Learning Unbounded Stress Systems via Local Inference

Jeffrey Heinz

University of California, Los Angeles

1. Introduction

Explaining how children infer grammatical rules based on their limited finite experience is central to the hypothesis of universal grammar—that collection of inductive principles which allows learning to happen. Because children and languages are complex and many factors influence acquisition—physiological, sociolinguistic, articulatory, perceptual, phonological, syntactic, semantic—a simpler question is often asked: How could *anything* learn some aspect of language? In this paper, the aspect under investigation is the unbounded stress rules found in the world's languages (described in §1.3). It is useful, however, to factor the problem even further: What contribution can a particular inductive principle—here a particular notion of locality—make to learning such stress rules?

To this end, this paper reveals a previously unnoticed universal property of the unbounded stress systems which I call *neighborhood-distinctness*. This property is interesting because (1) it speaks directly to the notion of locality in phonology and (2) the quantity-insensitive stress systems (Gordon 2002) *have the same property* (Heinz 2006), allowing a unified analysis of these seemingly disparate types of stress systems. This paper shows that the same learner in Heinz (2006), which uses neighborhood-distinctness to make generalizations, succeeds in learning every attested unbounded stress system described by Bailey (1995), in a sense made precise below.

This paper is organized as follows. The remainder of the introduction makes explicit the framework within which this learning study takes place, and describes the class of unbounded stress systems. §2 explains how the grammars of these stress patterns are represented, and defines the property neighborhood-distinctness in terms of this representation. §3 describes the learner in detail and explains why it works. §4 shows what predictions follow from the hypothesis that all stress patterns are neighborhood-distinct. §5 summarizes the main findings and makes some comparisons to other learning models. The appendix summarizes the target patterns and results.

1.1 Locality in Phonology

What contribution can an a priori notion of locality make to learning? It is generally agreed that locality is an important feature of phonological grammars:

> "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." (McCarthy and Prince 1986:1)

Kenstowicz (1994:597) also notes "... the well-established generalization that linguistic rules do not count beyond two ...". Even in the domain of stress it has been thought that the notion of locality is indispensable. In their *Essay on Stress*, Halle and Vergnaud write "... it was felt that phonological processes are essentially local and that all cases of nonlocality should derive from universal properties of rule application" (1987:ix).

Focusing exclusively on the role of locality does not mean other factors are unimportant or irrelevant. The attention given to it here is made only to obtain a clear understanding of the contribution an a priori notion of locality can make to learning.

1.2 Models of Learning in Phonology

Figure 1 makes explicit the learning framework adopted here. In this framework, there is a grammar G which generates a language L. The learner, exposed to a finite sample of the language, hypothesizes a grammar G_2. The learner is thus a function which maps finite samples to grammars. If, for successively larger samples, the language of G_2 is the same as the language of G, then the learner has succeeded—the learner has learned the language L from finitely many observations.[1] A class of languages is said to be learnable iff the learner can learn every language L in this class in this way from finite samples.[2] The interested reader should consult Nowak et al. (2002) for an excellent overview of this framework (and others), and more in-depth treatments are given by Niyogi (2006), Jain et. al. (1999), and Osherson et al. (1986).

This framework is implicitly adopted in many models of phonological grammar acquisition. For example, in Optimality Theory (Prince and Smolensky 1993, 2004), the grammar consists of a total ordering of a a priori constraints which generate a language of input/output pairs. If a sufficient sample of these input/output pairs is given to the Recursive Constraint Demotion (RCD) algorithm, it provably returns the target ranking of constraints (Tesar 1995, Tesar and Smolensky 1998). Similarly, the cue-based learner in Dresher and Kaye (1990) takes the grammars to be a finite set of parameter values. Learning occurs by detecting cues present in the sample to set the correct parameter values. Thus, the framework above is independent of any particular grammatical formalism.

[1] Of course this is obtained whenever $G_2 = G$. Identity of languages is preferable to identity of grammars as a criterion for successful language learning since, for any language, there are (infinitely) many different grammars which can generate it.

[2] This is called *exact identification in the limit* (Gold 1967).

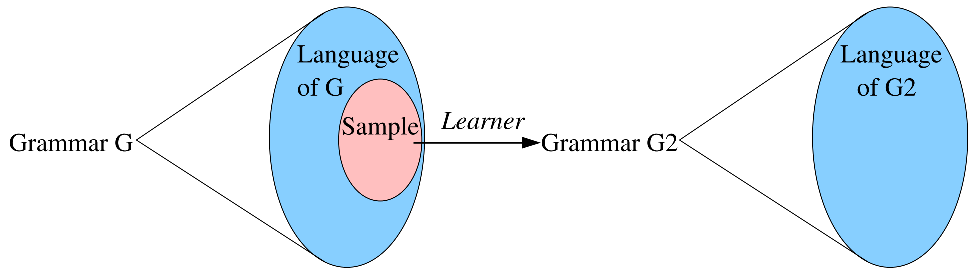

Figure 1: The learning framework.

1.3 Unbounded Stress Systems

Unbounded stress systems have received much attention in phonology (Stowell 1979, Hayes 1981, Prince 1983, Halle and Vergnaud 1987, Idsardi 1992, Hayes 1995, Bailey 1995, Walker 2000, Baković 2004). Two features distinguish unbounded stress systems. First, they are quantity-sensitive, which means the stress rule must distinguish between syllables which are light, and those which heavy. Different languages make the light/heavy distinction in different ways (see Hayes (1995) for an overview), but in this paper I abstract away from the segmental representation of words and represent words with strings of heavy or light syllables.

Second, unbounded stress systems place no limits on the distances between stress and word edges, or between stress and other stresses. In this way they differ from other quantity-sensitive systems which require stress to fall within a certain window of the word edge or of another stress. For example, Kwakwala assigns stress according to the 'Leftmost Heavy Otherwise Rightmost' stress rule, stated below in (1) (Walker 2000).

(1) Place stress on the leftmost heavy syllable in the word. If there are no heavy syllables, stress the rightmost syllable.

The words in (2) reveal the stress pattern in Kwakwala. Stressed syllables can occur in any position—the first, last or a middle syllable—as long as the rule in (1) is obeyed.

(2)

H́	Ĺ	H́ L	H́ H	L H́
L Ĺ	H́ L L	H́ L H	H́ H L	H́ H H
L H́ L	L H́ H	L L Ĺ	L L H́	L H́ L L
L H́ L H	H́ L L L	H́ L L H	H́ H L L	H́ H L H
L H́ H L	L H́ H H	H́ L H L	H́ L H H	H́ H H L
H́ H H H	L L H́ L	L L H́ H	L L L Ĺ	L L L H́

Bailey's (1995) typological survey of primary stress assignment in the world's languages includes 197 languages, of which 44 assign stress in an unbounded manner like the ones above. Table 2 (in the appendix) gives 22 schematic variations found among these 44 types. These variations include whether the left or rightmost heavy is stressed, which syllable is stressed in words without heavy syllables, whether there is a three-way distinction among syllable types (light, heavy, and superheavy), restrictions on the distribution of these syllable types, and secondary stress patterns (if any).

2. Representations

2.1 Finite-state acceptors

I use finite-state acceptors to represent the unbounded stress systems. There are several reasons for this. First, finite-state acceptors are well-defined, simple objects whose basic properties are well-understood (see Hopcroft et. al. (2001) for a good introduction).

Second, it has long been observed that virtually all phonological processes are regular (Johnson 1972, Kaplan and Kay 1981, 1994, Karttunen 1998, Frank and Satta 1998, Albro 2005), meaning that the function which maps underlying forms to surface forms is a finite-state function. It is well known that the range of a finite-state function is a regular set, i.e. describable by a finite-state acceptor. For phonological grammars, the range of this function is properly interpreted as the set of possible well-formed words. Thus it appears that almost all, if not all, phonotactic patterns can be represented as finite-state acceptors.

Finally, it is possible to compute an acceptor which represents well-formed words of a natural language given a phonological grammar made up of finite-state rules (Johnson 1972, Kaplan and Kay 1994). Riggle (2004), building on work in finite-state Optimality Theory (Ellison 1994, Eisner 1997, Albro 1998, Frank and Satta 1998), shows how a finite-state acceptor can be constructed from OT constraints, written as finite state machines. Not only can finite-state descriptions of grammars be obtained from other kinds of descriptions, insights in this domain can be extended if it is determined that more complex types of grammars are needed (e.g. Albro 2005).

As an example, the finite state acceptor in Figure 2 is a representation of the Leftmost Heavy Otherwise Rightmost stress pattern of Kwakwala. In finite-state diagrams in this paper, start states are indicated by triangles, and final states with shading.

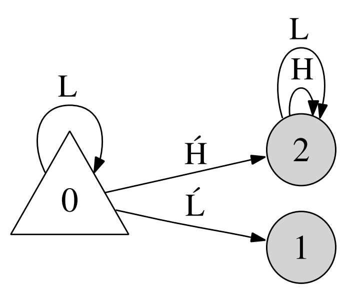

Figure 2: The Leftmost Heavy Otherwise Rightmost (LHOR) stress pattern.

This acceptor meets the minimum requirement for a phonotactic grammar– a device that at least answers Yes or No when asked if some word is possible (Chomsky 1957, Chomsky and Halle 1968, Halle 1978). It is easy to verify that every word this grammar accepts obeys the LHOR stress rule and every word it rejects disobeys it. In fact, this grammar recognizes infinitely many legal words, just like the generative grammars of earlier researchers.

The grammar in Figure 2 can also be related to finite state OT models. Riggle (2004) provides an algorithm which takes constraints, written as finite state machines, and outputs a transducer equivalent to Eval—that is, it maps any underlying form to an output. It is a simple matter to convert this transducer to an acceptor which only accepts words which are legal outputs of the constructed transducer. In fact, if the (different) OT analyses of the LHOR pattern given in Walker (2000) and Baković (2004) were encoded in finite-state OT, Riggles (2004) algorithm yields the (same) acceptor shown above.

Finally, it is possible to build on results in formal learning theory. It is known that learning is impossible if the learner's hypothesis space is too large. For example, the regular languages cannot be learned (Gold 1967, Osherson et. al. 1986, Jain et. al. 1999), though certain nontrivial subsets of this class can be (e.g. Angluin 1982, Kontorovich et. al. 2006). This work provides mechanisms which show how to generalize from finitely many instances of a finite-state function to the function itself. Therefore, it becomes possible to ask what subset of the regular languages delimits the class of unbounded stress patterns.

2.2 Neighborhood-distinctness

Here I describe a universal property of unbounded stress systems which speaks directly to the notion of locality in phonology. I call this property *neighborhood-distinctness*. It is helpful to first introduce the notion of a canonical acceptor for a regular language.

There is only one smallest forward deterministic acceptor for a regular language, called the canonical acceptor.[3] Each state in a canonical acceptor uniquely determines what sequences of symbols may follow the prefixes that lead to that state (Khoussainov and Nerode 2001). The acceptor in Figure 2 is canonical.

Interestingly, the states of the canonical acceptors for every unbounded stress patterns given in Table 2 (see appendix) can also be uniquely defined by locally-determined characteristics; i.e. what transitions go into and out of them, and whether they are start or final states. In other words, each state in the canonical acceptor is uniquely represented by its local environment. I call this environment the *neighborhood*, and it is has four parts.[4]

(3) The neighborhood of a state is defined by
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

I call these acceptors *neighborhood-distinct* because no two states in any of those machines have the same neighborhood. For example, the three states in the Leftmost Heavy Otherwise Rightmost acceptor in Figure 2 have different neighborhoods.

[3]An acceptor is forward deterministic iff it has exactly one start state and for all states q, there are no two transitions bearing the same label departing q.

[4]Readers of (Heinz 2006) may notice the addition of part 4. This addition makes no difference to the learning algorithm presented there, and allows for a simpler statement of the learning algorithm in §3.

Table 1: Neighborhoods of the acceptor in Figure 2.

State	The Neighborhood			
	Incoming Symbols	Outgoing Symbols	Is Final	Is Start
0	{L}	{L,H́,Ĺ}	NO	YES
1	{Ĺ}	∅	YES	NO
2	{H́,L,H}	{H,L}	YES	NO

Because *every* attested unbounded stress pattern has a neighborhood-distinct canonical acceptor, neighborhood-distinctness is a *universal property* of these patterns.

3. The Learner

Neighborhood-distinctness provides an inductive principle a learner can use to infer grammars from surface forms. I describe a simple version of the learner first (the Forward Neighborhood Learner), and then the actual learner used in this study (the Forward Backward Neighborhood Learner). Learning works in two stages. First, a structured representation of the input is built with a 'prefix' tree. Second, generalization occurs by merging states with the same neighborhood.

3.1 Prefix Trees

A prefix tree is a structured representation of a finite sample. The idea is that each state in the tree corresponds to a unique prefix in the sample. Constructing one is a standard algorithm (Angluin 1982). The prefix tree for the words in (2) is shown in Figure 3.

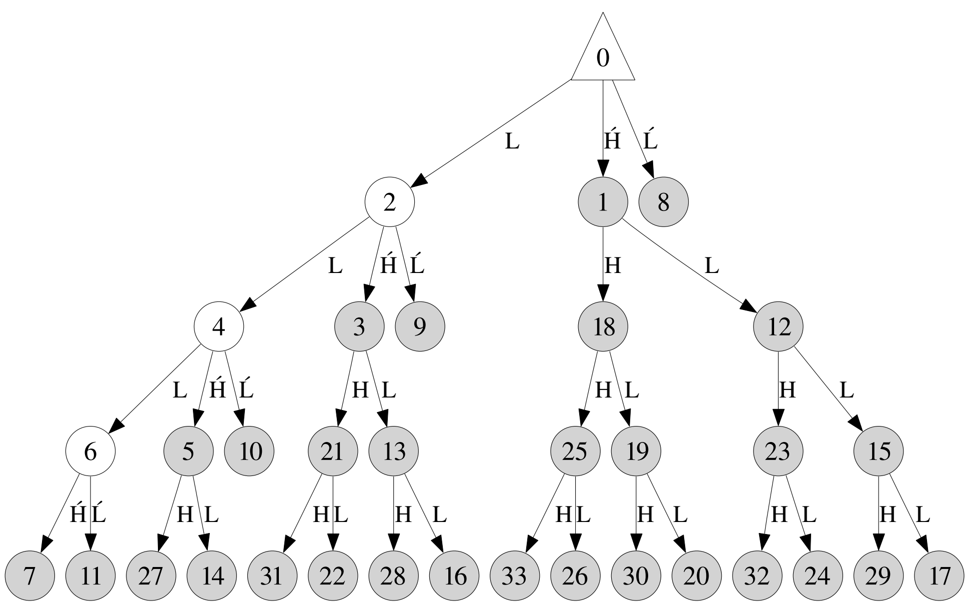

Figure 3: A prefix tree of the LHOR words in (2).

The prefix tree accepts only the finitely many forms that have been observed. I denote the function which maps some finite sample S to a prefix tree which accepts exactly S with PT. Note that $PT(S)$ can be computed efficiently in the size of the sample S (Angluin 1982).

3.2 State Merging

The next stage is to generalize by *state-merging*, a process where two states are identified as equivalent and then *merged* (i.e. combined). A key concept behind state merging is that transitions are preserved (Hopcroft et al. 2001, Angluin 1982). This is how generalizations may occur—because the post-merged machine accepts everything the pre-merged machine accepts, possibly more.

For example, in Figure 4 Machine B is the machine obtained by merging states 1 and 2 in Machine A. It is necessary to preserve 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,* . . . .

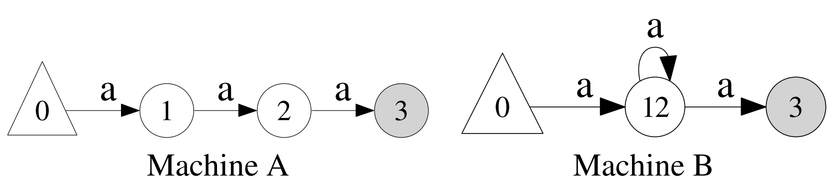

Figure 4: Machine B is obtained by merging states 1 and 2 in Machine A.

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. Furthermore, it has been proven that for any particular acceptor A, and a sufficient sample of words S generated by A, there exists some way to merge the states in the prefix tree of S which returns A (Angluin 1982). At this point the learning question comes into its sharpest focus: What merging strategy exists (if any) which can return acceptors for any unbounded stress pattern from sufficiently large finite samples?

3.3 The Forward Neighborhood Learner

The Forward Neighborhood Learner merges states in the prefix tree with the same neighborhood. If two environments in the prefix tree are locally the same, then they are, as far as the learner can tell, actually the same and are thereby merged.

In the machine in Figure 3 above, for example, states 2 and 4 are merged because they have the same neighborhood. States 2 and 4 are actually different environments in the prefix tree (they are different states), but because the learner is unable to distinguish them on the basis of their neighborhood, they are merged. Note that the order in which states are merged does not matter since state-merging does not alter the neighborhood of any state. Furthermore note that when all states with the same neighborhoods have been merged in the prefix tree, the merging process ends as the resulting acceptor is neighborhood-distinct.

I denote by M_{nd} the function which maps a prefix tree T to the neighborhood-distinct acceptor obtained by merging all states in T with the same neighborhood. Note that computing M_{nd} is efficient in the size of T. 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.[5] It is now possible to state precisely the Forward Neighborhood Learner (FNL).

(4) The Forward Neighborhood Learner

Input: a nonempty sample S.
Ouput: an acceptor A.
1. Let $A = M_{nd}(PT(S))$ and output acceptor A.

Given the sample shown in (2), the FNL returns a machine which accepts exactly those words which obey the Leftmost Heavy Otherwise Rightmost stress pattern. In other words, the learning procedure generalizes exactly as desired.[6]

To summarize, the FNL does two things: (1) it builds a prefix tree of the observed words and (2) merges states in this machine that have the same neighborhood. As shown in the appendix, this learner successfully learns 17 of the 22 systems.[7]

Interestingly, the five patterns the FNL fails to learn— Buriat, Hindi (per Kelkar), Kashmiri, Klamath, and Sindhi— are typically analyzed as having a metrical unit at the right word edge. In each of these five cases, the learner fails by *overgeneralizing*. In other words, the grammar returned by the learner accepts a language strictly larger than the target language because too many states were merged.

3.4 The Forward Backward Neighborhood Learner

This section presents an elaboration of the Forward Neighborhood Learner, which learns every attested unbounded stress system. The Forward Backward Neighborhood Learner uses the same generalization strategy—merging states with the same neighborhood. It differs from the FNL in that it addresses the inherent *left-to-right bias* present in prefix trees. This must be addressed since it is known that stress patterns can be sensitive to the

[5] How efficient depends on the representation of the acceptors (i.e. as matrices or as tuples of sets).

[6] Although the machine obtained by M_{nd} is not necessarily identical to the one shown in Figure 2, it does accept exactly the same language. The acceptor in Figure 2 can be obtained from the machine the learner returns by a process of determinization and minimization. Determinization is known not to be an efficient procedure given any acceptor (Hopcroft et al. 2001), but there are classes of acceptors for which determinization can proceed efficiently. It is a current conjecture that the machines obtained by the algorithm here can be determinized efficiently.

[7] This falsifies the conjecture made in Heinz (2006) that the FNL learns any language whose canonical acceptor is neighborhood-distinct.

left edge, the right edge, or sometimes both (Hayes 1995). Thus the few failures of the Forward Learner are attributed *not* to the generalization strategy but rather to an inherent bias of the (independent) choice of how the input is represented.

If the input were represented with a *suffix tree*, then the structure obtained has the reverse bias, a right-to-left bias. Figure 5 shows a suffix tree constructed from words the words in (2). This representation of the input is not a mirror image of the prefix tree shown in Figure 3. 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.

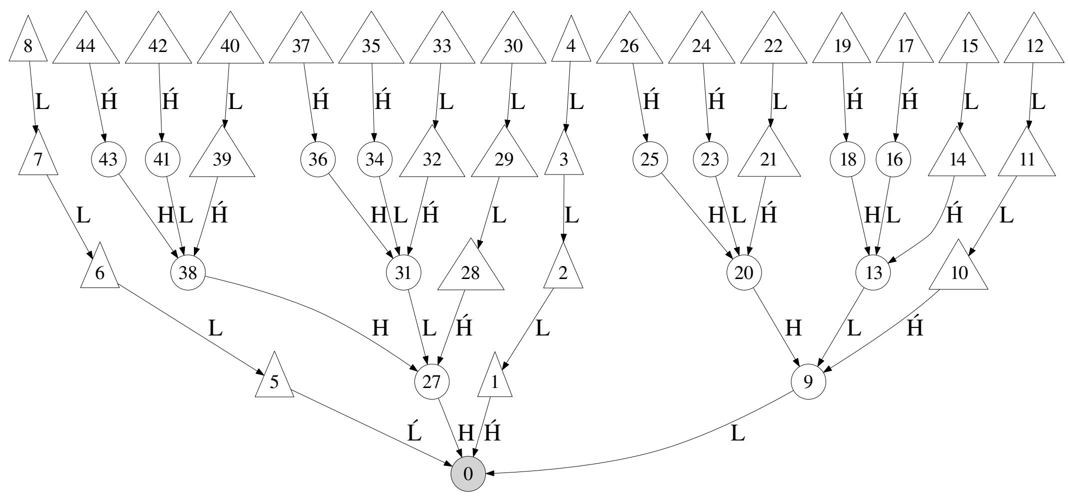

Figure 5: A suffix tree of the LHOR words in (2).

I denote the function which maps some finite sample S to a suffix tree which accepts exactly S with ST. $ST(S)$ can be computed efficiently in the size of the sample S.[8]

(5) The Forward Backward Neighborhood Learner (FBNL)

Input: a nonempty sample S.
Ouput: an acceptor A.

1. Let $A_1 = M_{nd}(PT(S))$.
2. Let $A_2 = M_{nd}(ST(S))$.
3. Let $A = A_1 \cap A_2$ and output the acceptor A.[9]

The FBNL successfully learns every unbounded stress system in Bailey's (1995) database (see appendix). Furthermore, because steps 1 and 2 of the algorithm are efficient

[8]This is because a suffix tree can be computed by reversing each word in the sample, building a prefix tree of this 'reversed' sample, and then reversing the prefix tree. Since each of these operations is efficient (in the size of its input), a suffix tree is efficiently computable as well.

[9]Intersection ($\cap$) of two acceptors A and B results in an acceptor which only accepts words accepted by both A and B (Hopcroft et al. 2001).

in the size of S, and intersecting two machines is efficient, the algorithm in (5) is efficient in the size of S.

The reason the FBNL succeeds is simple: intersection keeps the robust generalizations. The robust generalizations are the ones made in *both* the prefix and suffix trees. For instance, the overgeneralizations that are made by the Forward Neighborhood Learner with Buriat, Hindi (per Kelkar), Kashmiri, Klamath, and Sindhi, are not made by merging same-neighborhood states in the suffix tree. Consequently, these overgeneralizations do not survive the intersection process.[10]

However, the generalization strategy itself—the merging of same-neighborhood states—is the real reason for the algorithm's success. By merging states with the same neighborhood, the algorithm guarantees that its output is neighborhood-distinct, which is a universal property of the canonical acceptors of attested unbounded stress patterns. The necessity of utilizing both prefix and suffix trees appears to be a consequence of a curious fact about stress systems: that they can be sensitive to either word edge, or both.

4. Predictions

Most logically possible stress assignment rules cannot be learned by the Forward Backward Neighborhood Learner. For example, it is relatively straightforward to see that stress patterns describable with feet of size four or larger cannot be learned by the FBNL. This is because in longer words there are sequences of at least three unstressed syllables. Such patterns are not neighborhood-distinct since there are always two states with the same neighborhood in any acceptor for this pattern. In this way, the FBNL is unable to distinguish numbers larger than three from 'more than two'.

More subtle predictions are also made by this learner. It was discovered that if secondary stress is excluded from the grammars of Klamath (Barker 1963, 1964, Hammond 1986, Hayes 1995) and Seneca (Chafe 1977, Stowell 1979, Prince 1983, Hayes 1995), 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. Removing secondary stress causes the patterns to no longer be neighborhood-distinct and hence unlearnable. It is an open question whether humans exhibit similar behavior. To my knowledge, this is not a prediction made in Optimality-Theoretic approaches, nor in Principles and Parameters approaches. If this prediction is correct, it is true that parameters or constraints could be added to the current pantheons in order to make such a prediction. However, without accompanying independent reasons, the explanatory force of such additions is questionable.

Finally, the Forward Backward Neighborhood Learner can learn unattested patterns such as 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 Leftmost Heavy Otherwise Rightmost pattern, the fact is plausibly due to considerations separate from locality (e.g. factors related to syllable weight).

[10] The Backward Neighborhood Learner—which outputs an acceptor by merging same-state neighborhoods in the suffix tree of the sample— only fails to learn the Seneca pattern, which arguably builds iambs from the *left* word edge.

5. Discussion

Figure 6 summarizes the findings of this paper. The attested unbounded stress systems, just like the attested quantity-insensitive stress systems, belong to a small subset of the regular languages, those that are learnable by the Forward Backward Neighborhood Learner. This learner succeeds because it generalizes using a particular notion of locality, ensuring that the grammars obtained are neighborhood-distinct, a property which the attested patterns demonstrably possess. It remains to be determined to what extent the quantity-sensitive bounded patterns, and other phonotactic patterns, belong to this subset as well.

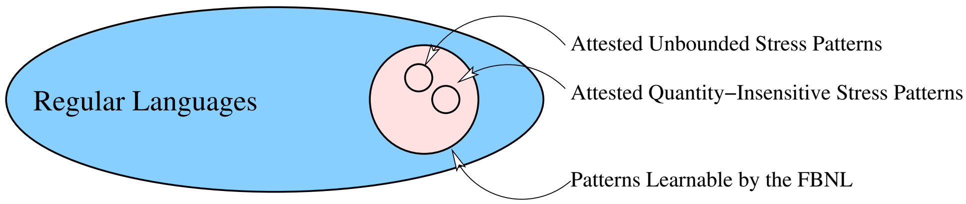

Figure 6: Attested stress patterns and their relation to the FBNL.

Appendix

Table 2 is interpreted as follows. In the FNL, BNL, and FBNL columns, circled numbers mean the Forward Neighborhood Learner, the Backward Neighborhood Learner, and Forward Backward Neighborhood 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) and other relevant stress patterns (such as optional or secondary stress), which are explained in Table 3.

Finally, The Syllable Priority Code (SPC) was developed by Bailey (1995) as a shorthand for indicating primary stress assignment rules. The SPC is read as a series of if/then/else statements. 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. The L or R at the right edge indicates which word edge *Left* or *Right*, respectively, is relevant. Each number indicates a syllable relative to the relevant word edge. Slashes demarcate syllable weight. Since words are unbounded in length, Bailey 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.

Table 2: Types of Unbounded Stress Systems and Results

	SPC	Notes	Example Language	FNL	BNL	FBNL
1.	12..89/1L		Amele	④	⑤	⑤
2.	12..89/1L	A	Murik	④	④	④
3.	12..89/1L	B	Serbo, Croatian	④	④	④
4.	12..89/12..89/1L		Maori	⑤	⑤	⑤
5.	12..78/12..78/1L		Kashmiri	×	⑥	⑥
6.	12..89/2L		Mongolian, Khalkha	⑤	⑤	⑤
7.	12..89/9L		Komi	④	④	④
8.	23..891/9R		Buriat	×	⑤	⑤
9.	23..89/9R	C	Cheremis, Eastern	④	④	④
10.	23..89/9R		Nubian, Dongolese	⑤	⑤	⑤
11.	12..89/9R		Chuvash	④	④	④
12.	1/23..89/9R		Arabic, Classical	④	④	④
13.	12..89/1R		Golin	⑤	④	⑤
14.	12..89/1R	A	Mayan, Aguacatec	④	④	④
15.	23..89/2R	D	Cheremis, Mountain	⑥	⑤	⑥
16.	23..89/2R		Cheremis, Western	⑥	⑤	⑥
17.	23..89@s@w2/0R	E	Seneca	⑦	×	⑦
18.	23..891/2R		Sindhi	×	⑤	⑥
19.	1/23..891/1R		Cheremis, Meadow	⑤	④	⑤
20.	23..891/23..891/2R		Hindi (per Kelkar)	×	⑤	⑥
21.	12..89/23/3R	F	Klamath	×	⑥	⑥
22.	12..89/12..89/12/2R		Mam	⑤	⑤	⑤

Table 3: Notes on Table 2.

Note	
A	At most one heavy syllable per word.
B	At least one heavy syllable per word.
C	Words optionally place stress word-finally.
D	Words with no heavies have lexical stress.
E	According to Chafe (1977), the rightmost nonfinal even-numbered syllable attracts stress if it's either heavy or followed by a heavy syllable. No stress is assigned if there no heavy syllable in the word. If primary stress is assigned, secondary stress is also assigned on even syllables.
F	According to Hammond (1986), secondary stress falls on a heavy (or superheavy) penult if there is a superheavy syllable to its left

References

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. Doctoral Dissertation, University of California, Los Angeles.

Angluin, Dana. 1982. Inference of reversible languages. *Journal for the Association of Computing Machinery* 29:741–765.

Bailey, Todd. 1995. Nonmetrical constraints on stress. Doctoral Dissertation, University of Minnesota. Ann Arbor, Michigan. Stress System Database available at http://www.cf.ac.uk/psych/ssd/index.html.

Baković, Eric. 2004. Unbounded stress and factorial typology. In *Optimality theory in phonology: A reader*, ed. John McCarthy. Blackwell, London. ROA-244.

Barker, Muhammad. 1963. *Klamath dictionary*, volume 31 of *University of California Publications in Linguistics*. University of California Press, Berkeley.

Barker, Muhammad. 1964. *Klamath grammar*, volume 32 of *University of California Publications in Linguistics*. University of California Press, Berkeley.

Chafe, Wallace. 1977. Accent and related phenomena in the five nations iroquois languages. In *Studies in stress and accent*, ed. Larry Hyman, volume 4 of *Southern California Occasional Papers in Lingustics*, 169–181. Department of Linguistics, University of Southern California.

Chomsky, Noam. 1957. *Syntactic structures*. Mouton & Co., Printers, The Hague.

Chomsky, Noam, and Morris Halle. 1968. *The sound pattern of english*. Harper & Row.

Dresher, Elan, and Jonathan Kaye. 1990. A computational learning model for metrical phonology. *Cognition* 34:137–195.

Eisner, Jason. 1997. What constraints should ot allow? Talk handout, Linguistic Society of America, Chicago. ROA#204-0797.

Ellison, Mark. 1994. Phonological derivation in optimality theory. In *COLING 94*, volume 2, 1007–1013. Kyoto, Japan.

Frank, Robert, and Giorgo Satta. 1998. Optimality theory and the generative complexity of constraint violability. *Computational Linguistics* 24:307–315.

Gold, E.M. 1967. Language identification in the limit. *Information and Control* 10:447–474.

Gordon, Matthew. 2002. A factorial typology of quantity-insensitive stress. *Natural Language and Linguistic Theory* 20:491–552. Additional appendices available at http://www.linguistics.ucsb.edu/faculty/gordon/pubs.html.

Halle, Morris. 1978. Knowledge unlearned and untaught: What speakers know about the sounds of their language. In *Linguistic theory and psychological reality*. The MIT Press.

Halle, Morris, and Jean-Roger Vergnaud. 1987. *An essay on stress*. The MIT Press.

Hammond, Michael. 1986. The obligatory branching parameter in metrical theory. *Natural Language and Linguistic Theory* 4:185–228.

Hayes, Bruce. 1981. A metrical theory of stress rules. Doctoral Dissertation, 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.

Heinz, Jeffrey. 2006. 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*, 21–30. New York City, USA.

Hopcroft, John, Rajeev Motwani, and Jeffrey Ullman. 2001. *Introduction to automata* *theory, languages, and computation*. Addison-Wesley.

Idsardi, William. 1992. The computation of prosody. Doctoral Dissertation, MIT.

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.

Johnson, C. Douglas. 1972. *Formal aspects of phonological description*. The Hague: Mouton.

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* 20:331–378.

Karttunen, Lauri. 1998. The proper treatment of optimality theory in computational phonology. *Finite-state methods in natural language processing* 1–12.

Kenstowicz, Michael. 1994. *Phonology in generative grammar*. Blackwell Publishers.

Khoussainov, Bakhadyr, and Anil Nerode. 2001. *Automata theory and its applications*. Birkhäuser.

Kontorovich, Leonid, Corinna Cortes, and Mehryar Mohri. 2006. Learning linearly separable languages. In *The 17th International Conference on Algorithmic Learning Theory (ALT 2006)*, volume 4264 of Lecture Notes in Computer Science, 288–303. Springer, Heidelberg, Germany.

McCarthy, John, and Alan Prince. 1986. Prosodic morphology. Ms., Department of Linguistics, University of Massachusetts, Amherst, and Program in Linguistics, Brandeis University, Waltham, Mass.

Niyogi, Partha. 2006. *The computational nature of language learning and evolution*. The MIT Press.

Nowak, Martin A., Natalia L. Komarova, and Partha Niyogi. 2002. Computational and evolutionary aspects of language. *Nature* 417:611–617.

Osherson, Daniel, Scott Weinstein, and Michael Stob. 1986. *Systems that learn*. Cambridge, Massachusetts: MIT Press.

Prince, Alan. 1983. Relating to the grid. *Linguistic Inquiry* 14.

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.

Riggle, Jason. 2004. Generation, recognition, and learning in finite state optimality theory. Doctoral Dissertation, University of California, Los Angeles.

Stowell, T. 1979. Stress patterns of the world, unite! In *Mit working papers in linguistics*. Department of Linguistics, MIT, Cambridge, MA.

Tesar, Bruce. 1995. Computational optimality theory. Doctoral Dissertation, University of Colorado at Boulder.

Tesar, Bruce, and Paul Smolensky. 1998. Learnability in optimality theory. *Linguistic Inquiry* 29:229–268.

Walker, Rachel. 2000. Mongolian stress, licensing, and factorial typology. ROA-172.