delta^r(ua, a) = u$. It is necessary to show that $h(\delta^r(ua, a)) = \delta'(h(ua), a)$. Consider: $h(\delta^r(ua, a)) = h(u) = u^r = \delta'(au^r, a) = \delta'(h(ua), a)$ since $h(ua) = au^r$. □

Corollary 15 For any positive sample S, $ST(S^r)^r$ is isomorphic to $PT(S)$.

E–2 Results of the Neighborhood Learning Study

The tables below are interpreted as follows. In the ‘FL’, ‘BL’, and ‘FBL’ columns, circled numbers mean the Forward Learner, the Backward Learner, and Forward Backward Learner identifies the pattern, respectively. The number inside the circle indicates which forms were necessary for convergence. Specifically, ⓝ means the learner succeeded learning the pattern with a sample of words consisting of one to n syllables. The ‘Notes’ column indicates whether or not there are any phonotactic restrictions (which the sample obeys) or other relevant information. In particular, X and Y indicate whetherthe stress pattern is not tail or head canonically neighborhood-distinct, respectively. Thus, absence of X (Y) indicates tail (head) canonical neighborhood-distinctness. Table 5.6 provides an explanation of the notes. The ‘Name’ column provides the name of a language in the typology which exemplifies the pattern, which is uniquely identified by the number in the ‘#’ column.

The ‘Main’ column contains the Syllable Priority Code (SPC), which was developed by Bailey (1995) as a shorthand for indicating primary stress assignment rules. The last character of the SPC (*L* or *R*) indicates from which edge of the word to begin counting. Thus the initial syllable is designated 1L, the peninitial 2L, the penultimate 2R, and the final syllable 1R. Thus the simplest SPC codes, such as *1L* (Afrikaans), simply mean main stress falls on the initial syllable.

Generally, more complex SPCs can be read as a series of if-then-else statements. Slashes indicate a quantity-sensitive rule with rules governing heavier syllables occurring left of the slash. Thus the SPC *12/2L* (Maidu) unpacks to the following: If the initial syllable is heavy, it gets stress, else if the peninitial syllable is heavy, it gets stress, else stress falls on the peninitial syllable. If the numbers are suffixed with *@s*, it means primary stress is assigned if the syllable position carries secondary stress.

Unbounded patterns, where the stress can fall any distance from the word edge, use the *12..89* construct. For example, the SPC for Amele *12..89/1L* unpacks to the following: If the first syllable counting from the left is heavy then it receives primary stress, else if the second syllable counting from the left is heavy then it receives primary stress ... otherwise (if there are no heavy syllables) the first syllable counting from the left receives primary stress. Since words are unbounded in length, Bailey (1995) uses *..89* to indicate "and so on" in the increasing order for any length. Thus 89 do not literally mean the 8th or 9th syllable. Rather 9 means the farthest syllable from the relevant edge and 8 means the next-to-farthest syllable from the relevant edge and so on. See Bailey (1995) for more details.

SPCs that are followed by (*n*+) means the code only applies to words that have at least *n* syllables. Likewise SPCs that are followed by (*n*-) means the code only applies to words that have at most *n* syllables.

The 'Secondary' column contains extensions I made to the SPC in order to describe secondary stress patterns. 'None' of course means that no secondary stress is present. 'Not included' means that source material reports secondary stresses, but that either 1) the source material did not describe it, usually because it was deemed too complex, or 2) the source material did describe it, but the pattern was either unclear or too complicated for me to incorporate into the study due to the usual suspect: time.

Since secondary stress patterns are often iterative (that is can be described recursively once the position of one stress is known), I indicate secondary stress patterns that can be described iteratively with the prefix *i-*. The prefix *i2* means the second syllable from a stress receives a stress (in both directions). The first stress is indicated with a SPC suffixed with a @ symbol. Thus *i2@1L* (Bagandji) indicates secondary stresses fall on odd syllables from the left, whereas *i2@2R* (Anejom) indicates secondary stresses fall on even syllables from the right. *@m* means that the first stress upon which the iterative procedure is based is the position of main stress. *@mL* means the iterations proceeds only leftwards of main stress. Likewise, *@mR* means the iterations proceeds only rightwards of main stress.

When the secondary stress rules are quantity-insensitive, I use H,L,X to designate heavy, light, and either heavy or light syllables, respectively. Thus a typical trochaic pattern is designated *i(’H,’LL)* and a typical iambic pattern *i(H’,LX’)*. If the iterative procedure begins from the word edge (as opposed to from a particular position), I forgoe the connective @ and just suffix *L* or *R* to indicate whether the pattern proceeds from the left or right edge, respectively. Thus *i(’H,’LL)R* (Inga) means trochees are iteratively constructed from the right word edge.

Whenever only heavy syllables bear secondary stress, I indicate this with *H*. Sometimes it is necessary to explictly mention that secondary stress only precedes main stress (as in cases describable with foot extrametricality), in which case I use the symbol <.

Table 5.1: Quantity-Insensitive Single and Dual Patterns

#	Name	Main	Secondary	Note	FL	BL	FBL
SINGLE							
1.	Afrikaans	1L	None		④	④	④
2.	Abun West	1R	None		④	④	④
3.	Diegueno (roots)	1R	None	B	④	④	④
4.	Agul North	2L	None		⑤	⑤	⑤
5.	Alawa	2R	None		⑤	⑤	⑤
6.	Mohawk	2R	None	A	⑤	⑤	⑤

Continued on next page

#	Name	Main	Secondary	Note	FL	BL	FBL
7.	Cora	1L (2-), 3R (3+)	None		⑥	⑥	⑥
8.	Paamese	3R (3+), 1L (2-)	None	B,X	×	⑥	⑥
9.	Bhojpuri	3R (4+), 2R (3-)	Not included	X	×	⑥	⑥
10.	Icua Tupi	3R (5+), 2R (4-)	None	X,Y	×	×	×
11.	Bulgarian	lexical	None		④	④	④
Dual							
12.	Gugu-Yalanji	1L	2R		⑥	⑥	⑥
13.	Sorbian	1L	None (3-), 2R (4+)	X	×	⑥	⑥
14.	Walmatjari	1L	2R or 3R (5+), 2R (4), None (3-)	Y	×	⑥	⑥
15.	Mingrelian	1L	3R (4+), None (3-)	X	×	×	×
16.	Armenian	1R	1L		⑤	⑤	⑤
17.	Udihe	1R	None (2-), 1L (3+)		⑤	⑥	⑥
18.	Anyula	2R	1L (4+), None (3-)		⑥	⑦	⑦
19.	Georgian	3R (3+), 2R (2-)	1L (5+), None (4-)		⑦	⑧	⑧

Table 5.2: Quantity-Insensitive Binary and Ternary Patterns

#	Name	Main	Secondary	Note	FL	BL	FBL
Binary							
20.	Bagandji	1L	i2@1L		⑤	⑤	⑤

Continued on next page

#	Name	Main	Secondary	Note	FL	BL	FBL
21.	Maranungku	1L	i2@1L	B	⑤	⑤	⑤
22.	Asmat	1R	i2@1R		⑤	⑤	⑤
23.	Araucanian	2L	i2@2L		⑥	⑥	⑥
24.	Anejom	2R	i2@2R		⑥	⑥	⑥
25.	Cavinena	2R	i2@2R	A	⑥	⑥	⑥
Binary with Lapse							
26.	Anguthimri	1L	i2@1L, no 1R		⑥	⑥	⑥
27.	Bidyara Gungabula	1L	i2@1L, no 1R	A	⑥	⑥	⑥
28.	Burum	1L	i2@1L, optional no 1R		⑤	⑤	⑤
29.	Garawa	1L	i2@2R, 1L, no 2L		⑥	⑥	⑥
30.	Indonesian	2R	i2@2R, 1L, no 2L (4+), None (3-)	X	×	⑧	⑧
31.	Piro	2R	i2@1L, 2R, no 3R		⑥	⑦	⑦
32.	Malakmalak	12@sL (3+), 1L (3-)	i2@2R (3+), None (3-)		⑥	⑥	⑥
Binary with Clash							
33.	Gosiute Shoshone	1L	i2@1L, 1R		⑤	⑥	⑥
34.	Tauya	1R	i2@1R, 1L		⑥	⑤	⑥
35.	Southern Paiute	2L (3+), 1L (2-)	i2@2L, 2R, no 1R (3+), None (2-)	B, Y	⑦	×	⑧
36.	Biangai	2R	i2@2R, 1L		⑦	⑥	⑦
37.	Central Alaskan Yupik	1R	i2@2L	B	⑥	⑥	⑥

Continued on next page

#	Name	Main	Secondary	Note	FL	BL	FBL
Ternary							
38.	Cayuvava	1L (2-), 3R (3+)	None (2-), i3@3R (3+)	A , X	×	⑧	⑨
39.	Ioway-Oto	2L	i3@2L		⑦	⑧	⑧

Table 5.3: Quantity-Sensitive Bounded Patterns

#	Name	Main	Secondary	Note	FL	BL	FBL
Leftmost Heavy otherwise Leftmost							
40.	Murik	12..89/1L	None	C	④	④	④
41.	Lithuanian	12..89/1L	None	D	④	④	④
42.	Amele	12..89/1L	None		④	⑤	⑤
43.	Mongolian Khalkha (per Street)	12..89/1L	H		④	⑤	⑤
44.	Yidin	12..89/1L	i2@m	B	⑤	×	⑤
45.	Kashmiri	12..78/ 12..78/1L	None		×	⑥	⑥
46.	Maori	12..89/ 12..89/1L	Not included		⑤	⑤	⑤
47.	Mongolian Khalkha (per Stuart)	12..89/2L	None		⑤	⑤	⑤
Leftmost Heavy otherwise Rightmost							
48.	Komi	12..89/9L	None		④	④	④
Rightmost Heavy otherwise Leftmost							
49.	Kuuku-Yau	12..89/9R	1L, H		⑤	⑤	⑤
50.	Nubian Dongolese	23..89/9R	H		⑤	⑤	⑤

Continued on next page

#	Name	Main	Secondary	Note	FL	BL	FBL
51.	Mongolian Khalkha (per Bosson)	23..891/9R	H		×	⑤	⑤
52.	Buriat	23..891/9R	1L, H		×	⑥	⑥
53.	Arabic Classical	1/23..89/ 9R	None		④	④	④
54.	Cheremis Eastern	23..89/9R	None		⑤	⑤	⑤
55.	Chuvash	12..89/9R	None		④	④	④
Rightmost Heavy otherwise Rightmost							
56.	Golin	12..89/1R	None		⑤	④	⑤
57.	Cheremis Meadow	1/23..891/ 1R	None		⑤	④	⑤
58.	Mam	12..89/12/ 2R	None	B	⑤	⑤	⑤
59.	Klamath	12..89/23/ 3R	if 3R=SH, 2R=H then 2R		×	×	⑥
60.	Seneca	see note	i2@m < m	E	⑦	⑦	⑦
61.	Cheremis Mountain	23..89/2R	None		⑥	⑤	⑥
62.	Hindi (per Jones)	23..891/2R	None		×	⑤	⑥
63.	Sindhi	23..891/2R	H		×	⑤	⑥
64.	Bhojpuri (per Shukla and Tiwari)	23..891/2R	’Hm’H, m’LL, 1L	Y	×	×	⑥
65.	Hindi (per Kelkar)	23..891/ 23..891/2R	H, i(’LL)@m < m, m <i(LL’)@m	X, Y	×	×	×

Table 5.4: Quantity-Sensitive Bounded Single, Dual, and Multiple Patterns

#	Name	Main	Secondary	Note	FL	BL	FBL
SINGLE							
66.	Maidu	12/2L	Not included		④	④	④
67.	Hopi	12/2L	None	B	④	④	④
68.	English verbs	12/2R	Not included		④	④	④
69.	Kawaiisu	12/2R	None	B	④	④	④
70.	Shoshone Tumpisa	21/1L	Not included		⑤	⑤	⑤
71.	Javanese	21/1R	None		⑤	⑤	⑤
72.	Manobo Sarangani (per Meiklejohn & Meiklejohn)	21/1R	None	B	⑤	⑤	⑤
73.	Awadhi	21/2R	None	B	⑤	⑤	⑤
74.	Malay (per Lewis)	23/3R (3+), 12/2L (2-)	None		×	⑤	⑤
75.	Latin Classical	23/3R (3+), 1L (2-)	None	B	⑤	⑤	⑤
76.	Hebrew Tiberian	12/21/1R	Not included		④	④	④
77.	English (nouns per Pater)	1@w3/234@sR	i('H,'LL)R		⑤	⑤	⑤
78.	Arabic Cairene	1@w3/23@sR	None	B	④	④	④
79.	Arabic Damascene	1@w3/23R	None		⑤	⑤	⑤
80.	Arabic Cyrenaican Bedouin	1@w3/23@sR (3+), 12/1R (2-)	i(H',LX')L (invs) (3+), None (2-)	B	×	×	×

Continued on next page

#	Name	Main	Secondary	Note	FL	BL	FBL
81.	Hindi (per Fairbanks)	12/2/34@sR (3+), 1L (2-)	i('H,'LL)R (invs) (3+), None (2-)	X	×	×	×
82.	Piraha	123/123/ 123/123/1R	None	X	×	×	×
Dual							
83.	Maithili	213/2R	1L	B	⑥	⑥	⑥
Multiple							
84.	Cambodian	1R	H	B, G	⑤	⑤	⑤
85.	Yapese	12/1R	H		④	④	④
86.	Tongan	12/2R	H	B	④	④	④
87.	Miwok Sierra	12/2L	H	B	④	④	④
88.	Gurkhali	12/1L	m < H		④	④	④

Table 5.5: Quantity-Sensitive Bounded Binary and Ternary Patterns

#	Name	Main	Secondary	Note	FL	BL	FBL
Binary							
89.	Aranda Western	12/2L (3+), 1L (2-)	i2@m, no 1R (3+), None (2-)		⑥	⑥	⑥
90.	Nyawaygi	12@sL	i('H,'LL)R		⑤	⑤	⑤
91.	Wargamay	12@sL	i('H,'LL)R, no 'H'L	B, I	⑥	⑥	⑥
92.	Romansh Berguener	12/2R	i('H,'LL)L		⑥	⑥	⑥
93.	Greek Ancient	12/2R	i('H,'LL)R		⑤	⑤	⑤
94.	Fijian	12/2R	i('H,'LL)R	B	⑤	⑤	⑤
95.	Romanian	12/2R	i2@m		⑤	⑤	⑤
96.	Seminole Creek	12@sR	i(H',LX')L	B	⑤	⑤	⑤

Continued on next page

#	Name	Main	Secondary	Note	FL	BL	FBL
97.	Aklan	21/1R	i('H,'LL)@m < m		⑤	⑤	⑤
98.	Malecite / Passamaquoddy	23@sR	i(H',LX')L		⑥	×	⑥
99.	Munsee	23@sR (3+), 12/2L (2-)	i(H',LX')L, no 1R (3+), None (2-)		×	⑥	⑥
100.	Cayuga	23@sR (3+), 1/0L (2-)	i(H',LX')L, no 1R (3+), None (2-)	B	⑥	⑥	⑥
101.	Manam	123/23/3R	i('H,'LL)@m < m		⑤	⑤	⑤
102.	Arabic Negev Bedouin	1@w3/23@sR (3+), 12/1R (2-)	i(H',LX')L (invs) (3+), None (2-)		×	×	×
103.	Arabic Bani-Hassan	1@w3/23@w2/2R	i('H,'LL)@m < m		⑤	⑤	⑤
104.	Arabic Palestinian	1/2/34@sR (3+), 1@w3/9R (2-)	i('H,'LL)L < m	B	×	×	×
105.	Asheninca	234/324@s/324@sR	i('H,'LL)L < m (w2=H)	B, X	×	×	×
106.	Dutch	1@w4/23@sR	i('H,'LL)R		⑤	⑤	⑤
Ternary							
107.	Estonian	1L	i('HX,'XLL,'LL)L		⑥	⑥	⑥
108.	Hungarian	1L	i('HX,'XLL,'LL)L, no 1R		⑥	⑥	⑥
109.	Sentani	12/2R	i('HX,'XLX)@mL		⑦	⑥	⑦

Table 5.6: Notes for Stress Patterns in Tables 5.1 - 5.5

ID	Note
A	no monosyllables
B	no light monosyllables
C	At most one heavy per word
D	At least one heavy per word
E	Rightmost even nonfinal syllable which is either heavy or followed by a (nonfinal) heavy. If no such syllables are present, none are stressed.
F	Pretonic heavies count as light
G	Light syllables occur only immediately following heavy syllables
H	w1,w2 = L, w3 = H
I	Heavy syllables only occur initially
X	Not tail canonically distinct
Y	Not head canonically distinct

CHAPTER 6

Deconstructing Neighborhood-distinctness

1 Overview

This chapter establishes a deeper understanding of the neighborhood-distinct languages. First it shows that precedence languages and trigram languages are tail canonically neighborhood-distinct, but that 4-gram languages are not even neighborhood-distinct. Second, it develops a strategy towards determining a language-theoretic characterization of the range of Forward Backward Neighborhood Learner which was introduced in Chapter 5.

There are three themes that run through the chapter: parameterizing the notion of locality, reverse determinism, and the compositional nature of the neighborhood. I elaborate on the relevance of these three themes here.

First, the notion of neighborhood can be generalized. In Chapter 5, the set of labels on the incoming and outgoing transitions made up part of the definition of the neighborhood. However, these are really the incoming and outgoing paths of length one. Thus we can distinguish various degrees of neighborhood-distinctness by varying the length of the incoming and outgoing paths.

Second, reverse deterministic acceptors are given a closer look. This follows from two observations. First, stress patterns with metrical units at the right word edge have a smaller head canonical acceptor (the smallest acceptor for a language which is reverse deterministic) than tail canonical acceptor. Secondly, the Forward

Backward Learner works in part by building a reverse deterministic representation of the input, the suffix tree. These facts suggest that reverse deterministic acceptors play a significant role, at least in right-edge based phonotactic patterns.

Third, the neighborhood is defined compositionally. In other words, the neighborhood is made up of four different components. By understanding the generalizations that are made over each of these component functions, we aim to understand the whole. Although the a complete understanding of the compositional nature of neighborhood-distinctness is still open, I present some interesting results that follow from this line of investigation, and make clear the remaining open questions.

2 Generalizing the Neighborhood

2.1 Preliminaries

For any acceptor $A = (Q, S, F, \delta)$ let $f : Q \rightarrow B$, where B is some set. Recall from Appendix A–1.3, that f induces an equivalence relation and partition over Q, which we denote $\sim_f$ and π_f, respectively. We call acceptors *f-distinct* iff A/π_f is isomorphic to A. A language L is *f-distinct* iff there exists a stripped acceptor A such that A is f-distinct and $L(A) = L$. We denote the class of languages that are f-distinct with $\mathcal{L}_{f-distinct}$.

Next we define a 'tupling' of equivalence relations.

Definition 15 Consider $f : Q \rightarrow B$ and $g : Q \rightarrow C$. Then $f \otimes g(q) : Q \rightarrow A \times B$, defined as follows:

$$f \otimes g(q) = (f(q), g(q))$$

We can now state our first theorem, which says that if a language is f-distinct, then it is also $f \otimes g$-distinct for any g.

Theorem 27 Let $f : Q \to B$. For any g, C, $g : Q \to C$, $\mathcal{L}_{f-distinct} \subseteq \mathcal{L}_{f \otimes g-distinct}$.

Proof: Consider any $L \in \mathcal{L}_{f-distinct}$. Since L is f-distinct, there is an acceptor $A = (Q, S, F, \delta)$, which accepts exactly L, such that A/π_f is isomorphic to A. It is sufficient to show that $A/\pi_{f \otimes g}$ is isomorphic to A. Consider any $p, q \in Q$ such that $p \neq q$. $f \otimes g(q) = (f(q), g(q))$ and $f \otimes g(p) = (f(p), g(p))$. Since A is f-distinct, $f(q) \neq f(p)$ and therefore $(f(q), g(q)) \neq (f(p), g(p))$. Thus every block in $A/\pi_{f \otimes g}$ is trivial, proving the theorem. □

The next theorem establishes another case where f-distinctness guarantees g-distinctness.

Theorem 28 For any acceptor $A = (Q, S, F, \delta)$, let $f, g : Q \to B$ such that for all $q \in Q$, $f(q) \subseteq g(q)$. Then $\mathcal{L}_{f-distinct} \subseteq \mathcal{L}_{g-distinct}$.

Proof: Assume A is f-distinct so A/π_f is isomorphic to A. It is sufficient to show that A/π_g is isomorphic to A. Consider any $p, q \in Q$ such that $p \neq q$. Note that by assumption $f(q) \subseteq g(q)$ and $f(p) \subseteq g(p)$. Since A is f-distinct, $f(q) \neq f(p)$, and thus $g(q) \neq g(p)$. Thus every block in A/π_g is trivial, proving the theorem. □

Lemma 23 If $A = (Q, S, F, \delta)$ is $f \otimes g$ distinct, then for all $p, q \in Q$, $p \neq q$, either $f(q) \neq f(p)$ or $g(q) \neq g(p)$.

Proof: This follows immediately from the definition of $\otimes$. □

2.2 Neighborhood-distinctness

Now we can generalize the concept of a neighborhood. Recall the definitions of $I_n(q) : Q \to \Sigma^{\leq n}$ and $O_n(q) : Q \to \Sigma^{\leq n}$ repeated from Equations 3.1 and 3.2, respectively.

$$I_n(q) = \{w \in \Sigma^{\leq n} : \exists p \in Q \text{ such that } w \text{ transforms } p \text{ to } q\} \qquad (6.1)$$

$$O_n(q) = \{w \in \Sigma^{\leq n} : \exists p \in Q \text{ such that } w \text{ transforms } q \text{ to } p\} \tag{6.2}$$

We now also define functions $final$ and $start$ whose codomain is $\{0, 1\}$.

$$final(q) = 1 \text{ iff } q \in F, \text{ otherwise } 0 \tag{6.3}$$

$$start(q) = 1 \text{ iff } q \in I, \text{ otherwise } 0 \tag{6.4}$$

Each of these functions naturally induce equivalence relations over Q and hence partitions over Q. For the functions $I_n, O_n, final, start$, denote the equivalence relations they induce $\sim_{I_n}, \sim_{O_n}, \sim_{final}, \sim_{start}$, respectively. Likewise, denote the partitions induced with $\pi_{I_n}, \pi_{O_n}, \pi_{final}, \pi_{start}$, respectively.

Next we define the *j-k neighborhood* function, which we denote $nh_{j,k}$, as the 'tupling' of Equations 6.1 - 6.4.

$$nh_{j,k} = I_j \otimes O_k \otimes final \otimes start$$

When j and k are understood from context we just write

$$nh(q) = (I(q), O(q), final(q), start(q))$$

We denote the equivalence relation and partition induced by nh with $\sim_{nh}$ and π_{nh}, respectively.

Lemma 24 Let $A = (Q, S, F, \delta)$. Then for all $q \in Q$, $I_n(q) \subseteq I_{n+1}(q)$.

Proof: Consider any w in $I_n(q)$. By definition, there is $p \in Q$ such that w transforms p to q and $|w| < n$. It follows that $|w| < n+1$ and so by definition, $w \in I_{n+1}$. □

Lemma 25 Let $A = (Q, S, F, \delta)$. Then for all $q \in Q$, $O_n(q) \subseteq O_{n+1}(q)$.

Proof: As above. □

Now we prove that any language which is j-k-neighborhood-distinct is also $j+1$-k-neighborhood-distinct and j-$k+1$-neighborhood-distinct. The converse, however, is false.

Theorem 29 $\mathcal{L}_{nh_{j,k}-distinct} \subset \mathcal{L}_{nh_{j+1,k}-distinct}$ and $\mathcal{L}_{nh_{j,k}-distinct} \subset \mathcal{L}_{nh_{j,k+1}-distinct}$.

Proof: Lemma 24 and Lemma 25 allow us to apply Theorem 28 and conclude that if L is I_n-distinct (O_n-distinct), then it is also I_{n+1}-distinct (O_{n+1}-distinct). Then it follows directly from Theorem 27 that $\mathcal{L}_{nh_{j,k}-distinct} \subseteq \mathcal{L}_{nh_{j+1,k}-distinct}$ and $\mathcal{L}_{nh_{j,k}-distinct} \subseteq \mathcal{L}_{nh_{j,k+1}-distinct}$. Finally, it is easy to see that there is some language L in $\mathcal{L}_{nh_{j+1,k}-distinct}(\mathcal{L}_{nh_{j,k+1}-distinct})$ which is not in $\mathcal{L}_{nh_{j,k}-distinct}$ because any acceptor A for L is I_{n+1}-distinct (O_{n+1}-distinct) but not I_n-distinct (O_n-distinct) (cf. the nested family of n-gram languages in Theorem 13 in Appendix C–2.3). $\square$

3 Subsets of Neighborhood-distinct Languages

Here we prove two results: that trigram languages (discussed in Chapter 3) and precedence languages are tail canonically 1-1 neighborhood-distinct.

3.1 Precedence Languages

The precedence languages are tail canonically 1-1 neighborhood-distinct, but the converse is false.

Theorem 30 $\mathcal{L}_{prec} \subset \mathcal{L}_{nh_{0,1}-distinct}$.

Proof: Consider any precedence language L. By Theorem 25, the tail canonical acceptor A for L is 0_1-distinct. Therefore, by Theorem 27, A is tail canonically $nh_{0,1}$-distinct and $L \in \mathcal{L}_{nh_{0,1}-distinct}$.

Now consider $L = \{aa\}$. The canonical acceptor for this language is 0-1-neighborhood distinct because it consists of three states, one is a nonfinal start state, one is a nonstart final state, and one is neither final nor start. Therefore $L \in \mathcal{L}_{nh_{0,1}-distinct}$. However, $L \notin \mathcal{L}_{prec}$ because the smallest precedence language which includes aa also includes aaa. □

3.2 N-gram Languages

Theorem 31 establishes that any n-gram language is also tail canonically j-k neighborhood-distinct, provided that $j+k \geq n-1$. Consequently, it follows that trigram languages are 1-1 neighborhood-distinct.

Theorem 31 For all $j, k, n \in \mathbb{N}$ such that $j + k = n - 1$, $\mathcal{L}_{n-gram} \subset \mathcal{L}_{nh_{j,k}-distinct}$.

Proof: Consider any $L \in \mathcal{L}_{n-gram}$ and the acceptor $A = (Q, I, F, \delta)$ for L constructed according to Theorem 16. Consider any two states $p', q' \in Q$ $(p' \neq q')$ such that $nh_{j,k}(p') = nh_{j,k}(q')$. Since $nh(p') = nh(q')$, in particular because $I_j(p') = I_j(q')$, there exists $p, q \in Q$ and $u \in \Sigma^{\leq j}$ such that u transforms p to p' and u transforms q to q'.

Now consider any $v \in \Sigma^{\leq k}$ such that $v \in O_k(p') = O_k(q')$. Let p'' denote the state to which v transforms p' and q'' the state to which v transforms q'. (We know there is exactly one state since A is forward deterministic.) However, since $|uv| = n - 1$ and each state in A is uniquely identified by suffixes (of prefixes of L) of length $n - 1$ (or less) by Theorem 18 and Corollary 10, it must be the case that $p'' = q''$. Since v was arbitrary in $O_k(p') = O_k(q')$, any string w which transforms p' to some final state q_f also transforms q' to q_f. In other words, the tails of p' are the same as the tails of q'. Therefore merging states p' and q' does not change the language L. Thus we have shown that A/π_{nh} accepts exactly L and therefore $L \in \mathcal{L}_{nh_{j,k}-distinct}$.

Next we show that the subset relation is proper. Consider a language $L = \{a^n, b^{n-1}a, ba^{n-1}\}$, where $a, b \in \Sigma$. Note that a canonical acceptor for L, a schematic of which is shown in Figure 6.1, is j-k neighborhood-distinct for any $j, k \in \mathbb{N}$ such that $j + k = n - 1$.

We show that the smallest n-gram language containing L is not equal to L. Note that $\{\#a^{n-1}, a^{n-1}\#\}$ is contained in $\gamma_{(n)gram}(L)$. Consequently, $L(\gamma_{(n)gram}(L))$, which is the smallest n-gram language containing L, includes a^{n-1} which is not an element of L. Therefore $L \notin \mathcal{L}_{n-gram}$. □

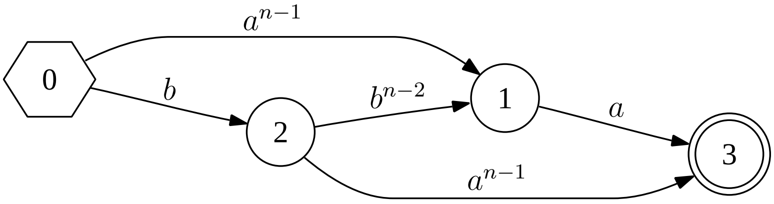

Figure 6.1: A Schema for an Acceptor for $L = \{a^n, b^{n-1}a, ba^{n-1}\}$

It follows from Theorem 31 that $\mathcal{L}_{3gram} \subset \mathcal{L}_{nh_{1,1}-distinct}$ since $1 + 1 = 3 - 1$.

The next theorem establishes, for example that four-gram languages are not comparable with 1-1 neighborhood-distinct languages.

Theorem 32 For all nonzero $j, k, n \in \mathbb{N}$ such that $j + k = n - 2$, $\mathcal{L}_{n-gram}$ is incomparable with $\mathcal{L}_{nh_{j,k}-distinct}$.

Proof: Consider $L = (a^{n-1}b)^*$. It is easy to verify that $L \in \mathcal{L}_{n-gram}$. However L does not belong to $\mathcal{L}_{nh_{j,k}-distinct}$ when $j + k = n - 2$. To show this, first consider $w = a^{n-1}ba^{n-1}b \in L$. Clearly, $u = a^{n-1}ba^j \in Pr(L)$ and $ua = a^{n-1}ba^ja \in Pr(L)$. Thus for any stripped acceptor $A = (Q, S, F, \delta)$ for L, there are states q_u and q_{ua} such that $\delta(I, u) = q_u$, and $\delta(I, ua) = q_{ua}$.

We show that q_u and q_{ua} have the same j-k neighborhood. States q_u and q_{ua} are both nonfinal states. If they were final states, then A would not be an acceptor for L since $u, ua \notin L$. By similar reasoning, since $j \neq 0$, both q_u and q_{ua} are not start states. $I_j(q_u) = \{a\}^{\leq j}$ since $u = a^{n-1}ba^j$. No strings with b belong to $I_j(p)$ because if they did then A would not be an acceptor for L. $I_j(q_{ua})$ also equals $\{a\}^{\leq j}$ since $ua = a^{n-1}ba^ja$. Similarly, $O_k(p) = \{a\}^{\leq k} = O_k(q)$. Therefore p and q have the same j-k neighborhood and A is not j-k neighborhood-distinct. Since A was arbitrary, L is not j-k neighborhood-distinct.

On the other hand, it is easy to establish that there are j-k distinct languages not recognizable by n-gram grammars. Consider $L = a^{2n}$. The canonical acceptor is j-k neighborhood-distinct, but no language belonging to $\mathcal{L}_{n-gram}$ equals L. □

3.3 Local Summary

The diagram in Figure 6.2 summarizes the known proper subset relationships that exist between the various languages classes for small values of j and k. It remains an open question what the relationship is between the j_1-k_1-neighborhood distinct class of languages and the j_2-k_2-neighborhood distinct class of languages when the $j_1 > j_2$ but $k_1 > k_2$ and vice versa.

We can also ask if precedence languages and trigram languages are in the range of the Forward Backward Neighborhood Learner. Simulations suggest that they are. Thus I conjecture that precedence and trigram languages belong to that part of the 1-1-neighborhood-distinct languages which can be learned by the Forward Backward Learner.

It is interesting to note that because the range of the Forward Backward Learner is a much larger hypothesis space than the range of the precedence and trigram languages, the Forward Backward Learner requires a much larger sufficient sample than the precedence or n-gram learners in order to converge to the correct language. Furthermore, it appears that the size of the sufficient sample is prohibitively large,

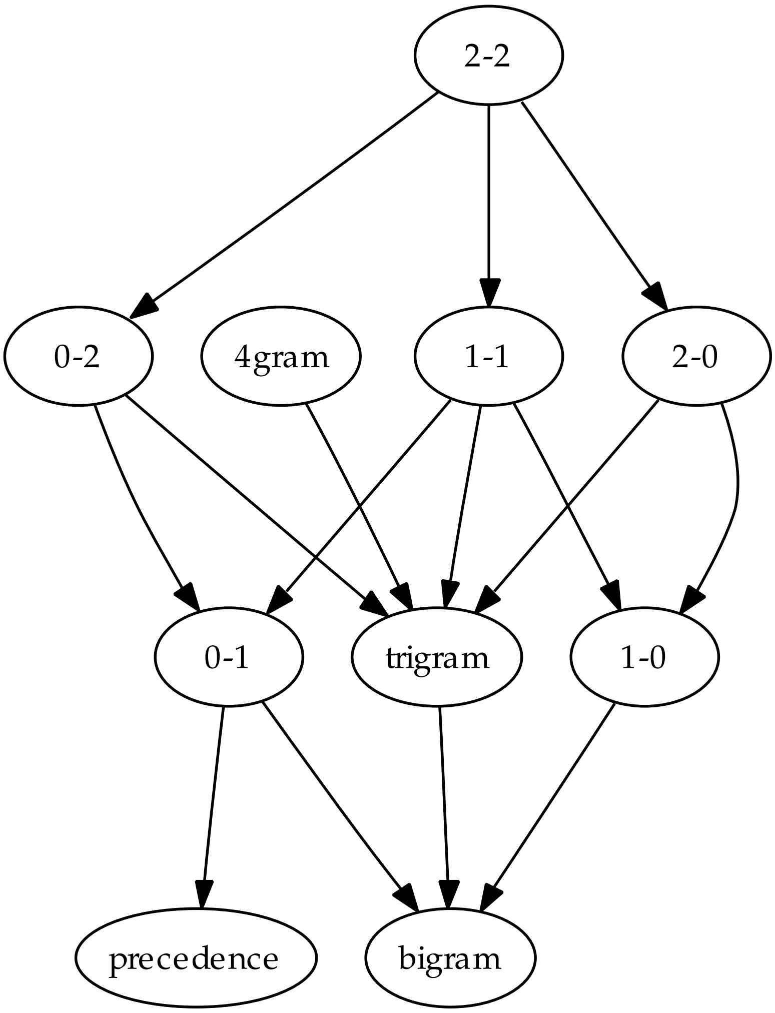

Figure 6.2: Subset Relations among Language Classes

in the sense that the kinds of strings which are required to be in the sample are not ones likely to be found in natural language. For example, when the Forward Backward Learner learns a precedence language, simulations suggest that words which contain long strings of contiguous consonants or vowels must be in the sample. Such words are not typically found in natural language because they are ruled out by constraints on syllable structure. Such complex samples are not needed by the precedence learner (see Chapter 4 §3.2).

4 Neighborhood-distinctness not Preserved Under Intersection

The following theorem proves that the 1-1 neighborhood-distinct languages are not closed under intersection.

Theorem 33 1-1 neighborhood-distinct languages are not closed under intersection.

Proof: Let $L_1 = \{a, aaa\}$ and $L_2 = \{aa, aaa\}$. Let $L_3 = L_1 \cap L_2 = \{aaa\}$. We show that L_1 and L_2 are 1-1 neighborhood-distinct, but L_3 is not. Figure 6.3 shows neighborhood-distinct acceptors for L_1 and L_2.

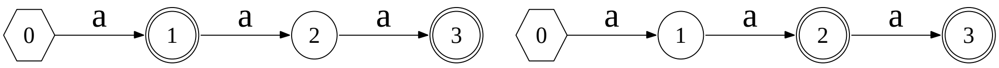

Figure 6.3: Acceptors for L_1 and L_2, respectively.

To prove that L_3 is not 1-1 neighborhood-distinct, consider any stripped acceptor $A = (Q, I, F, \delta)$ for L_3. Since A accepts aaa and is stripped, there are states p and q such that $\delta(I, a) = p$, $\delta(p, a) = q$, and $\delta(q, a) \in F$. We show that $nh(p) = nh(q)$. If either p or q were final (or start) states, then A would accept a language not equal to L_3. Therefore p and q are neither final nor start states. Finally, $\{\lambda, a\}$ is a subset of each $I_1(q)$, $I_1(p)$, $O_1(p)$, $O_1(q)$ because of the elements of δ identified above. Furthermore, there are no other elements in any of $I_1(q)$, $I_1(p)$, $O_1(p)$, $O_1(q)$ because A is stripped and elements of these sets must be of length less than or equal to 1. □

Corollary 16 The j-k-neighborhood distinct languages are not closed under intersection when both j and k are nonzero.

Proof: This follows as an immediate consequence of Theorem 29. □

It remains an open question whether the j-k-neighborhood distinct languages are closed under intersection when either j or k is zero.

This (negative) result is significant because it means intersecting neighborhood-distinct grammars does not guarantee that the resulting language is in the class. For example, the range of the Forward Backward Learner, which intersects two 1-1 neighborhood-distinct machines, may return a language which is not 1-1 neighborhood-distinct.

Thus the hypothesis that all phonotactic patterns are neighborhood distinct can be understood in two different ways. If we conceive of the whole phonotactic grammar as the intersection of its components, there is a question as to whether the hypothesis applies at the level of the whole grammar, or at the level of the components (an what precisely constitutes a component). One relevant fact here is that intersection of two languages which are not 1-1 neighborhood-distinct can yield languages which are 1-1 neighborhood-distinct. For example, it is easy to verify that $L_4 = \{a^2, a^3, a^8\}$ is not 1-1 neighborhood-distinct and neither is $L_5 = \{a^2, a^3, a^9\}$, but their intersection is.

5 Towards a Compositional Analysis of Neighborhood-distinctness

5.1 Strategy

The goal of this section is to increase our understanding of neighborhood-distinct languages by first understanding the range of the Forward Backward Learning function. The main reason for this is that the neighborhood-distinct languages include languages which are only recognized by non-deterministic neighborhood-distinct acceptors. Because non-determinism makes analysis difficult, we focus our attention on the range of the Forward Backward Learning function.

Since the neighborhood function is composed of four separate functions given by Equations 6.1 - 6.4 above, the strategy employed here is to understand the language classes induced by state merging prefix and suffix trees according to the partitions induced by the individual functions. In other words, what are language-theoretic characterizations of $\mathcal{L}_{I_n-distinct}$, $\mathcal{L}_{O_n-distinct}$, $\mathcal{L}_{final-distinct}$, and $\mathcal{L}_{start-distinct}$? However, even this question is dogged by non-determinism. Therefore, here we ask only what classes of languages are recognized by forward (and reverse) deterministic acceptors which are I_n-distinct, O_n-distinct, $final$-distinct, and $start$-distinct.

Once we know the language theoretic characterizations of the language classes learnable by merging states according to the functions individually, it may be possible to determine a language theoretic characterization of the range of the Forward Backward Learning function. This is because the set of possible partitions of some finite set (in this case the states of finite state machine) form a lattice (Grätzer 1979). The notions of least upper bound and greatest lower bound provide one way to navigate among the elements of the lattice. For example, suppose functions f and g induce equivalence relations and partitions π_f and π_g of some finite set Q. Suppose we are interested in a partition which refines both π_f and π_g. A natural choice is the coarsest partition which refines both; this partition equals the greatest lower bound of π_f and π_g. Similarly, the finest partition which coarsens both π_f and π_g is the least upper bound.

Consequently, the partition obtained by equating states with the same neighborhood is thus the greatest lower bound obtained of the partitions obtained by the equivalence relations $\sim_{I_n}$, $\sim_{O_n}$, $\sim_{final}$, and $\sim_{start}$. Also, when we consider the functions *start* and $final$, these are also decomposable. For example, the partition induced by $final$ is really the least upper bound of two partitions induced by two different equivalence relations. The first equivalence relations equates two states p, q in an acceptor $A = (Q, I, F, \delta)$ iff $p, q \in F$. The second relates two states p, q iff $p, q \notin F$. Likewise, the partition induced by the function *start* is the least upper bound of two equivalence relations denoted $p, q \in I$ and $p, q \notin I$.

The remainder of this section explores the languages obtainable by merging states in a learning algorithm while manipulating two variables: the initial representation of the input (prefix/suffix tree distinction), and the equivalence relation used to partition the states of that structure. Although there are still some open questions, the results reveal some interesting language classes that are plausibly relevant to phonotactic learning, help us understand the neighborhood, and which point to an algebraic structure underlying the problem.

5.2 Merging States in Prefix Trees with Same Incoming Paths of Length n

In this section we ask which class of languages is obtained by merging states in the π_{I_n} partition of a prefix tree. Theorem 18 in Chapter 3 Appendix C–4.3 establishes that this class of languages is $\mathcal{L}_{n-gram}$.

5.3 Merging States in Suffix Trees with Same Outgoing Paths of Length n

In this section we ask which class of languages is obtained by merging states in the π_{O_n} partition of a suffix tree built for any sample S. Notice each state in $(ST(S)/\pi_{O_n})^r$ can be identified uniquely by its incoming paths of length n. In other words, $(ST(S)/\pi_{O_n})^r$ is equivalent to some n-gram grammar and $L((ST(S)/\pi_{O_n})^r) \in \mathcal{L}_{n-gram}$. Since $\mathcal{L}_{n-gram}$ is closed under reversal (Theorem 12 in Appendix C–2.3), it follows that $L(ST(S)/\pi_{O_n})$ belongs to $\mathcal{L}_{n-gram}$ too. Thus I have sketched part of a proof that the set of languages obtainable by merging states in suffix trees with same outgoing paths of length n is $\mathcal{L}_{n-gram}$.

5.4 Merging States in Prefix Trees with Same Outgoing Paths of Length n

The question which class of languages is obtained by merging states in the π_{O_n} partition of a prefix tree remains open.

5.5 Merging States in Suffix Trees with Same Incoming Paths of Length n

The question which class of languages is obtained by merging states in the π_{I_n} partition of a suffix tree also remains open. It seems obvious that the solution to this open question will make the solution to the other one above immediate (and vice versa).

5.6 Merging Final States in Prefix Trees

In this section we ask which class of languages is obtained by merging final states of a prefix tree. Below I show that languages obtained in this way have the property that if u, uv belong to the language, then uv^* belongs to the languages. For phonologists, this result is interesting because this pattern can be understood as left-to-right iterativity. For formal learning theorists, this result is interesting because, unlike $\mathcal{L}_{n-gram}$, $\mathcal{L}_{prec}$, and $\mathcal{L}_{nd-distinct}$, this class of languages obtained is not finite. Additionally, it is nontrivially related to the zero-reversible languages (Angluin 1982) (though incomparable with them).

5.6.1 Definitions

An acceptor $A = (Q, I, F, \delta)$ is *1-final* iff $|F| \leq 1$. A language L is 1-final iff there exists a 1-final acceptor for L. Since every regular language has a head-canonical acceptor which by definition has at most 1 final state, the class of 1-final languages is equivalent to the regular languages.

A more interesting class of languages are those where there is a deterministic acceptor which has at most 1 final state. We call the class of languages accepted by such acceptors *1-final-deterministic* and denote this class with $\mathcal{L}_{1fd}$. We now give a language theoretic-characterization of these *1-final-deterministic* languages.

Theorem 34 Let L be a regular language. Then L is 1-final-deterministic iff whenever u, v are in L the $T_L(u) = T_L(v)$.

Proof: Suppose there exists a 1-final deterministic acceptor $A = (Q, I, F, \delta)$ for L. If $|F| = 0$ then the L is the empty set which vacuously satisfies the statement so consider the case when $|F| = 1$. Then for any strings u and v which are accepted by A, let $q = \delta(I, u)$ and $p = \delta(I, v)$. Since $|F| = 1$ and since A accepts u and v, it must be the case that $p = q$. Therefore, by Corollary 5, $T_L(u) = T_L(v)$.

Now suppose L is such that whenever $u, v \in L$, $T_L(u) = T_L(v)$. The canonical acceptor for L is 1-final by definition. This is because the set F for a canonical acceptor is defined as $\{T_L(w) : w \in L\}$. If L is the empty language, $|F| = 0$, otherwise $F = \{T_L(u)\}$, a singleton set. □

Corollary 17 Let $L \in \mathcal{L}_{1fd}$ and let $x, y_1, y_2, \ldots y_n \in \Sigma^*$ such that $x, xy_1, xy_2, \ldots xy_k \in L$ for some $k \in \mathbb{N}$. Then $x(y_1 + y_2 + \ldots + y_k)^* \subseteq L$.

Proof: For some $k \in \mathbb{N}$, let $x, xy_1, xy_2, \ldots xy_k \in L$. We show, by induction, that for any $n \in \mathbb{N}$, $x(y_1+y_2+\ldots+y_k)^n \subseteq L$. Clearly when $n = 0$, $x \in L$. So now let us assume that for some $n \in \mathbb{N}$, if $x, xy_1, xy_2, \ldots xy_k \in L$ then $x(y_1+y_2+\ldots+y_k)^n \subseteq L$. It remains to be shown that for all $1 \leq i \leq k$, $x(y_1 + y_2 + \ldots + y_k)^n y_i \subseteq L$. Since $x, xy_i \in L$ and $L \in \mathcal{L}_{1fd}$, by Theorem 34, for all $w \in L$, $y_i \in T_L(w)$. By the inductive hypothesis, for all $w \in x(y_1 + y_2 + \ldots + y_k)^n$, $w \in L$ and so therefore $wy_i \in L$. Since i, w are arbitrary it is the case that $x(y_1 + y_2 + \ldots + y_n)^{n+1} \subseteq L$. This completes the induction and the proof. □

The 1-final deterministic languages are not identifiable in the limit. A limit point proof (Osherson et al. 1986: §2.2) establishes this claim, which I sketch here. Since $\{abc\}, \{abc, abbc\}, \{abc, abbc, abbbc\}, \ldots ab^*c$ are all 1-final-deterministic languages, no learner will succeed for this subset (and hence the 1-final-deterministic languages.

5.6.2 1-final-cyclic-deterministic-languages

Here we introduce a proper subset of the 1-final-deterministic languages.[1] An acceptor $A = (Q, I, F, \delta)$ is 1-final-cyclic-deterministic iff it is 1-final-deterministic

[1] It is also of interest to point out that zero reversible languages (Angluin 1982) are also a (proper) subset of 1-final deterministic languages.

and the following property is true of A:

$$F \subseteq \bigcap \{range(\vec{q}) : \vec{q} \in loops_Q(q) \text{ for all } q \text{ cyclic in } A\}$$

(The definition for the range $range(x)$ is given in §A–1.5 and the $loops_Q(q)$ is defined in §B–3.9.) Thus 1-final-cyclic-deterministic acceptors are those in which all loops must pass through the final state (if there is one). Examples are given below.

1-final-cyclic-deterministic languages are those which can be accepted by 1-final-cyclic-deterministic acceptors. We denote these languages with $\mathcal{L}_{1fcd}$.

Example 14 The language accepted by the acceptor in Figure 6.4 belongs to $\mathcal{L}_{1fcd}$. The following table illustrates the computation which determines that the language of the acceptor in Figure 6.4 belongs to $\mathcal{L}_{1fcd}$.

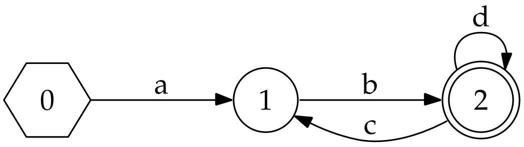

Figure 6.4: The FSA for a 1-Final-Cyclic-Deterministic Language.

(1)

$loops_Q(1)$	$=$	$\{121\}$	$R(121)$	$=$	$\{1,2\}$
$loops_Q(2)$	$=$	$\{212, 22\}$	$R(212)$	$=$	$\{1,2\}$
			$R(22)$	$=$	$\{2\}$
			$\cap$	$=$	$\{2\}$

Example 15 The language recognized by the acceptor in Figure 6.5 does not belong to $\mathcal{L}_{1fcd}$. The following table illustrates the computation which determines that the language of the acceptor in Figure 6.4 does not belong to $\mathcal{L}_{1fcd}$.

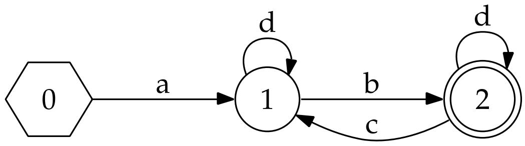

Figure 6.5: The FSA for a 1-Final-Cyclic-Deterministic Language.

(2)

$loops_Q(1) = \{121, 11\}$	$R(121) = \{1, 2\}$
	$R(11) = \{1\}$
$loops_Q(2) = \{121, 22\}$	$R(121) = \{1, 2\}$
	$R(22) = \{2\}$
	$\cap = \emptyset$

Since the final state is not in the intersection this machine is not 1-final-cyclic deterministic.

From the above examples, we see that 1-final-cyclic-deterministic automata are defined so that the only loops pass through the final state.

Lemma 26 Let $L = L_1 \cdot L_2{}^*$ where $L_1, L_2 \in \mathcal{L}_{fin}$. For all $w \in L$, there exists $x, y_1, y_2, \ldots y_n \in \Sigma^*$ $(n \in \mathbb{N})$ such that $w = xy_1y_2 \ldots y_n$ where $x \in L_1$ and for all $1 \leq i \leq n$, $y_i \in L_2$.

Proof: Omitted. □

Theorem 35 $\mathcal{L}_{1fcd} = \mathcal{L}_{1fd} \cap \mathcal{L}_{fin} \cdot (\mathcal{L}_{fin})^*$.

Proof: I only sketch a proof here. Consider any $L \in \mathcal{L}_{1fcd}$. $L \in \mathcal{L}_{1fd}$ by definition of $\mathcal{L}_{1fcd}$. To show that $L \in \mathcal{L}_{fin} \cdot (\mathcal{L}_{fin})^*$, it is sufficient to show that there exists

$L_1, L_2 \in \mathcal{L}_{fin}$ such that $L = L_1 \cdot L_2^*$. It is possible to show that L_1 is the language of the largest stripped acyclic subacceptor of A (see §B–3.9) and that L_2 is equal to the string loops of the final state ($loops_\Sigma(q_f)$) (if there is one).

Now consider any $L \in \mathcal{L}_{1fd} \cap \mathcal{L}_{fin} \cdot (\mathcal{L}_{fin})^*$. By definition of L, we know there is an acceptor $A = (Q, I, F, \delta)$ which is 1-final and deterministic. It remains to be shown that $F \subseteq \bigcap\{R(\vec{q}) : \vec{q} \in loops_Q(q)$ for all q cyclic in $A\}$. This is possible because the only loops that are formed by merging states necessarily include the states which are merged (see Appendix C–3) and since only final states are merged, this follows necessarily. □

5.6.3 Learning 1-final-cyclic-deterministic languages

Theorem 36 For any 1-final-cyclic-deterministic language L, there exists a characteristic sample.

Proof: Let L be a 1-final-cyclic-deterministic language. From Theorem 35, we know there exist $L_1, L_2 \in \mathcal{L}_{fin}$ such that $L = L_1 \cdot L_2^*$. Let the sample $S_0 = L_1 \cup L_1 \cdot L_2$.

Let L' be any 1-final-cyclic-deterministic language containing S_0. Consider any $w \in L$. It is sufficient to show that $w \in L'$. Since $w \in L$ and $L \in \mathcal{L}_{1fcd}$, there is some $n \in \mathbb{N}$ such that $w = xy_1y_2 \ldots y_n$ where $x \in L_1$ and for any $1 \leq i \leq n$, $y_i \in L_2$. It is also the case that $x, xy_i \in S_0$. Since $S_0 \subseteq L'$, $x(y_1 + y_2 + \ldots + y_n)^* \subseteq L'$ by Corollary 17. Clearly $w \in x(y_1 + y_2 + \ldots + y_n)^*$ and thus in L', completing the proof. □

Since it is established that there is a characteristic sample S for any 1-final-cyclic-deterministic language L, we know that any learner which guesses L after exposure to S has picked the smallest 1-final-cyclic-deterministic language consistent with S.

The algorithm given below is identical to the ZR algorithm by Angluin (1982) except that states which share the same b-predecessors are not merged. In other words, apart from requiring final states to be merged, reverse determinism is not enforced. The procedure s-UPDATE ensures that the any (forward) non-determinism that occurs by merging final states is removed by the time the algorithm terminates. It removes the nondeterminism by merging the same b-successors of some nondeterministic state (because of multiple departing b-transitions). It is easy to see that these merges do not change the language accepted by the pre- and post- merged acceptors. Formally, S-UPDATE(B_1, B_2, b) places $(s(B_1, b), s(B_2, b))$ on LIST if both $s(B_1, b)$ and $s(B_2, b)$ are nonempty and defines $s(B_3, b)$ to be $s(B_1, b)$ if this is nonempty and $s(B_2, b)$ otherwise (where B_3 is the union of B_1 and B_2) (cf. ZR algorithm by Angluin (1982).)

Note the algorithm defined above is the same as the algorithm ZR in Angluin (1982), the only difference being that there is no procedure for updating LIST when two blocks share the same b-predecessors. Because it does strictly less than ZR, which is tractable, 1FCD is also tractable.

It is possible to now prove that for any sample S, the output of Algorithm 7 is the smallest language in $\mathcal{L}_{1fcd}$ which contains S.

Theorem 37 Let S be any nonempty positive sample. Then the output of Algorithm 7 is the smallest 1-final-cyclic-deterministic language containing S.

Proof: Omitted. □

Now it is possible to prove that $\mathcal{L}_{1fcd}$ is identifiable in the limit.

Theorem 38 Let L be a 1-final-cyclic-deterministic language, $w_1, w_2, \ldots$ a positive representation of L. For any $i \in \mathbb{N}$ let $S_i = \{w_n : n \leq i\}$ and let $L(PT(S_1)), L(PT(S_2)), \ldots$ be a sequence of languages. This sequence converges to L.

Algorithm 7 The 1-Final-Cyclic-Deterministic (1FCD) Algorithm

```
Input: a nonempty positive sample $S$.
Ouput: a 1-final-cyclic-deterministic acceptor $A$.
Initialization
Let $A_0 = (Q_0, I_0, F_0, \delta_0) = PT(S)$.
Let $\pi_0$ be the trivial partition of $Q_0$.
For each $b \in \Sigma$ and $q \in Q_0$, let $s(\{q\}, b) = \delta_0(q, b)$.
Choose some $q'$ in $F_0$.
Let LIST contain all pairs $(q, q')$ such that $q \in F_0 - \{q'\}$.
Let $i = 0$.
Merging
while LIST $\neq \emptyset$ do
  Remove some element $(q_1, q_2)$ from LIST.
  Let $B_1 = B(q_1, \pi_i)$ and $B_2 = B(q_2, \pi_i)$
  if $B_1 \neq B_2$ then
    Let $\pi_{i+1}$ be $\pi_i$ with $B_1$ and $B_2$ merged.
    for all $b \in \Sigma$ do
      $s$-UPDATE$(B_1, B_2, b)$
    end for
    Increase $i$ by 1.
  end if
end while
Termination
Let $f = i$ and output the acceptor $A_0/\pi_f$.
```

Proof: This theorem follows from Theorem 36 and Theorem 37. By Theorem 36, L contains a characteristic sample. Let N be sufficiently large that S_N contains a characteristic sample for L. For $n \geq N$, the output of Algorithm 7 is the smallest 1-cyclic-deterministic language containing S_n by Theorem 37. □

5.7 Merging Start States in Suffix Trees

Merging start states in suffix trees is akin to merging final states in prefix trees. The languages obtained in this way have the property that if v, uv belong to the language, then u^*v belongs to the language as well. Consequently, this class of languages can be understood as those patterns which exhibit right-to-left iterativity. Because development of this section follows closely the one in §5.6, I omit many of the details.

An acceptor $A = (Q, I, F, \delta)$ is *1-start* iff $|I| \leq 1$. A language L is 1-start iff there exists a 1-start acceptor for L. Since every regular language has a tail-canonical acceptor which by definition has at most 1 start state, the class of 1-start languages is equivalent to the regular languages.

A more interesting class of languages are those where there is a reverse deterministic acceptor which has at most 1 start state. We call the class of languages accepted by such acceptors *1-start-reverse-deterministic* and denote this class with $\mathcal{L}_{1srd}$. We now give a language theoretic-characterization of these *1-start-reverse-deterministic* languages.

Theorem 39 Let L be a regular language. Then $L \in \mathcal{L}_{1srd}$ iff whenever u, v are in L the $H_L(u) = H_L(v)$.

Proof: Suppose there exists a 1-start reverse deterministic acceptor $A = (Q, I, F, \delta)$ for L. If $|I| = 0$ then the L is the empty set which vacuously satisfies the statement so consider the case when $|I| = 1$. Then for any strings u and v which are accepted by A^r, let $q = \delta^r(F, u^r)$ and $p = \delta^r(F, v^r)$. Since $|I| = 1$ and since A^r accepts u and v, it must be the case that $p = q$. Therefore, by Corollary 9, $H_L(u) = H_L(v)$.

Now suppose L is such that whenever $u, v \in L$, $H_L(u) = H_L(v)$. The head canonical acceptor for L is 1-start by definition. This is because the set I for a head canonical acceptor is defined as $\{H_L(w) : w \in L\}$. If L is the empty language, $|I| = 0$, otherwise $I = \{T_L(u)\}$, a singleton set. □

Like the 1-final deterministic languages, the 1-start reverse deterministic languages are not identifiable in the limit, which can be shown with a limit point proof (Osherson et al. 1986: §2.2).

Now we introduce a proper subset of the 1-start-deterministic languages.An acceptor $A = (Q, I, F, \delta)$ is 1-start-cyclic-reverse-deterministic iff it is 1-start-reverse-deterministic and the following property is true of A:

$$I \subseteq \bigcap\{range(\vec{q}) : \vec{q} \in loops_Q(q) \text{ for all } q \text{ cyclic in } A\}$$

Thus 1-start-cyclic-deterministic acceptors are those in which all loops must pass through the start state (if there is one).

It is now possible to show, though I omit the proof, what $\mathcal{L}_{1scrd}$ equals.

Theorem 40 $\mathcal{L}_{1scrd} = \mathcal{L}_{1srd} \cap (\mathcal{L}_{fin})^* \cdot \mathcal{L}_{fin}$.

Proof: Omitted □

Every language in this class of languages has a characteristic sample.

Theorem 41 For any 1-start-cyclic-deterministic language L, there exists a characteristic sample.

Proof: Let L be a 1-start-cyclic-deterministic language. From Theorem 35, we know there exist $L_1, L_2 \in \mathcal{L}_{fin}$ such that $L = L_1^* \cdot L_2$. Let the sample $S_0 = L_1 \cup L_1 \cdot L_2$.

I omit the remainder of the proof. □

Since there is a characteristic sample S for any 1-start-cyclic-deterministic language L, we know that any learner which guesses L after exposure to S has picked the smallest 1-start-cyclic-deterministic language consistent with S.

The algorithm given below is identical in structure to Algorithm 7. The procedure p-UPDATE ensures that the any (reverse) non-determinism that occurs by merging start states is removed by the time the algorithm terminates. It removes the (reverse) nondeterminism by merging the same b-predecessors of some reverse nondeterministic state (because of multiple incoming b-transitions). It is easy to see that these merges do not change the language accepted by the pre- and post-merged acceptors. Formally, P-UPDATE(B_1, B_2, b) places $(p(B_1, b), p(B_2, b))$ on LIST if both $p(B_1, b)$ and $p(B_2, b)$ are nonempty and defines $p(B_3, b)$ to be $p(B_1, b)$ if this is nonempty and $p(B_2, b)$ otherwise (where B_3 is the union of B_1 and B_2) (cf. Algorithm 7).

It is possible to now prove that for any sample S, the output of Algorithm 8 is the smallest language in $\mathcal{L}_{1fcd}$ which contains S.

Theorem 42 Let S be any nonempty positive sample. Then the output of Algorithm 7 is the smallest 1-start-cyclic-deterministic language containing S.

Proof: Omitted. □

Consequently, Algorithm 8 identifies $\mathcal{L}_{1fcd}$ in the limit.

Algorithm 8 The 1-Start-Cyclic-Reverse-Deterministic (1SCRD) Algorithm

```
Input: a nonempty positive sample $S$.
Ouput: a 1-start-cyclic-deterministic acceptor $A$.
Initialization
Let $A_0 = (Q_0, I_0, F_0, \delta_0) = ST(S)$.
Let $\pi_0$ be the trivial partition of $Q_0$.
For each $b \in \Sigma$ and $q \in Q_0$, let $p(\{q\}, b) = {\delta_0}^r(q, b)$.
Choose some $q'$ in $I_0$.
Let LIST contain all pairs $(q, q')$ such that $q \in I_0 - \{q'\}$.
Let $i = 0$.
Merging
while LIST $\neq \emptyset$ do
  Remove some element $(q_1, q_2)$ from LIST.
  Let $B_1 = B(q_1, \pi_i)$ and $B_2 = B(q_2, \pi_i)$
  if $B_1 \neq B_2$ then
    Let $\pi_{i+1}$ be $\pi_i$ with $B_1$ and $B_2$ merged.
    for all $b \in \Sigma$ do
      $P$-UPDATE$(B_1, B_2, b)$
    end for
    Increase $i$ by 1.
  end if
end while
Termination
Let $f = i$ and output the acceptor $A_0/\pi_f$.
```

Theorem 43 Let L be a 1-start-cyclic-deterministic language, $w_1, w_2, \ldots$ a positive representation of L. For any $i \in \mathbb{N}$ let $S_i = \{w_n : n \leq i\}$ and let $L(PT(S_1)), L(PT(S_2)), \ldots$ be a sequence of languages. This sequence converges to L.

Proof: Omitted. □

5.8 Merging Start States in Prefix Trees

In this section we ask which class of languages is obtained by merging start states of a prefix tree. Since such trees have but one start state, each state will always be in its own block, and thus the merging procedure leaves the prefix tree unaltered. Therefore $\mathcal{L}_{fin}$ is the class of languages identifiable in the limit by this merging procedure.

5.9 Merging Final States in Suffix Trees

Like §5.9, merging final states in suffix trees can makes no changes since suffix trees have only one final state. Therefore, $\mathcal{L}_{fin}$ is the class of languages obtainable in this way.

5.10 Merging Nonfinal States in Prefix Trees

The class of languages obtainable by merging nonfinal states in a prefix tree remains an open question.

5.11 Merging Nonstart States in Suffix Trees

The class of languages obtainable by merging nonstart states in a suffix tree remains an open question. By now it should be clear that this question is clearly related to the one of merging nonfinal states in prefix trees.

5.12 Merging Nonstart States in Prefix Trees

In this section we ask which class of languages is obtained by merging nonstart states in a prefix tree. Since prefix trees have only one start state, all states except the final are merged into a single state. The result is an acceptor which constrains which segments may begin words.

It is easy to see that this procedure is equivalent to a particular string extension learner. Consider $bw : \Sigma^* \rightarrow \Sigma$ defined below

$$bw(w) = \{a : \exists v \in \Sigma^* \text{ such that } av = w\}$$

We call the class of languages this function extends to $\mathcal{L}_{beginwith}$ since the grammars of these languages consists of a set of elements which all words the language accepts must begin with. I.e. G is a subset of Σ and $w \in L(G)$ iff the first segment of the word w belongs to G.

When nonstart states are merged in a prefix tree of some sample, this procedure identifies $\mathcal{L}_{beginwith}$ in the limit.

It is plausible that $\mathcal{L}_{beginwith}$ can be parameterized to yield a family of language classes, each which specifies which strings of length n may begin a given word. One way to accomplish this by state merging might be to compose the I_n function with the function which identifies nonstart states under $\otimes$.

5.13 Merging Nonfinal States in Suffix Trees

As above, when nonfinal states are merged in a suffix tree, the result is an acceptor which constrains which segments may end a word. This follows from the fact that suffix trees have only one final state, so all states but the final are merged into a single state.

It is easy to see that this procedure, like the one in §5.13, is equivalent to a particular string extension learner. Consider $ew : \Sigma^* \rightarrow \Sigma$ defined below

$$ew(w) = \{a : \exists v \in \Sigma^* \text{ such that } va = w\}$$

We call the class of languages this function extends to $\mathcal{L}_{endwith}$ since the grammars of these languages consists of a set of elements which all words the language accepts must end with. I.e. G is a subset of Σ and $w \in L(G)$ iff the last segment of the word w belongs to G.

When nonfinal states are merged in a suffix tree of some sample, this procedure identifies $\mathcal{L}_{endwith}$ in the limit.

Like $\mathcal{L}_{beginwith}$, it is possible to parameterize $\mathcal{L}_{endwith}$ to yield a family of language classes, each which specifies which strings of length n may begin a given word.

5.14 Local Summary

The results of §5 are summarized in Table 6.2. (Each of the language classes below is defined in Table 6.1.) In this table, the language classes shown are identifiable in the limit by the state merging procedure shown. Table 6.2 suggests an interesting relationship between suffix and prefix trees, incoming and outgoing paths, start and final states, and the reversal operator. It appears that that when the suffix tree representation is exchanged for a prefix tree, and a final state with a start state in the f column (or incoming for outgoing), then the class of languages obtained is the reverse of what otherwise would be obtained. For example it is easy to verify that the class of languages obtained by reversing languages in $\mathcal{L}_{endwith}$ is $\mathcal{L}_{beginwith}$. Similarly, reversing languages in $\mathcal{L}_{1fcd}$ yields the class of languages $\mathcal{L}_{1scrd}$. (And of course $\mathcal{L}_{n-gram}$ and $\mathcal{L}_{fin}$ are closed under reversal). These algebraic properties deserve closer study, but I leave such work for future endeavors.

$\mathcal{L}_{fin}$	$=$	$\{L : \lvert L\rvert$ is finite $\}$
$\mathcal{L}_{n-gram}$	$=$	$\{L : L$ is recognizable by ngram grammar $\}$
$\mathcal{L}_{1fd}$	$=$	$\{$ $L : L$ is recognizable by a deterministic acceptor with 1 final state $\}$
$\mathcal{L}_{1fcd}$	$=$	$\{L : L \in \mathcal{L}_{1fd} \cap \mathcal{L}_{fin} \cdot (\mathcal{L}_{fin})^{*}\}$
$\mathcal{L}_{1srd}$	$=$	$\{$ $L : L$ is recognizable by a reverse deterministic acceptor with 1 start state $\}$
$\mathcal{L}_{1scrd}$	$=$	$\{L : L \in \mathcal{L}_{1srd} \cap (\mathcal{L}_{fin})^{*} \cdot \mathcal{L}_{fin}\}$
$\mathcal{L}_{endwith}$	$=$	$\{L : L \in (\Sigma^{*}) \cdot \Sigma^{1}\}$
$\mathcal{L}_{beginwith}$	$=$	$\{L : L \in \Sigma^{1} \cdot (\Sigma^{*})\}$

Table 6.1: Definitions of Language Classes

f	$PT(S)/\pi_f$	$ST(S)/\pi_f$
I_{n+1}	$\mathcal{L}_{n-gram}$	?
O_{n+1}	?	$\mathcal{L}_{n-gram}$
$p, q \in F$	$\mathcal{L}_{1fcd}$	$\mathcal{L}_{fin}$
$p, q \in I$	$\mathcal{L}_{fin}$	$\mathcal{L}_{1scrd}$
$p, q, \notin F$	?	$\mathcal{L}_{endwith}$
$p, q, \notin I$	$\mathcal{L}_{beginwith}$	?

Table 6.2: Language Classes Identifiable in the Limit by Merging States in Prefix and Suffix Trees

6 Summary

This chapter generalizes the notion of neighborhood. It shows that precedence and trigram languages are tail canonically 1-1 neighborhood distinct, but that the class of 4-gram languages is incomparable with 1-1 neighborhood-distinct languages.

This chapter also begins to deconstruct the neighborhood-distinct languages by deconstructing the range of the Forward Backward Neighborhood learning function. It shows that this range is a composition of a number of language classes, many of which are shown to be learnable by some state merging procedure. These language classes are interesting because they plausibly describe other kinds of phonotactic patterns: those that are iterative in character, both left-to-right and right-to-left, those that permit a domain to begin or end in certain ways, the n-gram languages, and the finite languages.

There is still much to understand. Some additional language classes need to be determined. Also, the underlying algebraic structure which includes the prefix/suffix tree distinction, state merging and the reversal operator remains open for investigation.

CHAPTER 7

Conclusion

1 Results

This dissertation explores the thesis that if we understand how learners generalize on the basis of their linguistic experience, then we can understand the kinds of patterns found in natural language. In particular, I examined learners which generalize based on particular notions of locality in the domain of phonotactic patterns. Although real-life learners certainly employ additional principles when generalizing from their limited experience, the focus here makes clear the contribution locality-based inductive principles can make to the phonotactic learning problem in linguistics.

It was shown that bigram and trigram models are instantiations of two more general classes of learning functions: string extension and state-merging. It was shown that if people use inductive principles based on precedence, then it explains why Long Distance Agreement patterns exist and why they do not exhibit blocking effects. It was shown that the class of languages in the range of the precedence learning function and the class of languages in the range of trigram grammars are both proper subsets of a class whose languages obey a more general notion of locality, *neighborhood-distinctness.* A review of two surveys of stress patterns (Bailey 1995, Gordon 2002) reveals that all stress patterns are '2-2' neighborhood-distinct, and almost all stress patterns are '1-1' neighborhood-distinct. 1-1 neighborhood-distinctness thus becomes a universal of the phonotactic patterns discussed in this dissertation, with the few exceptional stress patterns meriting further study.

It was also shown that patterns acquired by generalizing on the basis of neighborhood-distinctness approximate the attested typology of stress patterns in certain significant respects. In particular, generalizing on the basis of neighborhood-distinctness explains why the attested stress patterns are neighborhood-distinct and why patterns describable with feet of size feet of four or more are unattested.

This formal universal also has implications for Optimality-theoretic approaches to phonology, where one goal has been to develop a restricted theory of Con for phonology (Eisner 1997b, McCarthy 2003). A theory of Con which requires all constraints to be neighborhood-distinct severely limits the kinds of possible phonological constraints, but allows precisely the constraints phonologists typically use.

This dissertation proves several results; the ones relevant to learning are summarized in Figure 7.1. The names in this figure are classes of languages for which it is possible to obtain the smallest language which includes any finite sample. The figure shows the subset relationships among the smallest languages of those classes which include the sample. For example, the smallest bigram language which includes a sample also includes the smallest trigram language (because the bigram language makes fewer distinctions). With the exceptions of 1FCD and 1SCRD, all of the language classes represented in Figure 7.1 are neighborhood-distinct.

In short, this dissertation shows that particular formulations what it means for a phonotactic patterns to be ‘local’ in character provide natural, powerful inductive principles by which learners can acquire patterns which resemble the attested typology in significant respects. This work thus suggests that similar discoveries can be made for linguistic rules in other domains.

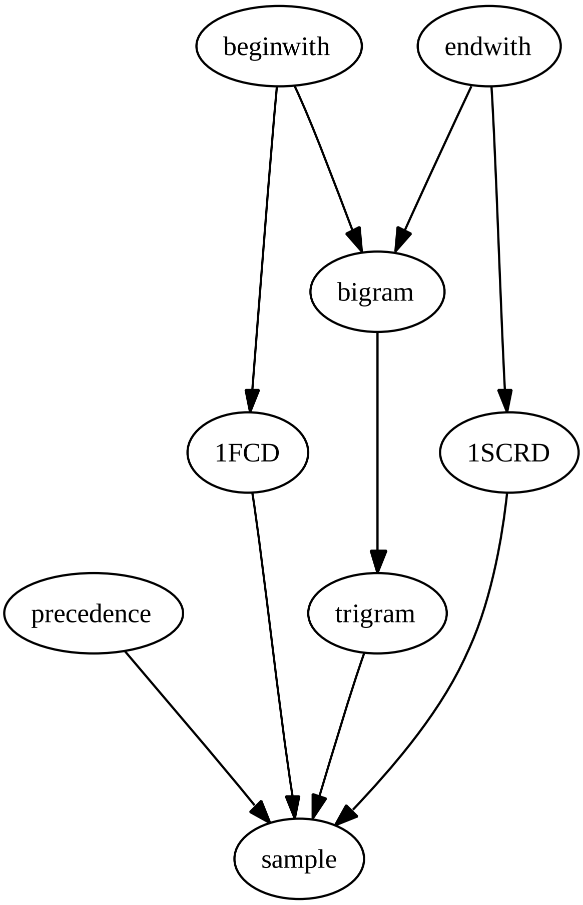

Figure 7.1: Subset Relations Among the Smallest Languages of Particular Classes which Contain some Finite Sample

2 Looking Ahead

It is not, of course, the case that we now know how children acquire phonotactic patterns. Far from it! In many ways, the learners presented here are unlike human learners. These learners are consistent, children are not. The many assumptions that were made at the beginning of this dissertation are now recalled. These learners already know the domain of application of the rules, children do not. These learners are fragile in the sense they cannot handle noisy input, children are not. Despite these dissimarities, there is still a new understanding—there are particular properties of the attested language patterns that human learners can exploit. To turn it around, if children generalize in the sorts of ways explored here, we understand certain aspects of the attested language patterns. Thus I believe that recollecting the ways in which the learners in these pages are unlike human learners is not problematic. Rather each assumption is now a challenge, to be met.

Consider the questions now brought to the fore. Some of these address the assumptions made:

- How can learning stress patterns proceed from segmental transcriptions as opposed to syllabic ones?
- More generally, how can we build a phonotactic learner which simultaneously discovers the phonotactic rule and the domain of its application?
- How can we build gradient counterparts to these hypothesis spaces and stochastic learners for them?
- How can we build versions of these learners that are robust in the presence of noise?
- How can we build iterative versions of the neighborhood learners?

Some of these address the psychological plausibility of the inference procedures investigated here:

- Can these learners explain known experimental results?
- Can predictions these learners make be verified in the lab?

Some of these are typological:

- What is the status of Greenberg's universal regarding completely and partially resolvable clusters?
- Is Long Distance Agreement with local blocking really rare, or should we look more carefully?
- Are all phonotactic patterns neighborhood-distinct, and what about the few stress patterns that are not?

We can also ask whether we can develop learners with properties which better approximate the known typologies:

- What contribution can features make to learning (cf. Minimal Generalization (Albright and Hayes 2002), also note Johnson (1993))?
- How can featural similarity be introduced into the precedence model to account for the similarity effects in Long Distance Agreement Patterns?
- What contribution can the Stress to Weight Principle make to learning stress patterns?

Some questions relate to the types of learners employed:

- What is the exact relation between the languages obtained by string extension learning and linearly separable languages (if any) (Kontorovich et al. 2006)?

- What is exactly the algebraic structure underlying forward and reverse deterministic acceptors, reversal operators, and different state merging procedures?

Finally, we can ask whether the notions of locality discussed here have analogs at higher levels of the Chomsky Hierarchy:

- Can neighborhood-distinctness be generalized to the context-free or context-sensitive domains, what classes of languages do they define, and to what extent do such extensions resemble other natural language patterns and do they lead to learnability in some sense?

There are thus many, many questions whose answers remain unknown but are sure to advance our understanding of how children acquire their knowledge of language. In this respect, I reminded of a saying by Dag Hammersköld[1]: "You climb to the top of a mountain only to see how small it is." That certainly is true, but you also have a better view.

[1] First United Nations Secretary General.

Appendix: The Stress Typology

This appendix lists all the languages included in the stress typology alphabetically by name. The column labeled '#' indicates which stress pattern the language has. Sources are given in the right hand column. The references for Bailey (1995), Gordon (2002) and Hayes (1995), are given in shorthand due to their prevalence in the typology (the full reference is in the bibliography).

Table 7.1: Languages in the Stress Typology

	Name	**#**	**Sources**
1	Abun West	1	Berry, Keith and Berry, Christine. 1999. A description of Abun: a West Papuan languageof Irian. Canberra: Australian National University. • Gordon 2002.
2	Afrikaans	2	Donaldson, B. C. 1993. A grammar of Afrikaans. New York: Mouton. • Gordon 2002.
3	Agul North	3	Alekseev, M. E. 2001. Xinalugskij Yazyk. In Alekseev, M. E., ed. Yazyki Mira:Kavkazskie Yazyki, pp. 460-469. Moscow: Izdatel'stvo Academia. • Gordon 2002.
4	Aklan	4	Halle, Morris and Jean-Roger Vergnaud. 1987. An Essay on Stress. Cambridge, MA: MIT Press. • Bailey 1995.
5	Alabamaa	1	Rand, Earl. 1968. The structural phonology of Alabaman, a Muskogean language.International Journal of American Linguistics 34, 94-103. • Gordon 2002.
6	Alawa	5	Sharpe, Margaret. 1972. Alawa phonology and grammar. Canberra: Australian NationalUniversity. • Gordon 2002.
7	Albanian	5	Hetzer, Armin. 1978. Lehrbuch der vereinheitlichten albanischen Schriftsprache miteinem deutsch-albanischen Worterbuch. Hamburg: Helmut Buske Verlag. • Gordon 2002.

Continued on next page

	Name	#	Sources
8	Amara	5	Thurston, William. 1996. Amara: an Austronesian language of northwestern New Britain.In Ross, M. D., ed. Studies in languages of New Britain and New Ireland, pp. 197-248. Canberra: Australian National University. • Gordon 2002.
9	Amele	6	Roberts, J. A. 1987. Amele. Guildford, England: Biddles. • Bailey 1995.
10	Andamanese	5	Manoharan, S. 1989. A descriptive and comparative study of Andamanese language.Calcutta: Anthropological Survey of India. • Gordon 2002.
11	Anejom	7	Lynch, John. 2000. A grammar of Anejom. Canberra: Australian National University. • Gordon 2002.
12	Anem East	5	Thurston, William. 1982. A comparative study in Anem and Lusi. Canberra: Australian National University. • Gordon 2002.
13	Anguthimri	8	Crowley, Terry. 1981. The Mpakwithi dialect of Anguthimri. In Dixon, R.M.W. and Blake, Barry, eds. Handbook of Australian languages, vol. 2, pp. 146-94. Amsterdam: John Benjamins. • Gordon 2002.
14	Anyula	9	Kirston, Jean. 1967. Anyula Phonology. In Glasgow, D., Glasgow, K., Kirton, J., and Oates, W. J., eds. Papers in Australian Linguistics, pp. 15-28. Canberra: Australian National University. • Gordon 2002.
15	Apalai'a	5	Koehn, Edward and Sally Koehn. 1986. Apalai. In Derbyshire, Desmond and Pullum, Geoffrey, eds. Handbook of Amazonian languages, vol. 1, pp. 3-127. New York: Mouton. • Gordon 2002.
16	Apinaye	1	Burgess, Eunice and Ham, Patricia. 1968. Multilevel conditioning of phoneme variants in Apinaye'. Linguistics: An International Review 41, 5-18. • Gordon 2002.

Continued on next page

	Name	#	Sources
17	Arabana-Wangkanguru	2	Hercus, L. A. 1994. A grammar of the Arabana-Wangkangurru language, Lake EyreBasin, South Australia. Canberra: Australian National University. • Gordon 2002.
18	Arabela	2	Rich, Furne. 1963. Arabela phone mes and high-level phonology. In Elson, Benjamin, ed. Studies in Peruvian Indian Languages I, pp. 193-206. • Gordon 2002.
19	Arabic, Bani-Hassan	10	Kenstowicz, Michael. 1983. Parametric Variation and Accent in the Arabic Dialects. Chicago Linguistic Society 19, 205-213. • Kenstowicz, Michael. 1986. Notes on Syllable Structure in Three Arabic Dialects. Revue quebecoise de linguistique 16, 101-128. • Irshied, Omar and Michael J. Kenstowicz. 1984. Some Phonological Rules of Bani-Hassan Arabic: A Bedouin Dialect. Studies in Linguistic Science 14.1. Dept. of Linguistics, University of Illinois, Urbana, pp. 227-248. • Bailey 1995. • Hayes 1995.
20	Arabic, Cairene	11	Halle, Morris and Jean-Roger Vergnaud. 1987. An Essay on Stress. Cambridge, MA: MIT Press. • Mitchell, T. F. 1975 Principles of Firthian Linguistics. pp. 75-98. Longsman, London. • McCarthy, John. 1979. On Stress and Syllabification. Linguistic Inquiry 10, 443-465. • Bailey 1995.
21	Arabic, Classical	12	McCarthy, John. 1979. On Stress and Syllabification. Linguistic Inquiry 10, 443-465. • Bailey 1995. • Hayes 1995.
22	Arabic, Cyrenaican Bedouin	13	Mitchell, T. F. 1975 Principles of Firthian Linguistics. pp. 75-98. Longsman, London. • Bailey 1995. • Hayes 1995.
23	Arabic, Damascene	14	Halle, Morris and Jean-Roger Vergnaud. 1987. An Essay on Stress. Cambridge, MA: MIT Press. • Bohas and Kouloughli 1981. Processus accentuels en arabe. In Theorie-Analyses. Departement d'arabe, Universite de Paris VIII. • Bailey 1995.

Continued on next page

	Name	#	Sources
24	Arabic, Negev Bedouin	15	Blanc, Haim. 1970. The Arabic Dialect of the Negev Bedouins. Proceedings of the Israeli Academy of Sciences and Humanities 4.7, pp. 112-150. • Kenstowicz. 1981. The Metrical Structure of Arabic Accent. Paper delivered at the UCLA-USC conference on Nonlinear Phonology, LakeArrowhead, Calif. • Kenstowicz, Michael. 1983. Parametric Variation and Accent in the Arabic Dialects. Chicago Linguistic Society 19, 205-213. • Bailey 1995. • Hayes 1995.
25	Arabic, Palestinian	16	Kenstowicz. 1981. The Metrical Structure of Arabic Accent. Paper delivered at the UCLA-USC conference on Nonlinear Phonology, LakeArrowhead, Calif. • Kenstowicz, Michael. 1983. Parametric Variation and Accent in the Arabic Dialects. Chicago Linguistic Society 19, 205-213. • Brame, Michael. 1973. On Stress Assignment in Two Arabic Dialects. In Stephen R. Anderson and Paul Kip[arsky, eds. A Festschrift for Morris Halle. Holt, Rinehart, and Winston, New York. pp 14-25. • Brame, Michael. 1974. The Cycle in Phonology: Stress in Palestinean, Maltese and Spanish. Linguistic Inquiry 5. 39-60. • Bailey 1995. • Hayes 1995.
26	Aramaic	1	Segert, Stanislaw. 1983. Altaramaische Grammatik mit Bibliographie, Chrestomathie und Glossar. Leipzig: Verlag Enzyklopadie. • Gordon 2002.
27	Aranda, Western	17	Halle, Morris and Jean-Roger Vergnaud. 1987. An Essay on Stress. Cambridge, MA: MIT Press. • Davis. 1985. Ternary Feet reconsidered. Ms. Department of Linguistics, MIT, Cambridge, MA. • Bailey 1995.
28	Araucanian	18	Echeverria, Max and Contreras, Helen. 1965. Araucanian phonemics. International Journal of American Linguistics 31, 132-35. • Bailey 1995. • Gordon 2002.

Continued on next page

	Name	#	Sources
29	Arawak	2	de Goeje, Claudius Henricus. 1928. The Arawak language of Guiana. Amsterdam: Amsterdam Koninklijke Akademie van Wetenschappen. • Gordon 2002.
30	Armenian	19	Thomson, R.W. 1975. An introduction to Classical Armenian. New York. • Vaux, Bert. 1998. The Phonology of Armenian. Oxford: Clarendon Press. • Bailey 1995.
31	Asheninca	20	Payne, Judith. 1990. Asheninca Stress Patterns. In Doris L. Payne, ed. Amazonian Linguistics, University of Texas Press, Austin, pp. 185-209. • Bailey 1995. • Hayes 1995.
32	Asmat	21	Voorhoeve, C. 1965. The Flamingo Bay dialect of the Asmat language. 's-Gravenhage:M. Nijhoff. • Gordon 2002.
33	Assiniboine	3	Levin, Norman. 1964. The Assiniboine language. Bloomington: Indiana University Press. Basque isolate Hualde, Jose' Ignacio. 1991. Basque phonology. New York: Routledge. • Gordon 2002.
34	Atayal	1	Egerod, Soren. 1966. A statement on Atayal phonology. Artibus Asiae Supplementum XXIII (Felicitation volume for the 75th birthday of Professor G. H. Luce) 1, 120-30. • Gordon 2002.
35	Atchin	5	Capell, Arthur and Layard, J. 1980. Materials in Atchin, Malekula: grammar, vocabulary, and texts. Canberra: Australian National University. • Gordon 2002.
36	Au	6	Scorza, D. 1985. A sketch of Au morphology and syntax. Papers in New Guinea Linguistics [Pacific Linguistics A63. Canberra: Australian National University], 22, 215-273. • Bailey 1995.
37	Awadhi	22	Saksena, Baburam. 1971. Evolution of Awadhi. Motilal Banarsidass, Delhi. • Bailey 1995. • Hayes 1995. • Hayes 1995.
38	Awtuw	9	Feldman, Harry. 1986. A grammar of Awtuw. Canberra: Australian National University. • Gordon 2002.

Continued on next page

	Name	#	Sources
39	Azerbaijani	1	Householder, Fred. 1965. Basic course in Azerbaijani. Bloomington: Indiana University. • Gordon 2002.
40	Badimaya	8	Dunn, Leone. 1988. Badimaya, a Western Australian language. Papers in Australian Linguistics 17, Pacific Linguistics A71, pp. 19-49. Canberra: Australian National University. • Bailey 1995. • Gordon 2002. • Hayes 1995.
41	Bagandji	23	Hercus, L. A. 1982. The Bagandji language. Canberra: Australian National University. • Gordon 2002.
42	Baluchi	1	Elfenbein, J. 1966. The Baluchi language; a dialectology with texts. London. • Bailey 1995.
43	Bashkir	1	Poppe, Nicholas. 1964. Bashkir manual: descriptive grammar and texts with a Bashkir-English glossary. Bloomington: Indiana University Press. • Gordon 2002.
44	Berbice	7	Kouwenberg, Silvia. 1994. A grammar of Berbice Dutch Creole. New York: Mouton deGruyter. • Gordon 2002.
45	Bhojpuri	24	Hyman, Larry. 1977. On the nature of linguistic stress. Studies in stress and accent: Southern California Occasional Papers in Linguistics 4 ed. by Larry Hyman. Los Angeles: Dept. of Linguistics, University of Southern California. • Bailey 1995.
46	Bhojpuri (per Shukla Tiwari)	25	Shukla, Shaligram. 1981. Bhojpuri Grammar. Georgetown University Press. Washington, D.C. • Tiwari, Udai, Narain. 1960. The Origin and Development of Bhojpuri. Asiatic Society Monograph 10, Asiatic Society, Calcutta. • Bailey 1995. • Hayes 1995.
47	Biangai	26	Dubert, Raymond and Dubert, Marjorie. 1973. Biangai phonemes. In Phonologies of Three Languages of Papua New Guinea, pp. 5-36. Ukarumpa, Papua New Guinea: Summer Institute of Linguistics. • Gordon 2002.

Continued on next page

	Name	#	Sources
48	Bidyara Gungabula	27	Breen, Gavan. 1973. Bidyara and Gungabala: Grammar and vocabulary. Linguistic Communications 8, Melbourne: Monash University. • Bailey 1995. • Gordon 2002. • Hayes 1995.
49	Big Nambas	5	Fox, G. J.. 1979. Big Nambas grammar. Canberra: Australian National University. • Gordon 2002.
50	Bukiyip	5	Conrad, Robert. 1991. An outline of Bukiyip grammar. Canberra: Australian National University. • Gordon 2002.
51	Bulgarian	28	Dogil, Grzegorz. 1995b. Stress and accent in Baltic languages. Word Stress, Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung (AIMS), 2.2:89-112. Universität Stuttgart, Germany. • Bailey 1995.
52	Buriat	29	Poppe, N. 1960. Buriat grammar. Bloomington: Indiana University. • Bailey 1995.
53	Burum	30	Olkkonen, S. 1985. Burum phonology. In Five Phonological Studies. Ukarumpa, Papua New Guinea: Summer Institute of Linguistics. • Gordon 2002.
54	Byelorussian	28	Dogil, Grzegorz. 1995b. Stress and accent in Baltic languages. Word Stress, Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung (AIMS), 2.2:89-112. Universität Stuttgart, Germany. • Bailey 1995.
55	Cahuilla	2	Seiler, Hansjakob. 1957. Die phonetischen Grundlagen der Vokalphoneme des Cahuilla.Zeitschrift fur Phonetik und allgemeine Sprachwissenschaft 10, 204-23. • Seiler, Hansjakob. 1965. Accent and morphophonemics in Cahuilla and Uto-Aztecan. International Journal of American Linguistics 31, 50-9. • Bailey 1995. • Gordon 2002. • Hayes 1995.
56	Cakchiquel	1	Berinstein, Ava. 1979. A cross-linguistic study on the perception and production of stress. UCLA MA Thesis. • Gordon 2002.

Continued on next page

	Name	#	Sources
57	Cambodian	31	Nacaskul, Karnchana. 1978. The Syllabic and Morphological Structure of Cambodian Words. Mon-Khmer Studies 7. 183-200. • Griffith, Teresa. Cambodian as an Iambic Language. Ms. Department of Linguistics, University of California, Irvine. • Bailey 1995. • Hayes 1995.
58	Canela-Kraho	1	Popjes, Jack and Popjes, Jo. 1986. Canela-Krahô. In Derbyshire, Desmond and Pullum, Geoffrey, eds. Handbook of Amazonian languages, vol. 1, pp. 128-129. New York: Mouton. • Gordon 2002.
59	Cavinena	32	Key, Mary. 1968. Comparative Tacanan phonology with Cavinena phonology and notes on Pano-Tacanan relationship. The Hague: Mouton. • Gordon 2002.
60	Cayapaa	2	Lindskoog, John and Ruth Brend. 1962. Cayapa phonemics. In Elson, Benjamin, ed. Studies in Ecuadorian Indian Languages I. Norman: OK: Summer Institute of Linguistics, pp. 31-44. • Gordon 2002.
61	Cayuga	33	Chafe, Wallace L. 1977. Accent and Related Phenomena in the Five Nations Iroquois Languages. In Hyman, Larry, ed. Studies in Stress and Accent, pp. 161-181. Los Angeles: USC Department of Linguistics. • Bailey 1995. • Hayes 1995.
62	Cayuvava	34	Hayes, Bruce. 1981. A metrical theory of stress rules. 1980. Ph.D. thesis, MIT. • Key, Harold. 1961. Phonotactics of Cayuvava. International Journal of AmericanLinguistics 27, 143-50. • Bailey 1995. • Hayes 1995.
63	Central Alaskan Yupik	35	Gordon 2002. • Hayes 1995.
64	Chamalal North	3	Magomedova, P. T. 2001. Chamalinskii Yazyk. In Alekseev, M. E., ed. Yazyki Mira:Kavkazskie Yazyki, pp. 291-298. Moscow: Izdatel'stvo Academia. • Gordon 2002.

Continued on next page

	Name	#	Sources
65	Chamorro	5	Topping, Donald. 1973. Chamorro reference grammar. Honolulu: University of Hawaii Press. • Bailey 1995. • Gordon 2002.
66	Chechen	2	Desherieva, T. I. 2001. Chechenskii Yazyk. In Alekseev, M. E., ed. Yazyki Mira:Kavkazskie Yazyki, pp. 173-185. Moscow: Izdatel'stvo Academia. • Gordon 2002.
67	Chepang	2	Caughley, Ross C. 1969. Chepang phonemic summary. Kirtipur: Summer Institute of Linguistics. • Gordon 2002.
68	Cheremis, Eastern	36	Bailey 1995. • Hayes 1995.
69	Cheremis, Meadow	37	Hayes, Bruce. 1981. A metrical theory of stress rules. 1980. Ph.D. thesis, MIT. • Bailey 1995.
70	Cheremis, Mountain	38	Hayes, Bruce. 1981. A metrical theory of stress rules. 1980. Ph.D. thesis, MIT. • Bailey 1995.
71	Cheremis, Western	38	Bailey 1995. • Hayes 1995.
72	Chimalapa Zoque	9	Knudson, Lyle. 1975. A natural phonology and morphophonemics of Chimalapa Zoque.Papers in Linguistics 8, 283-346. • Bailey 1995. • Gordon 2002. • Hayes 1995.
73	Chitimacha	2	Swadesh, Morris. 1946. Chitimacha. In Osgood, Cornelius, ed. Linguistic structures of Native America, pp. 312-336. New York: Viking Fund Publications in Anthropology. • Gordon 2002.
74	Chulupi	21	Stell, Nelida Noemi. 1972. Fonologia de la Lengua Alulaj. Buenos Aires: Universidad de Buenos Aires. • Gordon 2002.
75	Barbareño root verbs per	5	Beeler, M. S. 1976. Barbareño Chumash: a farrago. In Langdon, Margaret and Silver, Shirley, eds. Hokan Studies: Papers from the 1st Conference on Hokan Languages held in San Diego, California April 23-25, 1970, pp. 251-270. The Hague: Mouton. • Gordon 2002.

Continued on next page

	Name	#	Sources
76	Chutiya	2	Goswami, Upendranath. 1994. An introduction to the Deuri language. Assam:Anundoram Borooah Institute of Language, Art and Culture. • Gordon 2002.
77	Chuvash	39	Krueger, J. R. 1961. Chuvash manual. Bloomington: Indiana University. • Bailey 1995. • Hayes 1995.
78	Cocama	5	Espinosa, Lucas. 1935. Los tupí del oriente peruano, estudio lingüistico y etnográfico.Madrid: Publicaciones de la Expedición Lingüistico • Gordon 2002.
79	Cofán root verbs per	5	Borman, M. B. 1962. Cofan phonemes. In Elson, Benjamin, ed. Studies in Ecuadorian Indian Languages I, pp. 45-59. Norman: OK: Summer Institute of Linguistics.45-59. • Gordon 2002.
80	Comox	2	Hagege, Claude. 1981. Le comox lhaamen de Colombie Britannique: presentation d'unelangue amerindienne. Paris: AEA. • Gordon 2002.
81	Cora	40	Casad, Eugene. 1984. Cora. In Langacker, Ronald, ed. Southern Uto-Aztecan Grammatical Sketches. Dallas: Summer Institute of Linguistics. • Gordon 2002.
82	Coreguaje root	2	Gralow, Frances. 1985. Coreguaje: tone, stress, and intonation. In Brend, Ruth, ed. From Phonology to Discourse: Studies in Six Colombian Languages, pp. 3-11. Dallas: Summer Institute of Linguistics. • Gordon 2002.

Continued on next page

	Name	#	Sources
83	Czech	23	Jakobson, Roman. 1962. Contributions to the study of Czech accent. In Selected WritingsI: Phonological Studies, pp. 614-25. The Hague: Mouton. • Hyman, Larry. 1977. On the nature of linguistic stress. Studies in stress and accent: Southern California Occasional Papers in Linguistics 4 ed. by Larry Hyman. Los Angeles: Dept. of Linguistics, University of Southern California. • Dogil, Grzegorz. 1995a. Stress patterns in West Slavic languages. Word Stress, Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung (AIMS), 2.2:63-88. Universität Stuttgart, Germany. • Bailey 1995. • Gordon 2002. • Hayes 1995.
84	Dagaare (Dagaari)	5	Anttila, A. and Bodomo, A. 1996. Stress and tone in Dagaare. Rutgers Optimality Archive (ROA-169-1296). • Anttila, A. and Bodomo, A. 1996. Stress and tone in Dagaare. Rutgers Optimality Archive (ROA-169-1296). • Bailey 1995.
85	Dakota	3	Shaw, Patricia A. 1985a. Modularisation and Substantive Constraints in Dakota Lexical Phonology. Phonology Yearbook 2. 173-202. • Shaw, Patricia A. 1985b. Coexistent and Competing Stress Rules in Stoney (Dakota). International Journal of American Linguistics 51. 1-18. • Chambers, J. K. 1978. Dakot Accent. In Eung-Do Cook and Jonathan Kaye, eds. Linguistic Studies of North Native Canada. University of British Columbia Press, Vancouver. pp 3-18. • Bailey 1995. • Hayes 1995.
86	Dalabon	8	Capell, Arthur. 1962. Some linguistic types in Australia. Oceania Linguistic Monographs7. University of Sydney. • Bailey 1995. • Gordon 2002. • Hayes 1995.
87	Dani	1	Bromley, Myron. 1981. A grammar of Lower Grand Valley Dani. Canberra: Australian National University. • Gordon 2002.

Continued on next page

	Name	#	Sources
88	Danish	2	Haugen, Einar. 1976. The Scandinavian Languages. Cambrige, MA: Harvard University Press. • Gordon 2002.
89	Dayak (Ngaju)	5	Mihing, T. W. J. and Stokhof, W. A. L. 1977. On the Ngaju Dayak sound system.Miscellaneous Studies in Indonesian and Languages in Indonesia III, 49-59. • Gordon 2002.
90	Dehu	8	Tryon, Darrell. 1967. Nengone Grammar. Canberra: Australia National University. • Bailey 1995. • Gordon 2002. • Hayes 1995.
91	Diegueño root verbs per	41	Langdon, Margaret. 1970. A grammar of Dieguen o: The Mesa Grande dialect. Berkeley:University of California Press. • Bailey 1995. • Gordon 2002. • Hayes 1995.
92	Diola	2	Sapir, David. 1965. A grammar of Diola-Fogny: a language spoken in the Basse-Cassamance region of Senegal. Cambridge: Cambridge University Press. • Gordon 2002.
93	Diyari	27	Austin, Peter. 1981. A grammar of Diyari, South Australia. Cambridge Studies inLinguistics 32. Cambridge: Cambridge University Press. • Bailey 1995. • Gordon 2002. • Hayes 1995.
94	Dizi	2	Breeze, Mary. 1988. Phonological features of Gimira and Dizi. In Bechhaus, Marianneand Serzisko, Fritz, eds. Cushitic-Omotic Papers from the International Symposium on Cushitic and Omotic Languages, Cologne, January 6-9, 1986, pp. 475-490. Hamburg:Helmut Buske Verlag • Gordon 2002.
95	Djapu Yolngu	2	Morphy, Frances. 1983. Djapu, A Yolngu dialect. In Dixon, R.M.W. and Blake, Barry, eds. Handbook of Australian Languages, vol. 3, pp. 1-188. Amsterdam, John Benjamins. • Gordon 2002.
96	Djingili	7	Chadwick, Neil. 1975. A descriptive study of the Djingili language. Canberra: Australian National University. • Bailey 1995. • Gordon 2002. • Hayes 1995.

Continued on next page

	Name	#	Sources
97	Dumi nouns	2	Driem, George van. 1993. A grammar of Dumi. New York: Mouton de Gruyter. • Gordon 2002.
98	Dumi root verbs	1	Driem, George van. 1993. A grammar of Dumi. New York: Mouton de Gruyter. • Gordon 2002.
99	Dutch	42	Bailey 1995.
100	Dyirbal	8	Hyman, Larry. 1977. On the nature of linguistic stress. Studies in stress and accent: Southern California Occasional Papers in Linguistics 4 ed. by Larry Hyman. Los Angeles: Dept. of Linguistics, University of Southern California. • Bailey 1995.
101	Enets, Forest	2	Tereschenko, N. 1993. Enetskii jazyk. In Jartseva, V.N., ed. Jazyki Mira: Uralskiejazyki, pp. 343-349. Moscow: Nauka. • Gordon 2002.
102	Enets, Tundra	2	Tereschenko, N. 1993. Enetskii jazyk. In Jartseva, V.N., ed. Jazyki Mira: Uralskiejazyki, pp. 343-349. Moscow: Nauka. • Gordon 2002.
103	English nouns	14	Burzio, Luigi. 1994. Principles of English Stress. Cambridge: Cambridge University Press. • Hammond, Michael. 1999. The Phonology of English. Oxford University Press. • Bailey 1995.
104	English verbs	43	Burzio, Luigi. 1994. Principles of English Stress. Cambridge: Cambridge University Press. • Hammond, Michael. 1999. The Phonology of English. Oxford University Press. • Bailey 1995.
105	English (nouns, per Pater) nouns	44	Pater, Joe. 1995. ”On the nonuniformity of weight-to-stress and stress preservation effects in English.” Rutgers Optimality Archive (ROA-107-0000). • Bailey 1995.
106	Ese Ejja	7	Key, Mary. 1968. Comparative Tacanan phonology with Cavinena phonology and notes on Pano-Tacanan relationship. The Hague: Mouton. • Gordon 2002.

Continued on next page

	Name	#	Sources
107	Estonian	45	Prince, Alan. 1980. A Metrical Theory for Estonian Quantity. Linguistic Inquiry 11. 511-562. • Bailey 1995. • Hayes 1995.
108	Even	2	Benzing, Johannes. 1955. Lamutische Grammatik; mit Bibliographie, Sprachproben undGlossar. Wiesbaden: F. Steiner. • Gordon 2002.
109	Fijian	46	Schutz. 1985. The Fijian Language, University of Hawaii Press, Honolulu. • Bailey 1995. • Hayes 1995.
110	Finnish	45	Itkonen, E. 1955. Ueber die Betonungsverhältnisse in den finnisch- ugrischen Sprachen. Acta Linguistica Academiae Scientiarum Hungaricae, 5, 21-34. • Bailey 1995. • Hayes 1995.
111	French, European	1	Halle, Morris and Jean-Roger Vergnaud. 1987. An Essay on Stress. Cambridge, MA: MIT Press. • Delattre, Pierre. 1966. Les dix intonations de base du français. French Review 40, 1-14. • Bailey 1995. • Gordon 2002.
112	French,Canadian	19	Gendron, Jean Denis. 1966. Tendances Phonétiques Français Parlé au Canada. Québec: Les Presses de L'université Laval. • Gordon 2002.
113	Gagauz	1	Pokrovskaja, L.A. 1966. Gagauzskij jazyk. In Jazyki Narodov SSSR (Languages of the Soviet Union) 2. Tjurkskie jazyki, ed. by N.A. Baskakov et al. Moscow. • Pokrovskaia, L. A.. 1964. Grammatika gagauzskogo iazyka; fonetika i morfologiia.Moskva: Nauka. • Bailey 1995. • Gordon 2002.
114	Garawa	47	Furby, Christine. 1974. Garawa Phonology. Pacific Linguistics. Canberra: AustralianNational University. • Bailey 1995. • Gordon 2002.
115	Georgian	48	Bailey 1995.
116	Gilyak	2	Panfilov, V. Z. 1962. Grammatika nivkhskogo iazyka. Moskva: Izdatelstvo AkademiiNauk SSSR. • Gordon 2002.

Continued on next page

	Name	#	Sources
117	Golin	50	Bunn, G., and Bunn, R. 1970. Golin phonology. Pacific Linguistics, A23, Canberra: Australian National Univeristy, 1-7. • Bailey 1995. • Gordon 2002. • Hayes 1995.
118	Gondi	2	Steever, Sanford. 1998. Gondi. In Steever, Sanford, ed. The Dravidian Languages, pp.270-300. New York: Routledge. • Gordon 2002.
119	Greek, Ancient	52	Bailey 1995. • Hayes 1995.
120	Greenlandic Inuktitut	1	Thalbitzer, William. 1976. A phonetical study of the Eskimo language: based onobservations made on a journey in north Greenland, 1900-1901. New York: AMS Press. • Fortescue, Michael. 1984. West Greenlandic. London: Croom Helm. • Gordon 2002.
121	Guarani	1	Ayala, José Valentin. 1996. Gramática guaraní. Buenos Aires: Ministerio de Educació cela Nacion. • Gordon 2002.
122	Gugu-Yalanji	53	Oates, William and Lynette Oates. 1964. Gugu-Yalanji and Wik-Munkan LanguageStudies. Canberra: Australian Institute of Aboriginal Studies. • Bailey 1995. • Gordon 2002. • Hayes 1995.
123	Gurage nouns	5	Polotsky, Hans Jakob. 1951. Notes on Gurage grammar. Jerusalem: Israel Oriental Society. • Gordon 2002.
124	Gurage verbs	1	Polotsky, Hans Jakob. 1951. Notes on Gurage grammar. Jerusalem: Israel Oriental Society. • Gordon 2002.
125	Gurkhali	54	Bailey 1995. • Hayes 1995.
126	Gurung	2	Glover, Warren. 1969. Gurung phonemic summary. Kirtipur, Nepal: Summer Institute ofLinguistics. • Gordon 2002.

Continued on next page

	Name	**#**	**Sources**
127	Haitian Creole	1	d'Ans, Andre Marcel. 1968. Le creole francais d'Haiti: Etude des unites d'articulation, d'expansion et de communication. The Hague: Mouton. • Valdman, Albert. 1971. Basic course in Haitian Creole. Bloomington: Indiana University Hebrew Afro-Asiatic Chayen, M. J.. 1973. The phonetics of modern Hebrew. The Hague: Mouton • Gordon 2002.
128	Hanty	23	Re'dei, Ka'roly. 1965. Northern Ostyak chrestomathy. Bloomington: Indiana. Hungarian Uralic Hall, Robert. 1938. An analytical grammar of the Hungarian language. Baltimore: Linguistic Society of America. • Gordon 2002.
129	Hatam West	18	Reesink, Ger P. 1999. A grammar of Hatam. Canberra: Australian National University. • Gordon 2002.
130	Hawaiian	46	Elbert, Samuel and Mary Kawena Pukui. 1979. Hawaiian Grammar. University Press of Hawaii, Honolulu. • Bailey 1995. • Hayes 1995.
131	Hebrew, Tiberian	55	Halle, Morris and Jean-Roger Vergnaud. 1987. An Essay on Stress. Cambridge, MA: MIT Press. • Bailey 1995.
132	Hewa	2	Vollrath, Paul. 1985. Hewa phonemes: a tentative statement. In Five PhonologicalStudies, pp. 51-84. Ukarumpa, Papua New Guinea: Summer Institute of Linguistics. • Gordon 2002.
133	Hinalug root verbs	19	Alekseev, M. E. 2001. Xinalugskij Yazyk. In Alekseev, M. E., ed. Yazyki Mira:Kavkazskie Yazyki, pp. 460-469. Moscow: Izdatel'stvo Academia. • Gordon 2002.
134	Hinalug root nouns	1	Alekseev, M. E. 2001. Xinalugskij Yazyk. In Alekseev, M. E., ed. Yazyki Mira:Kavkazskie Yazyki, pp. 460-469. Moscow: Izdatel'stvo Academia. • Gordon 2002.
135	Hindi (per Fairbanks)	56	Fairbanks, Constance. 1981. The development of Hindi Oral Narrative Meter. Doctoral Dissertation, Dept. of South Asian Language and Literature, University of Wisconsin. • Bailey 1995. • Hayes 1995.

Continued on next page

	Name	#	Sources
136	Hindi (per Jones)	57	Jones, W.E. Syllables and Word Stress in Hindi. Jounral of the International Phonetic Association 1, 74-78. • Bailey 1995. • Hayes 1995.
137	Hindi (per Kelkar) root verbs per	58	Kelkar, A. R. 1968. Studies in Hindi-Urdu I: Introduction and word phonology. Deccan College, Poona. • Bailey 1995. • Hayes 1995.
138	Hindi (per Sharma)	57	Sharma, Aryendra. 1969. Hindi Word Accent. Indian Linguistics 30. 115-118. • Bailey 1995. • Hayes 1995.
139	Hixkarya'naa	1	Derbyshire, Desmond. 1985. Hixkaryana and linguistic typology. Dallas: Summer Institute of Linguistics. • Gordon 2002.
140	Hopi	59	Hayes, Bruce. 1981. A metrical theory of stress rules. 1980. Ph.D. thesis, MIT. • Bailey 1995. • Hayes 1995.
141	Hualpai	2	Redden, James. 1966. Walapai I: Phonology. International Journal of American Linguistics 32, 1-16. • Gordon 2002.
142	Huasteco	39	Larsen, R. S., and Pike, E. V. 1949. Huasteco intonations and phonemes. Language, 25, 268-277. • Bailey 1995.
143	Huitoto	2	Minor, Eugene and Minor, Dorothy. 1976. Fonologia del huitoto. Bogota: Ministerio de Gobierno. • Gordon 2002.
144	Hungarian	60	Kerek, A. 1971. Hungarian Metrics: Some Linguistic Aspects of Iambic Verse. Indiana University Publications, Uralic and Altaic Series 117. Mouton, The Hague. • Bailey 1995. • Hayes 1995.
145	Iban	1	Asmah Haji Omar. 1981. The Iban language of Sarawak : a grammatical description. Kuala Lumpur: Dewan Bahasa dan Pustaka, Kementerian Pelajaran Malaysia. • Gordon 2002.
146	Icelandic	23	Bailey 1995. • Gordon 2002. • Hayes 1995.

Continued on next page

	Name	#	Sources
147	Içuã Tupi	61	Hyman, Larry. 1977. On the nature of linguistic stress. Studies in stress and accent: Southern California Occasional Papers in Linguistics 4 ed. by Larry Hyman. Los Angeles: Dept. of Linguistics, University of Southern California. • Abrahamson, Arne. 1968. Contrastive distribution of Phoneme Classes in Içuã Tupi. Anthropological Linguistics 10.6. 11-22.
148	Ignaciano	3	Ott, Willis and Burke de Ott, Rebecca. 1983. Diccinario ignaciano y castellano.Cochabamba, Bolivia: Instituto Lingüístico de Verano. • Gordon 2002.
149	Indo-European (protolanguage)	6	Halle, Morris and Jean-Roger Vergnaud. 1987. An Essay on Stress. Cambridge, MA: MIT Press. • Bailey 1995.
150	Indonesian	62	Cohn, Abigail. 1993. The Initial Dactly Effect in Indonesian. Linguistic Inquiry 24. 372-381. • Gordon 2002. • Hayes 1995.
151	Inga	52	Levinsohn, Stephen H. 1976. The Inga Language. Janua linguarum, Series practica 188. Mouton, the Hague. • Bailey 1995. • Hayes 1995.
152	Ingush	2	Desheriev, Y. D. and Desherieva, T. I. 2001. Ingushskii Yazyk. In Alekseev, M. E., ed. Yazyki Mira: Kavkazskie Yazyki, pp. 186-195. Moscow: Izdatel'stvo Academia. • Gordon 2002.
153	Ioway-Oto	63	Whitman, William. 1947. Descriptive grammar of Ioway-Oto. International Journal ofAmerican Linguistics 13, 233-248. • Gordon 2002.
154	Irish	2	Mhac an Fhailigh, Éamonn. 1968. The Irish of Erris, Co. Mayo: a phonemic study.Dublin: Dublin Institute for Advanced Studies. • Gordon 2002.

Continued on next page

	Name	#	Sources
155	Ishkashim	1	Grierson, G.A. 1920. Ishkashmi, Zebaki, and Jazghulami, an account of three eranian dialects. London. • Pakhalina, T.N. 1966. Iskasimskij jazyk. In Jazyki Narodov SSSR (Languages of the Soviet Union) 1. Indo-evropejskie jazyki, ed. V.V. Vinogradov et al. Moscow. • Bailey 1995.
156	Itelmen	1	Stebnickij, S.N. 1934. Itel'menskij Jazyk. Jazyki narodov severa, ch. 3. Moscow-Leningrad. • Bailey 1995.
157	Ivatan	1	Hidalgo, Cesar and Hidalgo, Araceli. 1971. A tagmemic grammar of Ivatan. Manila:Linguistic Society of the Philippines. • Gordon 2002.
158	Jaqaru	5	Hardman-de-Bautista, Martha James. 1966. Jaqaru: outline of phonological andmorphological structure. The Hague: Mouton. • Gordon 2002.
159	Javanese	64	Herrfurth. 1964. Lehrbuch des modernen Djawanisch, Veb Verlag Enzyklopadie, Leipzig. • Bailey 1995. • Hayes 1995.
160	Jazghulam	1	Grierson, G.A. 1920. Ishkashmi, Zebaki, and Jazghulami, an account of three eranian dialects. London. • Edel'man, D.I. 1966. Jazguljamskij jazyk. In Jazyki Narodov SSSR (Languages of the Soviet Union) 1. Indo-evropejskie jazyki, ed. V.V. Vinogradov et al. Moscow. • Bailey 1995.
161	Jemez	2	Bell, Alan. 1993. Jemez tones and stress. Colorado Research in Linguistics 12, pp. 26-34.Boulder, Colorado: University of Colorado at Boulder. • Gordon 2002.
162	Kaliai-Kove	5	Counts, David. 1969. A grammar of Kaliai-Kove. Honolulu: University of Hawaii Press. • Gordon 2002.
163	Kalkatungu	2	Blake, Barry. 1979. A Kalkatungu grammar. Canberra: Australian National University • Gordon 2002.
164	Kamayura'	21	Saelzer, Meinke. 1976. Fonologia provisória da língua Kamayura'. In Bridgeman, Loraine, ed. Série Lingüística 5, 131-70. • Gordon 2002.

Continued on next page

	Name	#	Sources
165	Kambera	2	Klamer, Margaretha. 1994. Kambera: a language of Eastern Indonesia. The Hague: Holland Academic Graphics. • Gordon 2002.
166	Karaim	1	Musaev, K.M. 1966. Karaimskij jazyk. In Jazyki Narodov SSSR (Languages of the Soviet Union) 2. Tjurkskie jazyki, ed. by N.A. Baskakov et al. Moscow. • Bailey 1995.
167	Karelian	8	Leskinen, Heikki. 1984. Über die Phonemsystem der Karelischen Sprache. In Hajdú, Péter and Honti, László, eds. Studien zur Phonologischen Beschreibung uralischer Sprachen. pp. 247-257. Budapest: Akadémiai Kiadó. • Bailey 1995. • Gordon 2002. • Hayes 1995.
168	Kashmiri	65	Bhatt, R. 1989. Syllable weight and metrical structure of Kashmiri. Unpublished Ms., University of Illinois, Urbana. • Kenstowicz, M. 1993. Peak prominence stress systems and optimality theory. Proceedings of the 1st International Conference on Linguistics at Chosun University, Foreign Culture Research Institute, Chosun University, Korea. • Bailey 1995.
169	Kashubian	2	Dogil, Grzegorz. 1995a. Stress patterns in West Slavic languages. Word Stress, Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung (AIMS), 2.2:63-88. Universität Stuttgart, Germany. • Bailey 1995.
170	Kashubian, Slovincian	39	Dogil, Grzegorz. 1995a. Stress patterns in West Slavic languages. Word Stress, Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung (AIMS), 2.2:63-88. Universität Stuttgart, Germany. • Bailey 1995.
171	Kate	8	Flierl, W. and Strauss, H. 1977. Kâte dictionary. Canberra: Australian National University. • Gordon 2002.

Continued on next page

	Name	#	Sources
172	Kavalan	1	Li, Paul Jen-Kuei. 1982. Kavalan phonology. In Carle, Rainer, Heinschke, Martina, Pink,Peter, Rost, Christel, and Stadtlander, Karen, eds. GAVA, Studies in Austronesian Languages and Cultures, pp. 479-496. Berlin: Dietrich Reimer. • Gordon 2002.
173	Kawaiisu	66	Zigmond, Mauricew L., Curtis G. Booth, and Pamela Munro. 1990. Kawaiisu: A Grammar and Dictionary, University of California Publications in Linguistics 119, University of California Press, Berekeley and Los Angeles. • Bailey 1995. • Hayes 1995.
174	Kayabi'	1	Dobson, Rose. 1988. Aspectos da língua kayabi'. Brasília: Summer Institute of Linguistics. • Gordon 2002.
175	Kayapo'	1	Stout, Mickey and Ruth Thomson. 1974. Fonémica Txukuhamei (Kayapo'). In Bridgeman, Loraine, ed. Série Lingüística 3, pp. 153-176. Brasília: Summer Institute of Linguistics 3. • Gordon 2002.
176	Kazakh	1	Sovremennyi kazakhskii iazyk : fonetika i morfologiia. 1962. Alma-Ata: Izdatel'stvo Akademii nauk Kazakhskoi. • Gordon 2002.
177	Kela	40	Collier, Ken and Collier, Margaret. 1975. A tentative phonemic statement of the Apozedialect, Kela language. In Loving, Richard, ed. Phonologies of Five Austronesian Languages, pp. 129-61. Ukarumpa: Summer Institute of Linguistics. • Bailey 1995. • Gordon 2002. • Hayes 1995.
178	Ket	2	Skorit, P. Ja., et al. 1968. Jazyki Narodov SSSR (Languages of the Soviet Union) 5. Mongol'skie, tunguso-manczurskie i paleoaziatskie jazyki. Leningrad. • Bailey 1995. • Gordon 2002.
179	Khakas	1	Karpov, V.G. 1966. Xakasskij jazyk. In Jazyki Narodov SSSR (Languages of the Soviet Union) 2. Tjurkskie jazyki, ed. by N.A. Baskakov et al. Moscow. • Bailey 1995.

Continued on next page

	Name	#	Sources
180	Kinnauri	2	Sarma, Devidatta. 1988. A descriptive grammar of Kinnauri. Delhi: Mittal Publications. • Gordon 2002.
181	Klamath	67	Barker, Muhammad Abd-al-Rahman. 1964. Klamath Grammar. University of California Publications in Linguistics 32, University of California Press, Berkeley. • Barker, Muhammad Abd-al-Rahman. 1963. Klamath Dictionary. University of California Publications in Linguistics 31, University of California Press, Berkeley. • Bailey 1995. • Hayes 1995.
182	Kola	5	Takata, Masahiro and Takata, Yuko. 1992. Kola phonology. In Burquest, Donald and Laidig, Wyn, eds. Descriptive Studies in Languages of Maluku, pp. 31-46. Jakarta: Badan Penyelenggara Seri Nusa. • Gordon 2002.
183	Kolami	2	Emeneau, M. B. 1955. Kolami, a Dravidian language. Berkeley: University of California Press. • Gordon 2002.
184	Komi	68	Itkonen, E. 1955. Ueber die Betonungsverhältnisse in den finnisch- ugrischen Sprachen. Acta Linguistica Academiae Scientiarum Hungaricae, 5, 21-34. • Bailey 1995. • Hayes 1995.
185	Konkani	1	Maffei, Angelus Francis Xavier. 1986. A Konkani grammar. New Delhi: Asian Educational Services. • Gordon 2002.
186	Korafe	2	Farr, J. B. and Farr, C. J. M.. 1974. A preliminary Korafe phonology. Workpapers inPapua New Guinea Linguistics 3, 5-38. • Gordon 2002.
187	Koromfe'	2	Rennison, John. 1997. Koromfe. New York: Routledge. Kota Dravidian Emeneau, M. B. 1944. Kota texts Part I. Berkeley: University of California Press • Gordon 2002.
188	Koryak	3	Zhukova, Alevtina Nikodimovna. 1972. Grammatika Koriakskogo Iazyka: Fonetika,Morfologiia. Leningrad: Izdatelstvo Nauka. • Gordon 2002.
189	Koya	2	Idsardi, W. J. 1992. The computation of prosody. Ph.D. thesis, MIT. • Bailey 1995.

Continued on next page

	Name	#	Sources
190	Kumukh	1	Magomedov, A.G. 1966. Kumykskij jazyk. In Jazyki Narodov SSSR (Languages of the Soviet Union) 2. Tjurkskie jazyki, ed. by N.A. Baskakov et al. Moscow. • Bailey 1995.
191	Kung (Zu—' Hõasĩ)	2	Snyman, J. W. 1970. An introduction to the !Xu (!Kung) language. Capetown:Balkema academic and technical publications, University of Cape Town, School of African Studies. • Gordon 2002.
192	Kurdish	1	Kurdoev, K. K. 1957. Grammatika kurdskogo iazyka (kurmandzhi): fonetika,morfologiia. Moskva: Izdatel'stvo Akademii nauk SSSR. • Bakaev, U.Kh. 1966. Kyrdskij jazyk. In Jazyki Narodov SSSR (Languages of the Soviet Union) 1. Indo-evropejskie jazyki, ed. by V.V. Vinogradov et al. Moskow. • Bailey 1995. • Gordon 2002. • Gordon 2002.
193	Kutenai	5	Canestrelli, Philippo. 1926. Grammar of the Kutena: Language. International Journal ofAmerican Linguistics 4, 1-84. • Gordon 2002.
194	Kuuku Yaʔu	69	Thompson, D. A. 1976. A phonology of Kuuku-Yaʔu. In P. Sutton, ed. Languages of Cape York. Canberra: Australian Institute of Aboriginal Studies. • Bailey 1995. • Hayes 1995.
195	Kwaio	5	Keesing, Roger. 1985. Kwaio grammar. Canberra: Australian National University. • Gordon 2002.
196	Kwakw'ala Kwakiutl	68	Boas, F. 1947. Kwakiutl grammar with a glossary of the suffixes. Transactions of the American Philosophical Society, New Series 37, Part 3, 201-377. • Bailey 1995. • Hayes 1995.
197	Labu	5	Siegel, Jeff. 1984. Introduction to the Labu language. Papers in New Guinea Linguistics 23, 83-157. • Gordon 2002.

Continued on next page

	Name	#	Sources
198	Lakota	3	Boas, Franz and Deloria, Ella. 1933. Notes on the Dakota, Teton dialect. International Journal of American Linguistics 7, 97-121. • Boas, Franz and Deloria, Ella. 1941. Dakota Grammar. Memoirs of the National Academy of Sciences, vol. 33. Washington: United States Government Printing Office. • Gordon 2002.
199	Lamba	5	Doke, Clement. 1938. Text book of Lamba grammar. Johannesburg: Witwatersrand University Press. • Gordon 2002.
200	Lango root	1	Noonan, Michael. 1992. A grammar of Lango. New York: Mouton. • Gordon 2002.
201	Lappish, Central Norwegian	8	Itkonen, E. 1955. Ueber die Betonungsverhältnisse in den finnisch- ugrischen Sprachen. Acta Linguistica Academiae Scientiarum Hungaricae, 5, 21-34. • Bailey 1995. • Hayes 1995.
202	Larike root	7	Laidig, Carol. 1992. Segments, syllables, and stress in Larike. In Burquest, Donald and Laidig, Wyn, eds. Phonological studies in four languages of Maluku, pp. 67-126. Dallas: Summer Institute of Linguistics. • Gordon 2002.
203	Latin, Classical	70	Mester, R. Armin. 1994. The quantitative trochee in Latin. Natural Language and Linguistic Theory 12: 1-61. • Bailey 1995.
204	Latvian	2	Halle, Morris and Jean-Roger Vergnaud. 1987. An Essay on Stress. Cambridge, MA: MIT Press. • Endzelins, Janis. 1922. Lettisches lesebuch, grammatische und metrischevorbemerkungen, texte und glossar. Heidelberg: C. Winter. • Fennell, Trevor and Gelsen, Henry. 1980. A grammar of modern Latvian. New York:Mouton. • Bailey 1995. • Gordon 2002.

Continued on next page

	Name	#	Sources
205	Laz verbs	40	Marr, Nikolai Iakovlevich. 1910. Grammatika chanskago (lazskago) iazyka, skhrestomatieiu i slovarem. Saint Petersburg: Tipografiia Imperatorskoi Akademii Nauk. • Jgenti, S. 1959. Chanur-megrulis ponetika. Tbilisi. • Bailey 1995. • Gordon 2002.
206	Laz nouns	5	Marr, Nikolai Iakovlevich. 1910. Grammatika chanskago (lazskago) iazyka, skhrestomatieiu i slovarem. Saint Petersburg: Tipografiia Imperatorskoi Akademii Nauk. • Jgenti, S. 1959. Chanur-megrulis ponetika. Tbilisi. • Bailey 1995. • Gordon 2002.
207	Lenakel verbs	94	Lynch, John. 1974. Lenakel Phonology. Doctoral Dissertation, University of Hawaii. • Hayes 1995.
208	Lenakel nouns per	7	Lynch, John. 1974. Lenakel Phonology. Doctoral Dissertation, University of Hawaii. • Bailey 1995. • Hayes 1995.
209	Lezg	1	Meijlanova, U.A. 1967. Lezginskij jazyk. In Jazyki Narodov SSSR (Languages of the Soviet Union) 4. Iberijsko-kavkazskie jazyki, ed. V.I. Lytkin et al. Moscow. • Bailey 1995.
210	Lezgian North	3	Haspelmath, Martin, 1993. A grammar of Lezgian. New York: Mouton. • Gordon 2002.
211	Lingala	5	Redden, James and Bongo, F. 1963. Lingala, basic course. Washington: Department of State. • Gordon 2002.
212	Lithuanian	71	Halle, Morris and Jean-Roger Vergnaud. 1987. An Essay on Stress. Cambridge, MA: MIT Press. • Dogil, Grzegorz. 1995a. Stress patterns in West Slavic languages. Word Stress, Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung (AIMS), 2.2:63-88. Universität Stuttgart, Germany. • Bailey 1995.
213	Livonian	23	Kettunen, Lauri. 1938. Livisches Wo"rterbuch mit grammatischer Einleitung. Helsinki:Suomalais-Ugrilainen Seura. • Bailey 1995. • Gordon 2002. • Hayes 1995.

Continued on next page

	Name	#	Sources
214	Lushootseed	6	Hess, T. 1976. Dictionary of Puget Salish. Seattle: University of Washington Press. • Odden, D. 1979. Principles of stress assignment: a crosslinguistic view. Studies in the Linguistic Sciences, 9(1), 157-176. • Bailey 1995. • Hayes 1995.
215	Lusi	5	Thurston, William. 1982. A comparative study in Anem and Lusi. Canberra: Australian National University. • Gordon 2002.
216	Macedonian root verbs	40	Lunt, Horace. 1952. A grammar of the Macedonian Literary Language. Skopje. • Bailey 1995. • Gordon 2002.
217	Mae	40	Capell, Arthur. 1962. The Polynesian Language of Mae (Emwae), New Hebrides. Auckland: Linguistic Society of New Zealand. • Bailey 1995. • Gordon 2002. • Hayes 1995.
218	Maidu	72	Shipley, William F. 1964. Maidu Grammar. University of California Publications in Linguistics 41. University of California Press, Berkeley and Los Angeles. • Bailey 1995. • Hayes 1995.
219	Maithili	73	Jha, Subdara. 1940-1944. Maithili Phonetics. Indian Linguistics 8. 435-459. • Bailey 1995. • Hayes 1995.
220	Malakmalak	74	Birk, D. B. W. 1976. The Malakmalak Language, Daly River (Western Arnhem Land). Canberra: Australian National University. • Bailey 1995. • Gordon 2002. • Hayes 1995.
221	Malay (per Lewis)	75	Lewis, M.B. 1947. Teach Yourself Malay. English Universities Press, London. • Bailey 1995. • Hayes 1995.
222	Malay (per Winstedt)	64	Winstedt, Richard O. 1927. Malay Grammar. Oxford University Press, Oxford. • Bailey 1995. • Hayes 1995.
223	Malayalam	54	Mohanan. 1986. The Theory of Lexical Phonology. Reidel, Dordrecht. 113-19. • Bailey 1995. • Hayes 1995.
224	Malecite Passamaquoddy	76	Stowell, T. 1979. Stress systems of the world, unite! In K. Safir, ed. Papers on Syllable Structure, Metrical Structure, and Harmony Processes. MIT Working Papers in Linguistics 1. • Hayes 1995.

Continued on next page

	Name	#	Sources
225	Mam	77	England, Nora. 1983. A Grammar of Mam, a Mayan Language. University of Texas Press, Austin. • Bailey 1995. • Hayes 1995.
226	Mamainde	50	Eberhard, David. 1995. Mamaindé stress: The need for strata. Summer Institute of Linguistics and the University of Texas at Arlington Publications in Linguistics, 122. Dallas: Summer Institute of Linguistics and the University of Texas at Arlington. ix, 159 p. • Bailey 1995.
227	Manam	78	Buckley, Eugene. 1994. Alignment and constraint domains in Manam stress. Rutgers Optimality Archive (ROA-56-0000). • Bailey 1995.
228	Manobo, Sarangani (per DuBois)	22	DuBois, Carl D. 1976. Sarangani Manobo: An Introductory Guide, Philippine Journal of Linguistics, Special Monograph Issue 6. Linguistic Society of the Philippines, Manila. • Bailey 1995. • Hayes 1995.
229	Manobo, Sarangani (per Meiklejohn and Meiklejohn)	79	Meiklejohn, Percy and Kathleen Meiklejohn. 1958. Accentuation in Sarangani Manobo. Studies in Philippine Linguistics, Oceania Linguistic Monographs, no. 3. University of Sydney, Australia, pp. 1-3. • Bailey 1995. • Hayes 1995.
230	Mansi	23	Kálmán, Béla. 1965. Vogul Chrestomathy. Bloomington: Indiana University. • Bailey 1995. • Gordon 2002. • Hayes 1995.
231	Mantjiltjara	2	Marsh, James. 1969. Mantjiljara Phonology. Oceanic Linguistics 8. 131-152. • Bailey 1995. • Hayes 1995.
232	Maori	80	Hyman, Larry. 1977. On the nature of linguistic stress. Studies in stress and accent: Southern California Occasional Papers in Linguistics 4 ed. by Larry Hyman. Los Angeles: Dept. of Linguistics, University of Southern California. • Bailey 1995.

Continued on next page

	Name	#	Sources
233	Maranungku	81	Halle, Morris and Jean-Roger Vergnaud. 1987. An Essay on Stress. Cambridge, MA: MIT Press. • Bailey 1995. • Hayes 1995.
234	Maricopa	1	Gordon, Lynn. 1982. Maricopa morphology and syntax. Berkeley: University of California Press. • Gordon 2002.
235	Mayan, Aguacatec	50	Halle, Morris and Jean-Roger Vergnaud. 1987. An Essay on Stress. Cambridge, MA: MIT Press. • McArthur, H. and McArthur, L. 1956. Aguacatec (Mayan) phonemes within the stress group. IJAL, 22, 72-76. • Bailey 1995. • Hayes 1995.
236	Mayi	2	Breen, Gavan. 1981. The Mayi Languages of the Queensland Gulf Counttry. AIAS new series 29, Australia Institute of Aboriginal Studies, Canberra. • Bailey 1995. • Hayes 1995.
237	Mazatec	1	Jamieson, Allan. 1977. Chiquihuitlan Mazatec phonology. In Merrifield, William, ed.Studies in Otomanguean Phonology, pp. 93-106. Arlington, TX: Summer Institute of Linguistics. • Gordon 2002.
238	Mingrelian	115	Klimov, G. A. 2001. Megrel'skii Yazyk. In Alekseev, M. E., ed. Yazyki Mira: Kavkazskie Yazyki, pp. 52-58. Moscow: Izdatel'stvo Academia.
239	Miwok, Sierra	83	Bailey 1995. • Hayes 1995.
240	Mixe, Totontepec	2	Crawford, John. 1963. Totontepec Mixe phonotagmemics. Norman, Okla.: Summer Institute of Linguistics. • Gordon 2002.
241	Moghol	1	Weiers, Michael. 1972. Die Sprache der Moghol der Provinz Herat in Afghanistan. Opladen: Westdeutscher Verlag. • Gordon 2002.
242	Mohawk	84	Michelson, Karin. 1988. A comparative study of Lake-Iroquoian accent. Dordrecht:Kluwer. • Gordon 2002.

Continued on next page

	Name	#	Sources
243	Mongolian, Khalkha (per Bosson) root verbs per	85	Bosson, J. E. 1964. Modern Mongolian. Bloomington, Indiana: Indiana University. • Bailey 1995.
244	Mongolian, Khalkha (per Street)	86	Street, John C. 1963. Khalkha Structure. Uralic and Altaic series 24. Indiana University, Bloomington. • Bailey 1995.
245	Mongolian, Khalkha (per Stuart)	87	Stuart, Don G. and Matthew M. Haltod. 1957. The phonology of the word in modern standard Mongolian. Word 13. 65-99. • Bailey 1995.
246	Monumbo	5	Vormann, Franz. 1914. Die Monumbo-Sprache: Grammatik und Wörterverzeichnis. Wien: Mechitharisten Buchdruckerei. • Gordon 2002.
247	Mordwin, Erzyan	2	Tsygankin, D. B. and Debaev, C. Z. 1975. Ocherk Sravnitel'noj Grammatiki Mordovskix (Mokshanskoko i Erz'anskoko) Literaturnix Jazykov. Saransk. • Kenstowicz, M. 1994. Sonority-driven stress. In Rutgers Optimality Archive, ruccs.rutgers.edu. • Walker, R. 1996. Prominence-driven stress. Rutgers Optimality Archive (ROA-172-0197). • Bailey 1995.
248	Mordwin, Mokshan	6	Tsygankin, D. B. and Debaev, C. Z. 1975. Ocherk Sravnitel'noj Grammatiki Mordovskix (Mokshanskoko i Erz'anskoko) Literaturnix Jazykov. Saransk. • Kenstowicz, M. 1994. Sonority-driven stress. In Rutgers Optimality Archive, ruccs.rutgers.edu. • Walker, R. 1996. Prominence-driven stress. Rutgers Optimality Archive (ROA-172-0197). • Bailey 1995.
249	Movima	5	Judy, Roberto. 1962. Fonemas del movima con atención especial a la serie glottal. Cochabamba: Instituto Lingüístico de Verano. • Gordon 2002.

Continued on next page

	Name	#	Sources
250	Muna	7	Berg, René van den. 1989. A grammar of the Muna language. Dordrecht: Foris. • Gordon 2002.
251	Munsee	88	Goddard, Ives. 1979. Delaware Verbal Morphology. Garland Publishing, New York. • Goddard, Ives. 1982. The Historical Phonology of Munsee. International Journal of American Linguistics 48. 16-48. • Bailey 1995. • Hayes 1995.
252	Murik	89	Abbott, S. 1985. A tentative multilevel multiunit phonological analysis of the Murik language. Papers in New Guinea Linguistics [Pacific Linguistics A63, Canberra: Australian National Univeristy], 22, 339-373. • Bailey 1995. • Hayes 1995.
253	Murinbata	23	Street, Chester S. and Mollingin, Gregory P.. 1981. The phonology of Murinbata. In Waters, Bruce, ed. Australian phonologies: Collected papers, pp. 183-244. Darwin: Summer Institute of Linguistics. • Gordon 2002.
254	Murut	9	Prentice, D. J. 1971. The Murut languages of Sabah. Canberra: The Australian National University. • Gordon 2002.
255	Mussau	5	Blust , Robert. 1984. A Mussau vocabulary with phonological notes. Papers in New Guinea Linguistics 23, 159-208. • Gordon 2002.
256	Naga	2	Arokainathan, S. 1980. Tangkhul Naga phonetic reader. Mysore: Central Institute ofIndian Languages. • Gordon 2002.
257	Nahuatl	5	Tuggy, David. 1979. Tetelcingo Nahuatl. In Langacker, Ronald, ed. Studies in Uto-Aztecan Grammar, vol. 2, Modern Aztec Grammatical Sketches, pp. 1-140. Arlington, TX: Summer Institute of Linguistics. • Gordon 2002.
258	Nama	2	Hagman, Roy. 1977. Nama Hottentot grammar. Bloomington: Research Center for Language and Semiotic Studies, University of Indiana. • Gordon 2002.
259	Nanai	1	Avrorin, V. A. 1959. Grammatika nanaiskogo iazyka. Leningrad: Izdatel'stvo Akademii Nauk. • Gordon 2002.

Continued on next page

	Name	#	Sources
260	Nanay	1	Avrorin, V. A. 1959. Grammatika nanaiskogo iazyka. Leningrad: Izdatel'stvo Akademii Nauk. • Bailey 1995.
261	Nenets	2	De'csy, Gyula. 1966. Yurak Chrestomathy. Bloomington: Indiana University Press. • Gordon 2002.
262	Nengone	7	Tryon, Darrell. 1967. Nengone Grammar. Canberra: Australia National University. • Bailey 1995. • Gordon 2002. • Hayes 1995.
263	Ningil	23	Manning, Margaret and Saggers, Naomi. 1977. A tentative phonemic analysis of Ningil.Phonologies of Five Papua New Guinea Languages languages, pp. 49-72. Ukarumpa, Papua New Guinea: Summer Institute of Linguistics. • Gordon 2002.
264	Nubian, Dongolese	90	Hayes, Bruce. 1981. A metrical theory of stress rules. 1980. Ph.D. thesis, MIT. • Bailey 1995.
265	Nyawaygi	91	Dixon, Robert M. W. 1983. Nyawaygi. in R.M.W. Dixon and Barry J. Blake, eds. Handbook of Australian Languages, Vol. 2. John Benjamins, Amsterdam. pp. 430-531. • Bailey 1995. • Hayes 1995.
266	Olo	2	McGregor, Donald and McGregor, Aileen. 1982. Olo language materials. Canberra: Australian National University. • Gordon 2002.
267	Ono	23	Phinnemore, Thomas. 1985. Ono Phonology and Morphophonemics. Papers in NewGuinea Linguistics 22, 173-214. • Bailey 1995. • Gordon 2002. • Hayes 1995.
268	Onondaga	5	Chafe, Wallace. 1970. A semantically based sketch of Onondaga. Indiana University publications in anthropology and linguistics, memoir 25. Baltimore: Waverly Press (Indiana University). • Bailey 1995. • Gordon 2002. • Hayes 1995.
269	Oroch	1	Skorit, P. Ja., et al. 1968. Jazyki Narodov SSSR (Languages of the Soviet Union) 5. Mongol'skie, tunguso-manczurskie i paleoaziatskie jazyki. Leningrad. • Bailey 1995.

Continued on next page

	Name	#	Sources
270	Orokaiva	2	Healey, Alan, Isoroembo, Ambrose, and Chittleborough, Martin. 1969. Prelimary noteson Orokaiva grammar. Papers in New Guinea Linguistics 9, 33-64. • Gordon 2002.
271	Orokolo	7	Brown, Herbert. 1986. A comparative dictionary of Orokolo, Gulf of Papua. Canberra: Australian National University. • Gordon 2002.
272	Ossetic	72	Abaev, Vasilii Ivanovich. 1964. A Grammtical Sketch of Ossetic. International Journal of American Linguistics, v. 30, no. 4, pt. 2. Indiana University of Center in Anthropology, Folklore and Linguistics, Bloomington. • Bailey 1995. • Hayes 1995.
273	Paamese	92	Crowley, Terry. 1982. The Paamese Language of Vanuatu. Pacific Linguistics B87. Australian National University, Canberra. • Bailey 1995. • Hayes 1995.
274	Pagu West	5	Wimbish, Sandra. 1992. Pagu phonology. In Burquest, Donald and Laidig, Wyn, eds. Descriptive studies in languages of Maluku, vol. 34, pp. 69-90. Jakarta: Badan Penyelenggara Seri Nusa, Universitas Katolik Indonesia Atma Java. • Gordon 2002.
275	Paipaib	1	Joel, Dina Judith. 1966. Paipai phonology and morphology. UCLA Ph.D. dissertation. • Gordon 2002.
276	Paiwan	5	Ferrell, Raleigh. 1982. Paiwan dictionary. Canberra: Australian National University. • Gordon 2002.
277	Panamint	23	Dayley, Jon. 1989. Tümpisa (Panamint) Shoshone Grammar. University of California Publications in Linguistics 115. Berkeley: University of California Press. • Gordon 2002.
278	Papago root	2	Saxton, Dean. 1963. Papago phonemes. International Journal of American Linguistics 29,29-35. • Gordon 2002.
279	Paraujano	3	Patte, Marie France. 1989. Estudio descriptivo de la lengua añún (o paraujano). San Cristobal: Universidad Católica del Táchira. • Gordon 2002.

Continued on next page

	Name	#	**Sources**
280	Parintintin Tenharim root verbs per	2	Pease, Helen and Betts, LaVera. 1971. Parintintin phonology. Tupi Studies I, 1-14. • Gordon 2002.
281	Parnkalla	40	Schürmann, Clamor Wilhem. 1984. A vocabulary of the Parnkalla language. Adelaide: George Dehane. • Bailey 1995. • Gordon 2002. • Hayes 1995.
282	Pawnee nouns	2	
283	Pemon	1	Armellada, Cesáreo. 1943. Gramática y diccionario de la lengua pemón (arekuna, taurepán, kamarakoto) (familia Caribe). Caracas: Artes gráficas. • Gordon 2002.
284	Persian	1	Windfuhr, Gernott. 1990. Persian. In Comrie, Bernard, ed. The world's major languages, pp. 523-46. New York: Oxford. • Gordon 2002.
285	Persian, Old	6	Lambton, A.K.S. 1961. Persian grammer. (rev. edn.). University of Cambridge Press: Cambridge. • Oranskij, I.M. 1963. Iranskie jazyki. Moscow. (French translation: Les langues iraniennes. Paris, 1977). • Bailey 1995.
286	Pintupi	8	Hansen, Kenneth and L.E. 1969. Pintupi phonology. Oceanic Linguistics 8, 153-70. • Hansen, Kenneth and L.E. 1978. The core of Pintupi grammar. Alice Springs: Institute for Aboriginal Development. • Bailey 1995. • Gordon 2002. • Hayes 1995.
287	Piraha	93	Everett, D. and Everett, K. 1984. On the relevance of syllable onsets to stress placement. Linguistic Inquiry 15: 705-711. • Everett, Daniel L. 1988. On Metrical Constituent Structure in Pirahã. Natural Language and Linguistic Theory 6. 207-246. • Halle, Morris and Jean-Roger Vergnaud. 1987. An Essay on Stress. Cambridge, MA: MIT Press. • Bailey 1995. • Hayes 1995.

Continued on next page

	Name	#	Sources
288	Piro	94	Matteson, Esther. 1965. The Piro (Arawakan) Language. Berkeley: University of California Press. • Bailey 1995. • Gordon 2002. • Hayes 1995.
289	Pitta Pitta	27	Blake, Barry. 1969. Pitta-Pitta. In Dixon, R.M.W. and Blake, Barry, eds. Handbook of Australian languages, vol. 1. pp. 182-242. Amsterdam: John Benjamins. • Bailey 1995. • Gordon 2002. • Hayes 1995.
290	Polabian (per Dogil)	39	Dogil, Grzegorz. 1995a. Stress patterns in West Slavic languages. Word Stress, Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung (AIMS), 2.2:63-88. Universität Stuttgart, Germany. • Bailey 1995.
291	Polabian (per Olesch)	43	Dogil, Grzegorz. 1995a. Stress patterns in West Slavic languages. Word Stress, Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung (AIMS), 2.2:63-88. Universität Stuttgart, Germany. • Bailey 1995.
292	Polish (per Geert and Rubach) root verbs	94	Booij, Geert and Jerzy Rubach. 1985. A grid theory of stress in Polish. Lingua 66, 281-319. • Gordon 2002.
293	Polish (per Dogil)	9	Dogil, Grzegorz. 1995a. Stress patterns in West Slavic languages. Word Stress, Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung (AIMS), 2.2:63-88. Universität Stuttgart, Germany. • Bailey 1995.
294	Pomo, Eastern	2	McLendon, Sally. 1975. A grammar of Eastern Pomo. Berkeley: University of California Press. • Gordon 2002.
295	Quicha	5	Orr, Carolyn. 1962. Ecuador Quicha phonology. In Elson, Benjamin, ed. Studies in Ecuadorian Indian Languages I, pp. 60-77. Norman, OK: Summer Institute of Linguistics. • Gordon 2002.
296	Quileute	5	Powell, J. V and Woodruff, Fred. 1976. Quileute dictionary. Moscow: University of Idaho. • Gordon 2002.

Continued on next page

	Name	#	Sources
297	Rapanui	5	Du Feu, Veronica. 1996. Rapanui. New York: Routledge. • Gordon 2002.
298	Romanian nouns	95	Chitoran, Ioana. 1996. Prominence vs. rhythm: The predictability of stress in Romanian. Grammatical Theory and Romance Languages, ed. by K. Zagona. Current Issues in Linguistic Theory, 144. Amsterdam/Philadelphia: John Benjamins, pp. 47-58. • Bailey 1995.
299	Romanian verbs	21	Chitoran, Ioana. 1996. Prominence vs. rhythm: The predictability of stress in Romanian. Grammatical Theory and Romance Languages, ed. by K. Zagona. Current Issues in Linguistic Theory, 144. Amsterdam/Philadelphia: John Benjamins, pp. 47-58. • Bailey 1995.
300	Romansh, Berguener (Berguner)	96	Kamprath. 1987. Suprasegmental Structure in a Raeto-Romansch Dialect: A Case Study in Metrical and Lexical Phonology. Ph.D.dissertation. University of Texas, Austin. • Bailey 1995. • Hayes 1995.
301	Rotumen	72	Bailey 1995.
302	Russian	6	Idsardi, W. J. 1992. The computation of prosody. Ph.D. thesis, MIT. • Bailey 1995.
303	Sa'mi, Eastern	2	Aimä, Frans. 1914. Phonetik und Lautlehre des Inarilappischen. Helsinki: Druckerei der Finnischen Literaturgesellschaft. • Gordon 2002.
304	Saam	2	Kert, G.M. 1971. Saamskij jazyk. Leningrad. • Bailey 1995.
305	Sámi, Northern	30	Nielsen, Konrad. 1926. Laerebok i Lappisk. Oslo: A. W. Broggers. • Gordon 2002.
306	Sango	2	Samarin, William J. 1967. A grammar of Sango. The Hague: Mouton. • Gordon 2002.
307	Sanskrit, Vedic	6	Halle, Morris and Jean-Roger Vergnaud. 1987. An Essay on Stress. Cambridge, MA: MIT Press. • Bailey 1995. • Hayes 1995.

Continued on next page

	Name	#	Sources
308	Santali	2	Chakrabarti, Byomkes. 1994. A comparative study of Santali and Bengali. Calcutta: KPBagchit Company. • Gordon 2002.
309	Sanuma	9	Borgman, Donald. 1989. Sanuma. In Derbyshire, Desmond and Pullum, Geoffrey, eds. Handbook of Amazonian languages, vol. 2, pp. 15-248. New York: Mouton. • Gordon 2002.
310	Selepet	30	McElhanon, K. A. 1970. Selepet phonology. Canberra: Australian National University. • Bailey 1995. • Gordon 2002. • Hayes 1995.
311	Selkup	39	Kuznecova, A. N., Xelimskij, E. A., and Grushkina, E. V. 1980. Ocherki po sel'kupskomu jazyku. Moscow: Izdatel'stvo Moskovskogo Universiteta. • Halle, M., and Clements, G. N. 1983. Problem book in phonology. Cambridge, Mass.: MIT Press. • Idsardi, W. J. 1992. The computation of prosody. Ph.D. thesis, MIT. • Bailey 1995.
312	Semai	1	Means, Nathalie and Means, Paul. 1986. Sengoi-English, English-Sengoi dictionary. Toronto: University of Toronto. • Gordon 2002.
313	Seminole Creek	97	Haas, Mary. 1977. Tonal accent in Creek. In Hyman, Larry, ed. Studies in Stress and Accent, pp. 195-208. Los Angeles: USC Department of Linguistics. • Halle, Morris and Jean-Roger Vergnaud. 1987. An Essay on Stress. Cambridge, MA: MIT Press. • Bailey 1995. • Hayes 1995.
314	Seneca	98	Bailey 1995.
315	Senoufo	2	Mills, Elizabeth. 1984. Senoufo phonology, discourse to syllable (a prosodic approach). Dallas: Summer Institute of Linguistics. • Gordon 2002.
316	Sentani	99	Cowan, Hendrik K. J. 1965. Grammar of the Sentani Language. Verhandelingen van het Koninklijk Instituut voor Taal- Land- en Volkenkunde 47. Martinus Nijhoff, The Hague. • Bailey 1995. • Hayes 1995.

Continued on next page

	Name	**#**	**Sources**
317	Serbo-Croatian	6	Inkelas, Sharon and Draga Zec. 1988. Serbo-Croatian Pitch Accent: The Interaction of Tone, Stress, and Intonation. Language 64. 227-248. • Bailey 1995. • Hayes 1995.
318	Setswana	5	The sound system of Setswana. 1999. Department of African Languages and Literature, University of Botswana. Gaborone, Botswana: Lightbooks. • Gordon 2002.
319	Shilhab	1	Applegate, Joseph. 1958. An outline of the structure of Shilha. New York: American Council of Learned Societies. • Gordon 2002.
320	Shona	5	Stevick, Earl. 1965. Shona; basic course. Washington: Department of State. • Gordon 2002.
321	Shoshone, Gosiute	51	Miller, Wick. 1996. Sketch of Shoshone, a Uto-Aztecan language. In Ives Goddard (volume editor). Handbook of American Indian Languages, vol. 17 Languages. Washington: Smithsonian Institute. 693-720. • Gordon 2002.
322	Shoshone, Tümpisa	100	Dayley, Jon. 1989. Tümpisa (Panamint) Shoshone Grammar. University of California Publications in Linguistics 115. Berkeley: University of California Press. • Dayley, Jon. 1989. Tümpisa (Panamint) Shoshone Dictionary. University of California Publications in Linguistics 116. Berkeley: University of California Press. • Bailey 1995. • Hayes 1995.
323	Sibutu Sama	9	Kager, René. 1997. Generalized alignment and morphological parsing. Rivista di Linguistica 9, 245-82. • Allison, E. J. 1979. The phonology of Sibutu Sama: a language of the Southern Philippines. In Edrial-Luzares, C. and Hale, A., eds. Studies in Philippine Linguistics 3:2, pp. 63-104. Linguistic Society of the Philippines and Summer Institute of Linguistics. • Gordon 2002.

Continued on next page

	Name	#	Sources
324	Sinaugoro	23	Tauberschmidt, Gerhard. 1999. A grammar of Sinaugoro : an Austronesian language ofthe Central Province of Papua New Guinea. Canberra: Australian National University. • Gordon 2002.
325	Sindhi	101	Stowell, T. 1979. Stress systems of the world, unite! In K. Safir, ed. Papers on Syllable Structure, Metrical Structure, and Harmony Processes. MIT Working Papers in Linguistics 1. • Walker, R. 1996. Prominence-driven stress. Rutgers Optimality Archive (ROA-172-0197). • Bailey 1995. • Hayes 1995.
326	Siona	2	Wheeler, Alva and Wheeler, Margaret. 1962. In Elson, Benjamin, ed. Studies inEcuadorian Indian Languages I, pp. 96-113. Norman, Okla.: Summer Institute of Linguistics. • Gordon 2002.
327	Sirionó	5	Schermair, Anselmo. 1949. Gramática de la lengua Sirionó. La Paz. • Gordon 2002.
328	Siroi	3	Wells, Margaret. 1979. Siroi grammar. Canberra: Australian National University. • Gordon 2002.
329	Slovak	8	Dogil, Grzegorz. 1995a. Stress patterns in West Slavic languages. Word Stress, Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung (AIMS), 2.2:63-88. Universität Stuttgart, Germany. • Bailey 1995.
330	Solor	5	Arndt, P. P. 1937. Grammatik der Solor-Sprache. Ende, Flores: Arnoldus-Drukkerij. • Gordon 2002.
331	Sorbian	102	Dogil, Grzegorz. 1995a. Stress patterns in West Slavic languages. Word Stress, Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung (AIMS), 2.2:63-88. Universität Stuttgart, Germany. • Bailey 1995.
332	Sotho	5	Endemann, Karl. 1964. Versuch einer Grammatik des Sotho. Farnborough, England: Gregg Press. • Gordon 2002.

Continued on next page

	Name	#	Sources
333	Southern Paiute	103	Sapir, Edward. 1930. Southern Paiute: A Shoshonean Language. Cambridge, Mass: American Academy of Arts and Sciences. • Harms, Robert. 1966. Stress, Voice, and Length in Southern Paiute. International Journal of American Linguistics 32, 228-35. • Bailey 1995. • Gordon 2002. • Hayes 1995.
334	Spanish	5	Harris, James W. 1995. Projection and edge marking in the computation of stress in Spanish. A Handbook of Phonological Theory ed. by John Goldsmith, (Current issues in language-oriented phonological studies section). Oxford: Basil Blackwell, Ltd. • Harris, James W. 1995. Projection and edge marking in the computation of stress in Spanish. A Handbook of Phonological Theory ed. by John Goldsmith, (Current issues in language-oriented phonological studies section). Oxford: Basil Blackwell, Ltd. • Bailey 1995.
335	Stieng	1	Miller, Vera Grace. 1976. An overview of Stiêng grammar. Grand Forks: Summer Institute of Linguistics. • Gordon 2002.
336	Sumbanese	2	Klamer, Margaretha. 1994. Kambera: a language of Eastern Indonesia. The Hague: Holland Academic Graphics. • Gordon 2002.
337	Suruwaha'	21	Everett, Daniel. 1996. Prosodic levels and constraints in Banawa and Suruwaha. Ms. University of Pittsburgh. Available online, ROA-121, Rutgers Optimality Archive. • Gordon 2002.
338	Swahili	5	Ashton, E. O.1959. Swahili grammar. London: Longmans. • Bailey 1995. • Gordon 2002. • Hayes 1995.
339	Swedish	2	Haugen, Einar. 1976. The Scandinavian Languages. Cambrige, MA: Harvard University Press. • Gordon 2002.

Continued on next page

	Name	#	Sources
340	Tacana	5	Key, Mary. 1968. Comparative Tacanan phonology with Cavinena phonology and notes on Pano-Tacanan relationship. The Hague: Mouton. • Gordon 2002.
341	Tadzhic	1	Kerimova, A.A. 1966. Tadzhikskij jazyk. In Jazyki Narodov SSSR (Languages of the Soviet Union) 1. Indo-evropejskie jazyki, ed. V.V. Vinogradov et al. Moscow. • Bailey 1995.
342	Tagalog	5	Hyman, Larry. 1977. On the nature of linguistic stress. Studies in stress and accent: Southern California Occasional Papers in Linguistics 4 ed. by Larry Hyman. Los Angeles: Dept. of Linguistics, University of Southern California. • Bailey 1995.
343	Tajiki	1	Rastorgueva, V. S.. 1963. A Short sketch of Tajik grammar. Bloomington, Indiana University. • Gordon 2002.
344	Tamazight Berber	1	Abdel-Massih, Ernest. 1971. A reference grammar of Tamazight: a comparative study of the Berber dialects of the Ayt Ayache and Ayt Seghrouchen. Ann Arbor: Center for Near Eastern and North African Studies, University of Michigan. • Gordon 2002.
345	Tanna, Southwest	5	Lynch, John D. Southwest Tanna Grammar and Vocabulary. In J. Lynch, ed. Papers in the Linguistics of Melanesia 4, Australian National University, Canberra. 1-91. • Bailey 1995. • Hayes 1995.
346	Tatar	1	Poppe, Nicholas. 1963. Tatar manual: descriptive grammar and texts with a Tatar-English glossary. Bloomington: Indiana University Press. • Gordon 2002.
347	Tauya	104	MacDonald, Lorna. 1990. A grammar of Tauya. New York: Mouton de Gruyter. • Gordon 2002.
348	Tawala	7	Ezard, Bryan. 1997. A grammar of Tawala : an Austronesian language of the Milne Bayarea, Papua New Guinea. Canberra: Australian National University. • Gordon 2002.

Continued on next page

	Name	#	Sources
349	Temiar	1	Carey, Iskandar, 1961. Tengleq kui Serok; a study of the Temiar language, with anethnographical summary. Kuala Lumpur: Dewan Bahasa dan Pustaka. • Gordon 2002.
350	Tenango Otami	8	Blight, Richard and Pike, Eunice. 1976. The phonology of Tenango Otomi. International Journal of American Linguistics 42, 51-57. • Gordon 2002.
351	Ternate West	5	Watuseke, F. S. 1991. The Ternate language. In Dutton, Tom, ed. Papers in Papuan Linguistics 1, pp. 223-244. Canberra: Australian National University. • Gordon 2002.
352	Tetun	5	Morris, Cliff. 1984. Tetun-English dictionary. Canberra: Australian National University. • Gordon 2002.
353	Tewa	2	Harrington, John. 1910. A brief description of the Tewa language. American Anthropologist 12, 497-504. • Speirs, Randall. 1966. Some aspects of the structure of Rio Grande Tewa. SUNY Buffalo Ph.D. dissertation. • Gordon 2002.
354	Thai	1	Noss, Richard B. 1964. Thai: reference grammar. Washington: Foreign Service Institute,Department of State. • Gordon 2002.
355	Tibetan, Lhasa	6	Odden, D. 1979. Principles of stress assignment: a crosslinguistic view. Studies in the Linguistic Sciences, 9(1), 157-176. • Bailey 1995. • Hayes 1995.
356	Tigak	2	Beaumont, Clive H. 1979. The Tigak language of New Ireland. Canberra: Australian National University. • Gordon 2002.
357	Timucua	23	Granberry, Julian. 1993. A grammar and dictionary of the Timucua language. Tuscaloosa: University of Alabama Press. • Gordon 2002.
358	Tinrin	2	Walker, R. 1996. Prominence-driven stress. Rutgers Optimality Archive (ROA-172-0197). • Osumi, Midori, 1995. Tinrin grammar. Honolulu : University of Hawai'i Press. • Bailey 1995. • Gordon 2002.

Continued on next page

	Name	#	Sources
359	Tiwi	5	Osborne, C. R. 1974. The Tiwi language: grammar, myths and dictionary of the Tiwilanguage spoken on Melville and Bathurst Islands, Northern Australia. Canberra: Australian Institute of Aboriginal Studies. • Gordon 2002.
360	To'abaita	7	Lichtenberk, Frantisek. 1984. Toabaita language of Malaita, Solomon Islands. Auckland: University of Auckland. • Gordon 2002.
361	Tojolabal	5	Furbee-Losee, Louanna. 1976. The correct language, Tojolabal: a grammar with ethnographic notes. New York: Garland. • Gordon 2002.
362	Tol	43	Fleming, Ilah, and Ronald K. Dennis. 1977. Tol (Jicaque) Phonology. International Journal of American Linguistics 43. 121-127. • Bailey 1995. • Hayes 1995.
363	Tolai	3	Franklin, Karl, Kerr, Harland, and Beaumont, Clive. 1974. Tolai language course.Huntington Beach, Calif.: Summer Institute of Linguistics. • Gordon 2002.
364	Tolo	5	Crowley, Susan Smith. 1986. Tolo dictionary. Canberra: Australian National University. • Gordon 2002.
365	Tolowa Na	1	Bright, Jane. 1964. The phonology of Smith River Athapaskan (Tolowa). International Journal of American Linguistics 30, 101-7. • Gordon 2002.
366	Tongan	105	Churchward, C. Maxwell. 1953. Tongan Grammar. Oxford University Press, London. • Bailey 1995. • Hayes 1995.
367	Tonkawa	5	Hoijer, Harry. 1946. Tonkawa. In Osgood, Cornelius, ed. Linguistic Structures of Native America, pp. 55-84. New York: Viking Fund Publications in Anthropology. • Gordon 2002.
368	Toraja Kesu'	5	Sande, J. S. and Stokhof, W. A. L. 1977. On the phonology of the Toraja Kesu' dialect. Miscellaneous Studies in Indonesian and Languages of Indonesia IV, pp. 19-34. • Gordon 2002.

Continued on next page

	Name	#	Sources
369	Tsaxur North	3	Talibov, B. B. 2001. Tsaxurskii Yazyk. In Alekseev, M. E., ed. Yazyki Mira:Kavkazskie Yazyki, pp. 420-427. Moscow: Izdatel'stvo Academia. • Gordon 2002.
370	Tsotsil	1	Weathers, Nadine. 1947. Tsotsil phonemes with special reference to allophones of B. International Journal of American Linguistics 13, 108-111. • Gordon 2002.
371	Tübatulabal	1	Voegelin, Charles. 1935. Tübatulabal grammar. Berkeley: University of California Press. • Bailey 1995. • Gordon 2002. • Hayes 1995.
372	Tukang Besi	7	Donohue, Mark, 1999. A grammar of Tukang Besi. New York: Mouton. • Gordon 2002.
373	Tunica	2	Swanton, John. 1921. The Tunica language. International Journal of American Linguistics 2, 1-39. • Gordon 2002.
374	Turkish	1	Inkelas, Sharon. 1994. "Exceptional stress-attracting suffixes in Turkish: representations vs. the grammar". Paper presented at the Workshop on Prosodic Morphology, Utrecht University. Rutgers Optimality Archive, ROA39. • Bailey 1995.
375	Turkmen	1	Hanser, Oskar. 1977. Turkmen manual : descriptive grammar of contemporary literary Turkmen: texts: glossary. Wien: Verlag des Verbandes der wissenschaftlichen Gesellschaft Österreichs. • Bailey 1995. • Gordon 2002.
376	Tuscarora	5	Mithun, Marianne. 1976. A grammar of Tuscarora. New York: Garland. • Gordon 2002.
377	Tuva	1	Sat, Sh.Ch. 1966. Tuvinskij jazyk. In Jazyki Narodov SSSR (Languages of the Soviet Union) 2. Tjurkskie jazyki, ed. by N.A. Baskakov et al. Moscow. • Bailey 1995.
378	Tzutujil	1	Dayley, Jon. 1985. Tzutujil grammar. Berkeley: University of California Press. • Gordon 2002.

Continued on next page

	Name	#	Sources
379	Udi North	1	Dzeyranishvili, E. F. 2001. Udinskij Yazyk. In Alekseev, M. E., ed. Yazyki Mira:Kavkazskie Yazyki, pp. 453-458. Moscow: Izdatel'stvo Academia. • Gordon 2002.
380	Udihe	106	Nikolaeva, Irina and Maria Tolskaya. 2001. A Grammar of Udihe. New York: Mouton. • Kormushin, Igor Valentinovich. 1998. Udykheiskii (Udegeiskii) Iazyk. Moscow: Nauka. • Gordon 2002.
381	Udmurt	1	Kel'makov, V. K. 1993. Udmurtskij Jazyk. In Jartseva, V.N., ed. Jazyki Mira: Uralskiejazyki, pp. 239-255. Moscow: Nauka. • Lytkin, V.I., et al. 1966. Jazyki Narodov SSSR (Languages of the Soviet Union) 3. Finno-ugorskie jazyki i samodijskie jazyki. Moscow. • Bailey 1995. • Gordon 2002.
382	Uighur	1	Nadzhip, E. N. 1971. Modern Uigur. Moscow, Nauka. • Gordon 2002.
383	Ukrainian	28	Dogil, Grzegorz. 1995b. Stress and accent in Baltic languages. Word Stress, Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung (AIMS), 2.2:89-112. Universität Stuttgart, Germany. • Bailey 1995.
384	Ulwa	72	Thomas Green. 1999. A Lexicographic Study of Ulwa. PhD Thesis. MIT. • Bailey 1995. • Hayes 1995.
385	Unami	88	Goddard, Ives. 1979. Delaware Verbal Morphology. Garland Publishing, New York. • Goddard, Ives. 1982. The Historical Phonology of Munsee. International Journal of American Linguistics 48. 16-48. • Bailey 1995. • Hayes 1995.
386	Ura	7	Crowley, Terry. 1998. Ura. München: Lincom Europa. Warao isolate Osborn, Henry: 1966. Warao I: Phonology and morphophonemics. International Journal of American Linguistics 32, 108-23. • Gordon 2002.

Continued on next page

	Name	#	Sources
387	Urubú Kaapor	21	Kakumasu, James. 1986. Urubu-Kaapor. In Derbyshire, Desmond and Pullum, Geoffrey, eds. Handbook of Amazonian languages, vol. 1, pp. 326-406. New York: Mouton. • Gordon 2002.
388	Usan	5	Reesink, Ger. 1987. Structures and their functions in Usan: a Papuan language of Papua New Guinea. Philadelphia: Benjamins. • Gordon 2002.
389	Uzbek	1	Poppe, Nicholas. 1962. Uzbek newspaper reader, with glossary. Bloomington: Indiana University Press. • Bailey 1995. • Gordon 2002.
390	Votic	23	Ariste, Paul. 1968. A grammar of the Votic language. Bloomington: Indiana University Press. • Bailey 1995. • Gordon 2002. • Hayes 1995.
391	Waiwai	1	Hawkins, W. Neill. 1952. A fonologia da língua uáiuái. São Paulo: Brazil Universidade. • Gordon 2002.
392	Walmatjari	107	Hudson, Joyce. 1978. The core of Walmatjari grammar. Canberra: Australian Institute of Aboriginal Studies. • Gordon 2002.
393	Wangkumara	27	McDonald, M. and Wurm, Stephen. 1979. Basic materials in Wangkumara (Galali):Grammar, sentences, and vocabulary. Pacific Linguistics B65. Canberra: Australian National University. • Bailey 1995. • Gordon 2002. • Hayes 1995.
394	Waorani root	23	Echeverria, Max and Contreras, Helen. 1965. Araucanian phonemics. International Journal of American Linguistics 31, 132-35. • Gordon 2002.
395	Wappo	40	Radin, Paul. 1929. A grammar of the Wappo language. Berkeley: University of California Press. • Gordon 2002.

Continued on next page

	Name	#	Sources
396	Warao	7	Osborn, Henry. 1966. Warao I: Phonology and Morphophonemics. International Journal of American Linguistics 32. 108-123. • Halle, Morris and Jean-Roger Vergnaud. 1987. An Essay on Stress. Cambridge, MA: MIT Press. • Bailey 1995. • Hayes 1995.
397	Wardaman	5	Merlan, Francesca. 1994. A grammar of Wardaman: a language of the Northern Territory of Australia. New York: Mouton de Gruyter. • Gordon 2002.
398	Wargamay	108	Dixon, Robert M. W. 1981. Wargamay. In R.M.W. Dixon and Barry J. Blake, eds. Handbook of Australian Languages, Vol. 2. John Benjamins, Amsterdam. pp. 1-144. • Bailey 1995. • Hayes 1995.
399	Wari'	1	MacEachern, Margaret, Barbara Kern, and Peter Ladefoged. 1997. Wari' phonetic structures. The Journal of Amazonian Languages 1, 5-30. • Gordon 2002.
400	Waskia	1	Ross, Malcolm. 1978. A Waskia grammar sketch and vocabulary. Canberra: Australian National University. • Gordon 2002.
401	Watjarri	53	Douglas, Wilfrid. 1981. Watjarri. In Dixon, R.M.W. and Blake, Barry, eds. Handbook of Australian Languages, vol. 2, pp. 196-272. Amsterdam,: J. Benjamins. • Gordon 2002.
402	Welsh	5	Williams, B. J. (1983). Stress in modern Welsh. Unpublished Ph.D. thesis, University of Cambridge, Cambridge, UK. • Bailey 1995.
403	Wembawemba	2	Hercus, L. A. 1986. Victorian languages, a late survey. Canberra: Australian National University. • Gordon 2002.
404	Weri	21	Boxwell, Helen and Maurice Boxwell. 1966. Weri phonemes. In Wurm, Stephen A., ed. Papers in New Guinea Linguistics 5, pp. 77-93. Canberra: Australian National University. • Bailey 1995. • Gordon 2002.

Continued on next page

	Name	#	Sources
405	Wikchamni	5	Gamble, Geoffrey. 1978. Wikchamni grammar. Berkeley: University of California Press Final. • Gordon 2002.
406	Wirangu	8	Hercus, L. A. 1999. A grammar of the Wirangu language from the west coast of South Australia. Canberra: Pacific Linguistics. • Gordon 2002.
407	Yagua	1	Payne, Doris and Payne, Thomas. 1990. Yagua. In Derbyshire, Desmond and Pullum, Geoffrey, eds. Handbook of Amazonian Languages, vol.2, pp. 249-474. New York: Mouton. • Gordon 2002.
408	Yakut	1	Krueger, John. 1962. Yakut manual; area handbook, grammar, graded reader and glossary. Bloomington: Indiana University. • Gordon 2002.
409	Yana	6	Sapir, E. and Swadesh, M. 1960. Yana dictionary. Berkeley: University of California Press. • Bailey 1995. • Hayes 1995.
410	Yapese	109	Hayes, Bruce. 1981. A metrical theory of stress rules. 1980. Ph.D. thesis, MIT. • Bailey 1995.
411	Yavapai	1	Walker, R. 1996. Prominence-driven stress. Rutgers Optimality Archive (ROA-172-0197). • Kendall, M. B. 1976. Selected problems in Yavapai syntax. New York: Garland. • Langdon, Margaret. 1977. Stress, length, and pitch in Yuman languages. In Hyman, Larry, ed. Studies in Stress and Accent, pp. 239-259. Los Angeles: USC Department of Linguistics. • Bailey 1995. • Gordon 2002.
412	Yawelmani	5	Archangeli, D. 1984. Underspecification in Yawelmani Phonology and Morphology. PhD thesis, MIT. • Bailey 1995. • Hayes 1995.

Continued on next page

	Name	#	Sources
413	Yeletnye (Yele)	2	Henderson, J. E. 1975. Yeletnye, the language of Rossel Island. In T. E. Dutton, ed., Studies in languages of central and south-east Papua, (Pacific Linguistics C29, pp. 817-834). Canberra: Australian National Univerisity. • Walker, R. 1996. Prominence-driven stress. Rutgers Optimality Archive (ROA-172-0197). • Bailey 1995.
414	Yiddish	2	Birnbaum, S.A. 1979. Yiddish: a survey and grammer. Toronto. • Fal'kovich, E.M. 1966. Evbreijskij (Idish). In Jazyki Narodov SSSR (Languages of the Soviet Union) 1. Indo-evropejskie jazyki, ed. V.V. Vinogradov et al. Moscow. • Bailey 1995.
415	Yidiɲ	110	Hayes, Bruce. 1981. A metrical theory of stress rules. 1980. Ph.D. thesis, MIT. • Dixon, R.M.W. 1977. Some phonolgical rules of Yidin. LinguisticInquiry 8. 1-34. • Dixon, R.M.W. 1977. A Grammar of Yidin. Cambridge, England. Cambridge University Press. • Bailey 1995.
416	Yil	54	Martens, Mary and Salme Tuominen. 1977. A Tentative Phonemic Statement in Yil in West Sepik District. In Richard Loving, ed. Phonologies of Five Papua New Guinea Languages. Workpapers in Papua New Guinea Languages 19, Summer Institute of Linguistics. Ukarumpa, Papua New Guinea, pp. 29-48. • Bailey 1995. • Hayes 1995.
417	Yingkarta	8	Dench, Alan Charles. 1998. Yingkarta. München: Lincom Europa. • Gordon 2002.
418	Yuchi	1	Ballard, W. L. 1975. Aspects of Yuchi morphonology. In Crawford, James, ed. Studies in Southeastern Indian Languages, pp. 237-250. Athens, Georgia: University of Georgia Press. • Gordon 2002.

Continued on next page

	Name	**#**	**Sources**
419	Yupik, Sirenik	18	Menovshchikov, G. A. 1962. Grammatika iazyka aziatskikh eskimosov. Leningrad:Izdatelstvo Akademii Nauk SSR. • Menovshchikov, G. A. 1975. Iazyk naukanskikh eskimosov. Leningrad: Nauka. • Gordon 2002.
420	Yurok	2	Robins, R. H. 1958. The Yurok language: grammar, texts, lexicon. Berkeley: University of California Press. • Gordon 2002.
421	Zapotec, Mitla	1	Briggs, Elinor. 1961. Mitla Zapotec grammar. Mexico: Instituto Linguístico de Verano. • Gordon 2002.
422	Zazaki	1	Paul, Ludwig. 1998. Zazaki: Grammatik und Versuch einer Dialektologie. Wiesbaden: L. Reichert • Gordon 2002.

.

Bibliography

Abrahamson, A. 1968. Contrastive Distribution of phoneme classes in Içuã Tupi. *Anthropological Linguistics* 10(6):11–21.

Albright, Adam and Bruce Hayes. 2002. Modeling English past tense intuitions with minimal generalization. *SIGPHON 6: Proceedings of the Sixth Meeting of the ACL Special Interest Group in Computational Phonology* :58–69.

Albright, Adam and Bruce Hayes. 2003a. Learning NonLocal Environments. Talk Handout of 77th Annual Meeting of LSA, Atlanta Georgia.

Albright, Adam and Bruce Hayes. 2003b. Rules vs. Analogy in English Past Tenses: A Computational/Experimental Study. *Cognition* 90:119–161.

Albro, Dan. 1998. Evaluation, implementation, and extension of Primitive Optimality Theory. Master's thesis, University of California, Los Angeles.

Albro, Dan. 2005. A Large-Scale, LPM-OT Analysis of Malagasy. Ph.D. thesis, University of California, Los Angeles.

Alderete, John, Adrian Brasoveanua, Nazarre Merchant, Alan Prince, and Bruce Tesar. 2005. Contrast analysis aids in the learning of phonological underlying forms. In *The Proceedings of WCCFL 24*. pages 34–42.

Angluin, Dana. 1980. Inductive inference of formal languages from positive data. *Information Control* 45:117–135.

Angluin, Dana. 1982. Inference of Reversible Languages. *Journal for the Association of Computing Machinery* 29(3):741–765.

Angluin, Dana. 1992. Computational Learning Theory: Survey and Selected Bibliography. In *24th Annual ACM*.

Anthony, M. and N. Biggs. 1992. *Computational Learning Theory*. Cambrisge University Press.

Applegate, R.B. 1972. Ineseño Chumash Grammar. Ph.D. thesis, University of California, Berkeley.

Bailey, T. M. and U. Hahn. 2001. Determinants of Wordlikeness: Phonotactics or Lexical Neighborhoods? *Journal of Memory and Language* :568–591.

Bailey, Todd. 1995. Nonmetrical Constraints on Stress. Ph.D. thesis, University of Minnesota. Ann Arbor, Michigan. Stress System Database available at http://www.cf.ac.uk/psych/ssd/index.html.

Baković, Eric. 2000. Harmony, Dominance and Control. Ph.D. thesis, Rutgers University.

Barkanyi, Zsuzsa. 2007. Blick testing word-initial consonant clusters in Slovak. Ms.

Baum, L. E. 1972. An inequality and associated maximization technique in statistical estimation of probabilistic functions of a Markov process. *Inequalities* 3.

Biermann, A. W. and J. A. Feldman. 1972. On the Synthesis of Finite State Machines from Samples of their Behavior. *IEEE Transactions on Computers* :592–297.

Blum, M. and L. Blum. 1975. Towards a mathematical theory of inductive inference. *Information and Control* 28:125–155.

Blumer, Anselm, A. Ehrenfeucht, David Haussler, and Manfred K. Warmuth. 1989. Learnability and the Vapnik-Chervonenkis dimension. *J. ACM* 36(4):929–965. ISSN 0004-5411.

Boersma, Paul. 1997. How we learn variation, optionality, and probability. *Proceedings of the Institute of Phonetic Sciences* 21. University of Amsterdam.

Boersma, Paul. 1998. *Functional phonology: Formalizing the interactions between articulatory and perceptual drives*. University of Amsterdam. LOT International Series 11. The Hague: Holland. Http://www.fon.hum.uva.nl/paul/diss/.

Boersma, Paul and Bruce Hayes. 2001. Empirical tests of the Gradual Learning Algorithm. *Linguistic Inquiry* 32:45–86.

Brill, Eric. 1995. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics* 21(4):543–566.

Browman, C. and L. Goldstein. 1992. Articulatory Phonology: An Overview. *Phonetica* :155–180.

Byrd, Dani, Fidèle Mpiranya, Sungbok Lee, Celeste DeFreitas, and Rachel Walker. 2006. The Articulation of Consonants in Kinyarwanda's Sibilant Harmony. Handout of poster presented at the meeting of the Acoustical Society of America, Honolulu, Hawaii.

Chiosáin, Maire Ní and Jaye Padgett. 2001. Markedness, segment realization, and locality in spreading. In *Constraints and representations: segmental phonology in Optimality Theory*, edited by Linda Lombardi. Cambridge University Press, Cambridge, pages 118–156. Version of 1997 Report LRC-97-01 from Linguistics Research Center, Santa Cruz, CA.

Chomsky, Noam. 1956a. On Certain Formal Properties of Grammars. *Information and Control* 2:137–167.

Chomsky, Noam. 1956b. Three Models for the Description of Language. *IRE Transactions on Information Theory* IT-2.

Chomsky, Noam. 1957. *Syntactic Structures*. Mouton & Co., Printers, The Hague.

Chomsky, Noam. 1965. *Aspects*. MIT Press.

Chomsky, Noam. 1981. *Lectures on government and binding*. Dordrecht, the Netherlands: Foris.

Chomsky, Noam and Morris Halle. 1968. *The Sound Pattern of English*. Harper & Row.

Christiansen, M., J. Allen, and M. Seidenberg. 1998. Learning to segment speech using multiple cues: A connectionist model. *Language and Cognitive Processes* 13:221–268.

Clark, Alexander and Franck Thollard. 2004. PAC-learnability of probabilistic deterministic finite state automata. *Journal of Machine Learning Research* 5:473–497.

Clark, Robin. 1992. The selection of syntactic knowledge. *Language Acquisition* 2:83–149.

Clements, George and Jay Keyser. 1983. *CV phonology: a generative theory of the syllable*. Cambridge, MA: MIT Press.

Coleman, J. S. and J. Pierrehumbert. 1997a. Stochastic Phonological Grammars and Acceptability. In *Computational Phonology*. Somerset, NJ: Association for Computational Linguistics, pages 49–56. Third Meeting of the ACL Special Interest Group in Computational Phonology.

Coleman, John and Janet Pierrehumbert. 1997b. Stochastic Phonological Grammars and Acceptability. In *Computational Phonology*. Somerset, NJ: Association for Computational Linguistics, pages 49–56. Third Meeting of the ACL Special Interest Group in Computational Phonology.

de la Higuera, Colin. 1997. Characteristic Sets for Polynomial Grammatical Inference. *Machine Learning* 27(2):125–138.

Dehaene, Stanislas, Laurent Cohen, Mariano Sigman, and Fabien Vinckier. 2005. The neural code for written words: a proposal. *Trends in Cognitive Science* 9(7):335–341.

Dresher, Elan. 1999. Charting the Learning Path: Cues to Parameter Setting. *Linguistic Inquiry* 30:27–67.

Dresher, Elan and Jonathan Kaye. 1990. A Computational Learning Model for Metrical Phonology. *Cognition* 34:137–195.

E. Newport, H. Gleitman and L. Gleitman. 1977. Mother, I'd rather do it myself: Some effects and non-effects of maternal speech style. In *Talking to Children: Language Input and Acquisition*, edited by C. Snow and C. A. Ferguson. Cambridge: Cambridge University Press.

Eisner, Jason. 1997a. What Constraints Should OT Allow? Talk handout, Linguistic Society of America, Chicago. Available on the Rutgers Optimality Archive, ROA#204-0797, http://roa.rutgers.edu/.

Eisner, Jason. 1997b. What constraints should OT allow? Talk handout, Linguistic Society of America, Chicago.

Eisner, Jason. 1998. FOOTFORM Decomposed: Using Primitive Constraints in OT.

In *Proceedings of SCIL VIII*, edited by Benjamin Bruening, number 31 in MIT Working Papers in Linguistics. Cambridge, MA, pages 115–143.

Ellison, M. T. 1992. The Machine Learning of Phonological Structure. Ph.D. thesis, University of Western Australia.

Ellison, Mark. 1994a. Phonological Derivation in Optimality Theory. In *COLING 94*, volume 2. pages 1007–1013. Kyoto, Japan.

Ellison, T.M. 1994b. The Iterative Learning of Phonological Constraints. *Computational Linguistics* .

Elman, Jeffrey. 2003. Generalization from Sparse Input. In *Proceedings of the 38th Meeting of the Chicago Linguistics Society*.

Everett, Daniel. 1988. On Metrical Constituent Structure in Pirahã Phonology. *Natural Language and Linguistic Theory* .

Fountain, Amy. 1998. An Optimality Theoretic Account of Navajo Prefixal Syllables. Ph.D. thesis, University of Arizona.

Frank, Robert and Shyam Kapur. 1996. On the use of triggers in parameter setting. *Linguistic Inquiry* 27(623-660).

Frank, Robert and Giorgo Satta. 1998. Optimality Theory and the Generative Complexity of Constraint Violability. *Computational Linguistics* 24(2):307–315.

Friederici, Angela and Jeanine Wessels. 1993. Phonotactic knowledge of word boundaries and its use in infant speech perception. *Perception and Psychophysics* 54:287–295.

Frisch, S., M. Broe, and J. Pierrehumbert. 1995. The Role of Similarity in Phonology: Explaining OCP-Place. In *Proceedings of the 13th International Congress of Phonetic Sciences*. Stockholm.

Frisch, S., N.R. Large, and D.B. Pisoni. 2000. Perception of wordlikeness: Effects of segment probability and length on the processing of nonwords. *Journal of Memory and Language* 42:481–496.

Frisch, S., J. Pierrehumbert, and M. Broe. 2004. Similarity Avoidance and the OCP. *Natural Language and Linguistic Theory* 22:179–228.

Gafos, Adamantios. 1999. *The Articulatory Basis of Locality in Phonology*. New York, Garland.

Gafos, Adamantios. 2002. A Grammar of Gestural Coordination. *Natural Language and Linguistic Theory* 20(2). 269-237.

Gibson, Edward and Kenneth Wexler. 1994. Triggers. *Linguistic Inquiry* .

Gick, Bryan, Douglas Pulleyblank, Fiona Campbell, and Ngessimo Mutaka. 2006. Kinande Vowel Harmony. *Phonology* 23(1):1–20.

Gildea, Daniel and Daniel Jurafsky. 1996. Learning Bias and Phonological-rule Induction. *Association for Computational Linguistics* .

Gillis, Steven, Gert Durieux, and Walter Daelemans. 1995. A computational model of P&P: Dresher & Kaye (1990) revisited. In *Approaches to parameter setting*, edited by Frank Wijnen and Maaike Verrips. Vakgroep Algemene Taalwetenschap, Universiteit van Amsterdam, pages 135–173.

Goedemans, R.W.N., H.G. van der Hulst, and E.A.M. Visch. 1996. *Stress Patterns of the World Part 1: Background*. HIL Publications II. Holland Academic Graphics. The Hague.

Gold, E.M. 1967. Language Identification in the Limit. *Information and Control* 10:447–474.

Goldsmith, J. A. and G.N. Larson. 1990. Local modeling and syllabification. In *Papers from the 26th Regional Meeting of the Chicago Linguistic Society, Volume 2: The Parasession on the Syllable in Phonetics and Phonology*, edited by M. Ziolkowsky, M. Noske, and K. Deaton. Chicago Linguistic Society, pages 129–141.

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

Goldsmith, John. 2006. Information Thoery and Phonology. Slides presented at the 80th Annual LSA in Alburquerque, New Mexico.

Goldwater, Sharon. 2006. Non Parametric Bayesian Models of Language Acquisition. Ph.D. thesis, Brown University.

Goldwater, Sharon and Mark Johnson. 2003. Learning OT constraint rankings using a maximum entropy model. In *Proceedings of the Stockholm Workshop on Variation within Optimality Theory*, edited by Jennifer Spenader, Anders Eriksson, and Osten Dahl. pages 111–120.

Gordon, Matthew. 1999. The "neutral" vowels of Finnish: How neutral are they? *Linguistica Uralica* 35:17–21.

Gordon, Matthew. 2002. A Factorial Typology of Quantity-Insensitive Stress. *Natural Language and Linguistic Theory* 20(3):491–552. Additional appendices available at http://www.linguistics.ucsb.edu/faculty/gordon/pubs.html.

Gordon, Matthew. 2006. *Syllable Weight: Phonetics, Phonology, Typology*. Routledge.

Grainger, J. and C. Whitney. 2004. Does the huamn mnid raed wrods as a wlohe? *Trends in Cognitive Science* 8:58–59.

Grätzer, George. 1979. *Universal Algebra*. Springer Verlag, 2nd edition.

Greenberg, Joseph. 1978. Initial and Final Consonant Sequences. In *Universals of Human Language: Volume 2, Phonology*, edited by Joseph Greenberg. Stanford University Press, pages 243–279.

Greenberg, Joseph and J. J. Jenkins. 1964. Studies in the psychological correlates of the sound system of American English. *Word* 20:157–177.

Gupta, Prahlad and David Touretzky. 1991. What a Perceptron Reveals about Metrical Phonology. In *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*. pages 334–339.

Gupta, Prahlad and David Touretzky. 1994. Connectionist models and linguistic theory: Investigations of stress systems in language. *Cognitive Science* 18(1):1–50.

Häggström, Olle. 2002. *Finite Markov Chains and Algorithmic Applications*. Cambridge University Press. London Mathematical Society Student Texts 52.

Hale, Mark and Charles Reiss. 2000. Substance abuse and dysfunctionalism: Current trends in phonology. *Linguistic Inquiry* 31:157–169.

Halle, Morris and G. N. Clements. 1983. *Problem Book in Phonology*. Cambridge, MA: MIT Press.

Halle, Morris and Jean-Roger Vergnaud. 1987. *An Essay on Stress*. The MIT Press.

Hammond, Michael. 2004. Gradience, phonotactics, and the lexicon in English phonology. *International Journal of English Studies* 4:1–24.

Hansen, Kenneth and L.E. Hansen. 1969. Pintupi Phonology. *Oceanic Linguistics* 8:153–170.

Hansson, Gunnar. 2001. Theoretical and typological issues in consonant harmony. Ph.D. thesis, University of California, Berkeley.

Hansson, Gunnar. 2006. Locality and similarity in phonological agreement. University of Indiana, PhonologyFest 2006. Talk Handout.

Hansson, Gunnar. 2007. Local spreading vs. non-local agreement: reconciling ends and means. GLOW 30, University of Tromsø. Talk Handout.

Hay, Jennifer, Janet B. Pierrehumbert, and Mary Beckman. 2003. Speech perception, well-formedness, and the statistics of the lexicon. In *Papers in Laboratory Phonology VI*, edited by John Local, Richard Ogden, and Rosalind Temple. Cambridge: Cambridge University Press, pages 58–74.

Hayes, Bruce. 1981. A Metrical Theory of Stress Rules. Ph.D. thesis, Massachussetts Institute of Technology. Revised version distribued by Indiana University Linguistics Club, Bloomington, and published by Garland Press, New York 1985.

Hayes, Bruce. 1995. *Metrical Stress Theory*. Chicago University Press.

Hayes, Bruce. 1999. Phonetically-Driven Phonology: The Role of Optimality Theory and Inductive Grounding. In *Functionalism and Formalism in Linguistics, Volume I: General Papers*. John Benjamins, Amsterdam, pages 243–285.

Hayes, Bruce. 2004. Phonological acquisition in Optimality Theory: the early stages. In *Fixing Priorities: Constraints in Phonological Acquisition*, edited by Rene Kager, Joe Pater, and Wim Zonneveld. Cambridge University Press.

Hayes, Bruce, Robert Kirchner, and Donca Steriade, eds. 2004. *Phonetically-Based Phonology*. Cambridge University Press.

Hayes, Bruce and Colin Wilson. To appear. A Maximum Entropy Model of Phonotactics and Phonotactic Learning. Available at http://www.linguistics.ucla.edu/people/wilson/papers.html.

Heinz, Jeffrey. 2006a. Learning Phonotactic Grammars from Surface Forms. In *Proceedings of the 25th West Coast Conference of Formal Linguistics*, edited by Donald Baumer, David Montero, and Michael Scanlon. University of Washington, Seattle.

Heinz, Jeffrey. 2006b. Learning Quantity Insensitive Stress Systems via Local Inference. In *Proceedings of the Eighth Meeting of the ACL Special Interest Group in Computational Phonology at HLT-NAACL.* pages 21–30. New York City, USA.

Hopcroft, John, Rajeev Motwani, and Jeffrey Ullman. 2001. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley.

Hyde, Brett. 2002. A Restrictive Theory of Metrical Stress. *Phonology* 19:313–319.

Hyman, Larry. 1977. On the nature of linguistic stress. In *Studies in stress and accent: Southern California Occasional Papers in Linguistics 4*, edited by Larry Hyman. Dept. of Linguistics, University of Southern California.

Hyman, Larry. 1995. Nasal consonant harmony at a distance: The case of Yaka. *African Linguistics* 24:5–30.

Idsardi, William. 1992. The Computation of Prosody. Ph.D. thesis, MIT.

Jacobs, Haike. 1989. Nonlinear Studies in The Historical Phonology of French. Ph.D. thesis, Katholiek Universiteit te Nijmegen.

Jain, Sanjay, Daniel Osherson, James S. Royer, and Arun Sharma. 1999. *Systems That Learn: An Introduction to Learning Theory (Learning, Development and Conceptual Change).* The MIT Press, 2nd edition.

Jakobson, Roman, C. Gunnar, M. Fant, and Morris Halle. 1952. *Preliminaries to Speech Analysis*. MIT Press.

Jelenik, Frederick. 1997. *Statistical Methods for Speech Recognition*. MIT Press.

Johnson, C. Douglas. 1972. *Formal Aspects of Phonological Description*. The Hague: Mouton.

Johnson, Kent. 2004. Golds Theorem and Cognitive Science. *Philosophy of Science* 71:571–592.

Johnson, Mark. 1993. The Complexity of Inducing a Rule from Data. In *The Proceedings of the Eleventh West Coast Conference of Formal Linguistics*, edited by J. Mead. Stanford Linguistics Association, CSLI Press, pages 289–297.

Jurafsky, Daniel and James Martin. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics*. Prentice-Hall.

Jusczyk, Peter, Anne Cutler, and Nancy Redanz. 1993a. Infants' preference for the predominant stress patterns of English words. *Child Development* 64:675–687.

Jusczyk, Peter, Angela Friederici, Jeanine Wessels, Vigdis Svenkerund, and Ann Marie Jusczyk. 1993b. Infants' sensitivity to the sound patterns of native language words. *Journal of Memory and Language* 32:402–420.

Jusczyk, Peter, Paul Luce, and Jan Charles-Luce. 1994. Infants' sensitivity to phonotactic patterns in the native language. *Journal of Memory and Language* 33:630–645.

Kager, René. 1999. *Optimality Theory*. Cambridge University Press.

Kanazawa, Makoto. 1996. Identification in the Limit of Categorical Grammars. *Journal of Logic, Language, and Information* 5:115–155.

Kaplan, Ronald and Martin Kay. 1981. Phonological Rules and Finite State Transducers. Paper presented at ACL/LSA Conference, New York.

Kaplan, Ronald and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics* 20(3):331–378.

Karttunen, Lauri. 1998. The proper treatment of optimality theory in computational phonology. *Finite-state methods in natural language processing* :1–12.

Kearns, Michael and Umesh Vazirani. 1994. *An Introduction to Computational Learning Theory*. MIT Press.

Kelkar, A. R. 1968. Studies in Hindi-Urdu I: Introduction and Word Phonology. Deccan College, Poona.

Kenstowicz, Michael. 1994. *Phonology in Generative Grammar*. Blackwell Publishers.

Khoussainov, Bakhadyr and Anil Nerode. 2001. *Automata Theory and its Applications*. Birkhäuser.

Klimov, G.A. 2001. Megrelskii Yazyk. In *Yazyki Mira: Kavkazskie Yazyki*, edited by M.E. Alekseev. Moscow: Izdatelstvo Academia, pages 52–58.

Kobele, Gregory. 2006. Generating Copies: An Investigation into Structural Identity in Language and Grammar. Ph.D. thesis, University of California, Los Angeles.

Kontorovich, Leonid, Corinna Cortes, and Mehryar Mohri. 2006. Learning Linearly Separable Languages. In *The 17th International Conference on Algorith-*

mic Learning Theory (ALT 2006), volume 4264 of Lecture Notes in Computer Science. Springer, Heidelberg, Germany, pages 288–303.

Krämer, Martin. 2003. *Vowel Harmony and Correspondence Theory*. Mouton de Gruyter.

Kuznecova, A. N., E. A. Xelimskij, and E. V. Gruŝkina. 1980. Oĉerki po selkupskomu jazyku. Izdatelstvo Moskovskogo Universiteta, Moscow.

Leben, William. 1973. Suprasegmental Phonology. Ph.D. thesis, Massachussetts Institute of Technology.

Lehiste, I. 1970. *Suprasegmentals*. Cambridge: MIT Press.

Lin, Ying. 2002. Probably Approximately Correct Learning of Constraint Ranking. Master's thesis, University of California, Los Angeles.

Lin, Ying. 2005. Learning Features and Segments from Waveforms: A Statistical Model of Early Phonological Acquisition. Ph.D. thesis, University of California, Los Angeles.

Lin, Ying and Jeffrey Mielke. 2007. Discovering manner and place features - What can be learned from acoustic and articulatory data? Talk at the 31st Penn Linguistics Colloquium.

Manning, Christopher and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.

Marcus, Gary. 1993. Negative Evidence in Language Acquisition. *Cognition* 46:53–85.

Martin, Andrew. 2004. The effects of distance on lexical bias: sibilant harmony in Navajo compounds. Masters Thesis. University of California, Los Angeles.

Mays, E., F.J. Damerau, and R.L. Mercer. 1991. Context based spelling correction. *Information Processing and Management* 27(5):517–522.

McCarthy, John. 1979. Formal problems in Semitic phonology and morphology. Ph.D. thesis, Massachussetts Institute of Technology.

McCarthy, John. 1981. A Prosodic Theory of Nonconcatenative Morphology. *Linguistic Inquiry* 12:373–418.

McCarthy, John. 2003. OT constraints are categorical. *Phonology* 20:75–138.

McCarthy, John. To appear. Consonant harmony via correspondence: Evidence from Chumash. In *Papers in Optimality Theory III*, edited by Leah Bateman, Adam Werle, Michael O'Keefe, and Ehren Reilly. Amherst, MA: GLSA.

McCarthy, John and Alan Prince. 1986. Prosodic Morphology. Ms., Department of Linguistics, University of Massachusetts, Amherst, and Program in Linguistics, Brandeis University, Waltham, Mass.

McCarthy, John and Alan Prince. 1993. Prosodic Morphology I: Constraint Interaction and Satisfaction. Technical Report 3, Rutgers University Center for Cognitive Science. Available on Rutgers Optimality Archive, ROA#482-1201. http://roa.rutgers.edu.

Merchant, Nazarre and Bruce Tesar. 2006. Learning underlying forms by searching restricted lexical subspaces. In *The Proceedings of CLS 41*. ROA-811.

Mester, Armin. 1992. The quantitative trochee in Latin. *Natural Language and Linguistics Theory* 12(1):1–61.

Mielke, Jeffrey. 2004. The Emergence of Distinctive Features. Ph.D. thesis, Ohio State University.

Mpiranya, Fidèle and Rachel Walker. 2005. Sibilant harmony in Kinyarwanda and coronal opacity. GLOW 28, University of Geneva. Talk Handout.

Myers, Scott. 1997. Expressing Phonetic Naturalness in Phonology. In *Derivations and Constraints in Phonology*, edited by Iggy Roca. Oxford: Clarendon Press, pages 125–152.

Newell, A., S. Langer, and M. Hickey. 1998. The rôle of natural language processing in alternative and augmentative communication. *Natural Language Engineering* 4(1):1–16.

Niyogi, Partha. 2006. *The Computational Nature of Language Learning and Evolution*. The MIT Press.

Niyogi, Partha and Robert Berwick. 1996. A language learning model for finite parameter spaces. *Cognition* 61:161–193.

Nowak, Martin A., Natalia L. Komarova, and Partha Niyogi. 2002. Computational and evolutionary aspects of language. *Nature* 417:611–617.

Ohala, John J. and Manjari Ohala. 1986. Testing hypotheses regarding the psychological reality of morpheme structure constraints. In *Experimental Phonology*, edited by John J. Ohala and Jeri J. Jaeger. San Diego, CA: Academic Press, pages 239–252.

Onn, F. 1980. Aspects of Malay Phonology and Morphology. Ph.D. thesis, Universiti Kebangsaan Malaysia, Bangi.

Osherson, Daniel, Scott Weinstein, and Michael Stob. 1986. *Systems that Learn*. MIT Press, Cambridge, Massachusetts.

Papadimitriou, Christon. 1993. *Computational Complexity*. Addison Wesley.

Partee, Barbara, Alice ter Meulen, and Robert Wall. 1993. *Mathematical Methods in Linguistics*. Dordrect, Boston, London: Kluwer Academic Publishers.

Pater, Joe. 2004. Exceptions and Optimality Theory: Typology and Learnability. Conference on Redefining Elicitation: Novel Data in Phonological Theory. New York University.

Pater, Joe and Anne Marie Tessier. 2003. Phonotactic Knowledge and the Acquisition of Alternations. In *Proceedings of the 15th International Congress on Phonetic Sciences, Barcelona*, edited by M.J. Solé, D. Recasens, and J. Romero. pages 1777–1180.

Pierrehumbert, Janet. 1994. Syllable structure and word structure: a study of triconsonantal clusters in English. In *Phonological Structure and Phonetic Form: Papers in Laboratory Phonology III*, edited by Patricia Keating. Cambridge University Press, pages 168–188.

Pitt, Leonard. 1989. Inductive inference, dfas and computational complexity. In *Proceedings of the International Workshop on Analogical and Inductive Inference*. Springer-Verlag, pages 18–44. Lecture Notes in Artificial Intelligence (v. 397).

Popper, Karl. 1959. *The Logic of Scientific Discovery*. Basic Books, Inc. New York.

Port, Robert and Adam Leary. 2005. Against Formal Phonology. *Language* 81(4):927–964.

Poser, William. 1982. Phonological Representation and Action-at-a-Distance. In *The Structure of Phonological Representations*, edited by H. van der Hulst and N.R. Smith. Dordrecht: Foris, pages 121–158.

Prince, Alan. 1983. Relating to the Grid. *Linguistic Inquiry* 14(1).

Prince, Alan. 1992. Quantitative Consequences of Rhythmic Organization. *CLS* 26:355–398. Parasession of the Syllable in Phonetics and Phonology.

Prince, Alan and Paul Smolensky. 1993. Optimality Theory: Constraint Interaction in Generative Grammar. Technical Report 2, Rutgers University Center for Cognitive Science.

Prince, Alan and Paul Smolensky. 2004. *Optimality Theory: Constraint Interaction in Generative Grammar*. Blackwell Publishing.

Prince, Alan and Bruce Tesar. 2004. Fixing priorities: constraints in phonological acquisition. In *Fixing Priorities: Constraints in Phonological Acquisition*. Cambridge: Cambridge University Cambridge: Cambride University Press.

Pulleyblank, Douglas and William J. Turkel. 1998. The logical problem of language acquisition in optimality theory. In *Is the best good enough? Optimality and competition in syntax*, edited by P. Barbosa, D. Fox, P. Hagstrom, M. McGinnis, and D. Pesetsky. Cambridge, MA: MIT Press, pages 399–420.

Recasens, D., J.M. Sole, and J. Romero, eds. 2003. *On Neutral Vowels in Hungarian*. Universitat Autonoma de Barcelona, Spain.

Riggle, Jason. 2004. Generation, Recognition, and Learning in Finite State Optimality Theory. Ph.D. thesis, University of California, Los Angeles.

Riggle, Jason. 2006. Using Entropy to Learn OT Grammars From Surface Forms Alone. In *Proceedings of the 25th West Coast Conference of Formal Linguistics*, edited by Donald Baumer, David Montero, and Michael Scanlon. Cascadilla Proceedings Project.

Ringen, Catherine. 1988. *Vowel Harmony: Theoretical Implications*. Garland Publishing, Inc.

Ron, Dana, Yoram Singer, and Naftali Tishby. 1996. The power of amnesia: Learning probabilistic automata with variable memory length. *Machine Learning* 25(2-3):117–149.

Rose, Sharon and Rachel Walker. 2004. A Typology of Consonant Agreement as Correspondence. *Language* 80(3):475–531.

Rumelhart, D. E. and J. L. McClelland. 1986. On Learning the Past Tenses of English Verbs. In *Parallel Distributed Processing, volume 2*, edited by J.L. McClelland and D. E. Rumelhart. Cambridge MA: MIT Press, pages 216–271.

Sapir, Edward and Harry Hojier. 1967. The Phonology and Morphology of the Navaho Language. *University of California Publications* 50.

Schein, Barry and Donca Steriade. 1986. On geminates. *Linguistic Inquiry* :691–744.

Schoonbaert, S. and J. Grainger. 2004. Letter position coding in printed word perception: effects of repeated and transposed letters. *Lang. Cogn. Process.* 19:333–367.

Schuh, Russell. 1978. Bade/Ngizim vowels and syllable structure. *Studies in African Linguistics* 9:247–83.

Schuh, Russell. 1997. Changes in obstruent voicing in Bade/Ngizim. Ms. University of California, Los Angeles.

Shieber, Stuart. 1985. Evidence against the context-freeness of natural language. *Linguistics and Philosophy* 8:333–343.

Shillcock, R., J. Levy, G. Lindsey, P. Cairns, and N. Chater. 1993. Connectionist modelling of phonological space. In *Computational Phonology*, edited by T. M. Ellison and J. M. Scobbie. Centre for Cognitive Science, pages 157–178.

Sipser, Michael. 1997. *Introduction to the Theory of Computation*. PWS Publishing Company.

Stabler, Edward P. 2007. Computational models of language universals: Expressiveness, learnability and consequences. Cornell Symposium on Language Universals.

Steriade, Donca. 2001. Directional Asymmetries in Place Assimilation: A Perceptual Account. In *The role of speech perception in phonology*, edited by Elizabeth Hume and Keith Johnson. Academic Press, pages 219–250.

Tenenbaum, Josh. 1999. A Bayesian Framework for Concept Learning. Ph.D. thesis, MIT.

Tesar, Bruce. 1995. Computational Optimality Theory. Ph.D. thesis, University of Colorado at Boulder.

Tesar, Bruce. 1998. An Interative Strategy for Language Learning. *Lingua* 104:131–145.

Tesar, Bruce and Paul Smolensky. 1998. Learnability in Optimality Theory. *Linguistic Inquiry* 29:229–268.

Tesar, Bruce and Paul Smolensky. 2000. *Learnability in Optimality Theory*. MIT Press.

Tessier, Anne-Michelle. 2006. Stages of OT Phonological Acquisition and Error-Selective Learning. In *Proceedings of the 25th West Coast Conference of Formal Linguistics*, edited by Donald Baumer, David Montero, and Michael Scanlon. Cascadilla Proceedings Project.

Treiman, Rebecca, Brett Kessler, Stephanie Knewasser, Ruth Tincoff, and Margo Bowman. 2000. English speakers sensitivity to phonotactic patterns. In *Papers* *in Laboratory Phonology V: Acquisition and the Lexicon*, edited by Michael B. Broe and Janet Pierrehumbert. Cambridge University Press, pages 269–282.

Tuuk, H. N. Van Der. 1971. *A Grammar of Toba Batak*. The Hague.

Valiant, L.G. 1984. A theory of the learnable. *Communications of the ACM* 27:1134–1142.

Viterbi, Andrew J. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory* .

Vitevitch, Michael S., Paul A. Luce, Jan Charles-Luce, and David Kemmerer. 1997. Phonotactics and syllable stress: implications for the processing of spoken nonsense words. *Language and Speech* 40:47–62.

von Humboldt, Wilhelm. 1836 (1999). *On Language*. Cambridge Texts in the History of Philosophy. Cambridge University Press. Michael Losonsky, ed. Translated by Peter Heath.

Walker, Rachel. 1998. Nasalization, Neutral Segments, and Opacity Effects. Ph.D. thesis, University of California, Santa Cruz.

Walker, Rachel. 2000. Mongolian Stress, Licensing, and Factorial Typology. ROA-172, Rutgers Optimality Archive, http://roa.rutgers.edu/.

Walker, Rachel. 2003. Reinterpreting Transparency in Nasal Harmony. In *The Phonological Spectrum, Part I: Segmental Structure*, edited by Jeroen van de Weijer, Vincent van Heuven, and Harry van der Hulst. Amsterdam: John Benjamins, pages 37–72. Current Issues in Linguistic Theory, No. 233.

Walker, Rachel. 2007. Phonetics and phonology of coronal harmony: The case of Kinyarwanda. Handout of talk given as part of the UCLA Colloquium Series.

Whitney, C. 2001. How the brain encodes the order of letters in a printed word: the SERIOL model and selective literature review. *Psychonomic Bull. Rev.* 8:221–243.

Whitney, C. and R.S. Berndt. 1999. A new model of letter string encoding: simulating right neglect dyslexia. *Prog. Brain Res.* 121:143–163.

Wilson, Colin. 2006a. Learning Phonology With Substantive Bias: An Experimental and Computational Study of Velar Palatalization. *Cognitive Science* 30(5):945–982.

Wilson, Colin. 2006b. The Luce Choice Ranker. Talk handout, UCLA Phonology Seminar.

Yang, Charles. 2000. Knowledge and Learning in Natural Language. Ph.D. thesis, MIT.

Yokomori, Takashi. 2003. Polynomial-time identification of very simple grammars from positive data. *Theoretical Computer Science* 298(1):179–206.

Zuraw, Kie. 2000. Patterned Exceptions in Phonology. Ph.D. thesis, University of California, Los Angeles.