University of Pennsylvania Working Papers in Linguistics

Volume 20
Issue 1 *Proceedings of the 37th Annual Penn Linguistics Conference*

Article 5

1-1-2014

Learning Local Phonological Processes

Jane Chandlee
University of Delaware, janemc@udel.edu

Cesar Koirala
University of Delaware, koirala@udel.edu

Learning Local Phonological Processes

Abstract
We present a learning algorithm for local phonological processes that relies on a restriction on the expressive power needed to compute phonological patterns that apply locally. Representing phonological processes as a functional mapping from an input to output form (an assumption compatible with either the SPE or OT formalism), the learner assumes the target process can be described with the functional counterpart to the Strictly Local (McNaughton and Papert 1971, Rogers and Pullum 2011) formal languages. Given a data set of input-output string pairs, the learner applies the two-stage grammatical induction procedure of 1) constructing a prefix tree representation of the input and 2) generalizing the pattern to words not found in the data set by merging states (Garcia and Vidal 1990, Oncina et al. 1993, Heinz 2007, 2009, de la Higuera 2010). The learner's criterion for state merging enforces a locality requirement on the kind of function it can converge to and thereby directly reflects its own hypothesis space. We demonstrate with the example of German final devoicing, using a corpus of string pairs derived from the CELEX2 lemma corpus. The implications of our results include a proposal for how humans generalize to learn phonological patterns and a consequent explanation for why local phonological patterns have this property.

Learning Local Phonological Processes

Jane Chandlee and Cesar Koirala*

1 Introduction

This paper presents a learning algorithm for local phonological processes that exploits a restriction on the expressive power needed to compute phonological patterns that apply locally. This restriction reflects the well-recognized idea that locality plays an important role in phonology (Kenstowicz 1994, Gafos 1996) and arguably in learning (Heinz 2009). We first characterize phonological processes as functions mapping underlying to surface forms and then demonstrate how a learner with a locality bias can generalize such a function from a finite set of underlying-surface pairs.

This approach to the learning problem recognizes the utility of restricting the hypothesis space of a learner. The formal grammars needed to analyze phonological patterns are less powerful than those needed for syntactic patterns (Heinz and Idsardi 2011). It has been known since Johnson (1972), Koskenniemi (1983), and Kaplan and Kay (1994) that phonological rules of the form A $\Rightarrow$ B / C $\_\_$ D, where A, B, C, and D are regular expressions, can be described with regular relations, making them more restricted than the context-free and context-sensitive patterns found in syntax (Chomsky 1956, Schieber 1985). The class of regular relations, however, does not appear to be learnable in the limit from positive data (Gold 1967). In truth, however, learning the entire class of regular relations is not desirable from the phonological standpoint, since this class includes many patterns that are unattested and unexpected in natural language. Thus we propose a further restriction on the computational bound of a local phonological process, one that rules out many unexpected patterns and defines a subregular class of functions that is identifiable in the limit. The learning results presented here propose such a class, one that is delimited by a requirement of locality.

There is a large and growing body of literature on learning in Optimality Theory (Tesar and Smolensky 1993, 1998, Tesar 1995, 1998, Boersma 1997, Boersma and Hayes 2001, Pater and Tessier 2003, Hayes 2004, Riggle 2004, 2006, Pater 2004, Prince and Tesar 2004, Alderete et al. 2005, Merchant and Tesar 2008, Magri 2010, 2012), with relatively less attention being paid to the learning of phonological rules (Johnson 1984, Gildea and Jurafsky 1996, Albright and Hayes 2002, 2003). The learner we present targets a functional mapping from an input to output form, an abstract characterization of phonological processes that is compatible with either the SPE or OT formalism. The findings then contribute to our understanding of how phonological rules can be learned, but also suggests a inductive principle that could be incorporated into a variety of learning frameworks.

The organization of the paper is as follows. Section 2 introduces the computational framework we are assuming, in which phonological processes are represented as functions that map input ('underlying') to output ('surface') forms. Section 3 formalizes the notion of locality and presents a two-step algorithm that uses the locality assumption to learn a particular class of formal languages. Section 4 adapts this algorithm to the functional domain in order to learn phonological processes, and section 5 demonstrates the algorithm with the example of German final devoicing. Section 6 discusses the results of this demonstration, and section 7 concludes and identifies future directions of this research.

2 Phonological Processes as Mappings

In an SPE-style grammar, phonological processes are expressed with a rewrite rule such as in (1).

(1) +voice, $-$son $\Rightarrow$ $-$voice / $\_\_$#

This rule describes the process of final obstruent devoicing, attested in German and other languages, by which voiced obstruents at the ends of words surface as their voiceless counterparts. Alternatively, the devoicing phenomenon could be captured with an OT-style analysis of ranked

*We thank Jeffrey Heinz, Irene Vogel, and James Rogers for valuable feedback and support.

constraints, such that a constraint against word-final voiced obstruents outranks faithfulness to voicing, as in (2).

(2) *[+voice, −son]# >> IDENT(voice)

Both of these grammatical formalisms in fact converge on the same *mapping* of underlying to surface forms. In other words, both (1) and (2) would map an input form such as */ba:d/*, 'bath', to *[ba:t]*, and the form */za:g/*, 'say', to *[za:k]*. Thus, abstracting away from a particular formalism, phonological processes can be represented as pairs of underlying and surface forms: (*ba:d*, *ba:t*), (*za:g*, *za:k*), (*alt*, *alt*), etc.

Our approach is to represent such mappings as functions encoded as finite state transducers (FSTs), such as in Figure 1. In the figure, the symbols N, V, T, and D represent nasal, vowel, voiceless obstruent, and voiced obstruent, respectively. The FST reads an input form one symbol at a time and produces an output. At any given time the FST is in one of its two *states*; the labeled arrows represent transitions between the states that are taken if the input matches the symbol on the lefthand side of the label (the symbol on the righthand side is the output).[1]

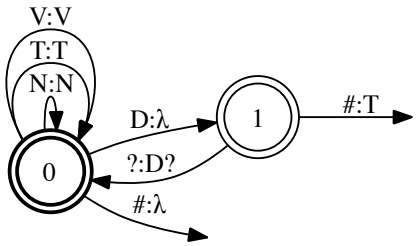

Figure 1: Finite state transducer for final devoicing.

Starting in state 0, any input symbol except for a voiced obstruent is unchanged in the output and the FST remains in state 0. When it does read a voiced obstruent, it transitions to state 1 and outputs λ, the empty string. This temporary withholding of output is necessary because the voiced obstruent may need to be 'devoiced', but the FST must verify that it is in fact a *word-final* obstruent before it does so. If this turns out to be the case (i.e. if the next input symbol is the word-final marker #), the voiceless obstruent is outputted. If the next input symbol is anything else (represented with the symbol ? in the figure), then the voiced obstruent is outputted unchanged and the FST returns to state 0. Figure 2 below shows an example derivation for the input string VDVND. The FST in Figure 1 will in fact be the target of the phonological learner we present in this paper, since a learner that can generalize this FST from input data will have successfully learned the mapping that underlies the final devoicing process.

Input:		V	D	V	N	D	
State:	0	$\mapsto 0$	$\mapsto 1$	$\mapsto 0$	$\mapsto 0$	$\mapsto 1$	
Output:		V	λ	DV	N	λ	T

Figure 2: Example derivation for FST in Figure 1.

[1] In this and all FSTs in the paper, the state marked in bold is the initial state, where the FST is before reading any input, and the states marked with double circles are accepting states, the states where the FST must be when it reaches the end of the input in order for the input form to be valid.

3 Locality in Phonology

The learning algorithm we will present uses an assumption of locality as the inductive principle for generalizing an infinite function from finite data. What this means is that of all the logically possible functions that are consistent with the data, the learner will only consider those that are local in a way we will make more precise below. Functions that do not meet this criterion of locality will not even be considered by the learner. Though locality has been articulated in various ways to accommodate a range of linguistic phenomena (e.g., non-contiguous segments may be considered adjacent when projected onto a tier), for our purposes the target of a rule and its triggering contexts must form a contiguous substring in the input form in order to be captured with a local function. This conception of locality of course rules out certain phonological processes, most notably consonant harmony (Hansson 2001, 2007, Rose and Walker 2004), and it does follow that such nonlocal processes fall outside of the scope of our learner. There is potential, however, to extend our methodology to the nonlocal domain (see Section 6 for discussion).

3.1 Strictly Local Languages

As stated above, the notion of locality that will serve as the inductive principle of the learner is based on contiguous segments. In the realm of phonotactics, this notion has been formalized with the class of formal languages known as Strictly Local languages (McNaughton and Papert 1971, Rogers and Pullum 2011), which are languages for which string membership can be determined by examining the substrings of length k (called k-factors). For example, consider the well-attested phonotactic restriction that requires nasal-stop clusters to be homorganic. This constraint can be restated as, 'the following 2-factors are not permitted in any string in the language: nb, nk, mt, ŋp, etc.' Given a string like [θɪŋk], we can determine whether it is part of the language (i.e., whether or not it violates the constraint) by scanning its 2-factors and verifying that none of them are in the set of illegal ones. In this case, all of the 2-factors (#θ, θɪ, ɪŋ, ŋk, k#, where # is the word-boundary) are permitted, so the string is in the language. On the other hand, the string [θɪnk], is not, because one of its 2-factors (nk) is not permitted.

The 'strict locality' of these languages means that no information beyond the k-factors can be used to determine string membership. This same restriction is the basis for an approach to learning them (García and Vidal 1990, Heinz 2010). As our own learning algorithm is an adaptation of this approach to the functional domain, we will first describe this method for learning SL languages and then our adaptation to learning functions.

3.2 Learning SL Languages

Learning SL languages was first discussed by García and Vidal (1990). The learning algorithm presented here takes as input a set of strings that are in the target language and generalizes a finite state acceptor that accepts all and only the strings of the language. Its first step is to construct a representation of the input data called a prefix tree, which is a finite state acceptor in which the states correspond to prefixes of the strings in the data set. Note that the term *prefix* is not being used in the morphological sense; here a prefix is any initial portion of the string. As an example, Figure 3 is a prefix tree representation of the data set in (3).

(3) $\{$θɪn, θɪŋk, θaeŋk, θʌm, nʌmbɚ$\}$

The prefix tree in Figure 3 accepts *only* the five words in the data set that was used to construct it. In order to learn the target language this dataset is a sample from, the learner has to somehow generalize from the prefix tree. Its method of generalization is *state merging*, a strategy by which certain states that meet a predetermined criterion are merged while their respective transitions are preserved. State merging is a way of discarding irrelevant information and gradually paring the (maximally informative) prefix tree down to a FSA that only keeps track of the information necessary to distinguish legal from illegal strings in the target language. A trivial example demonstrates. The

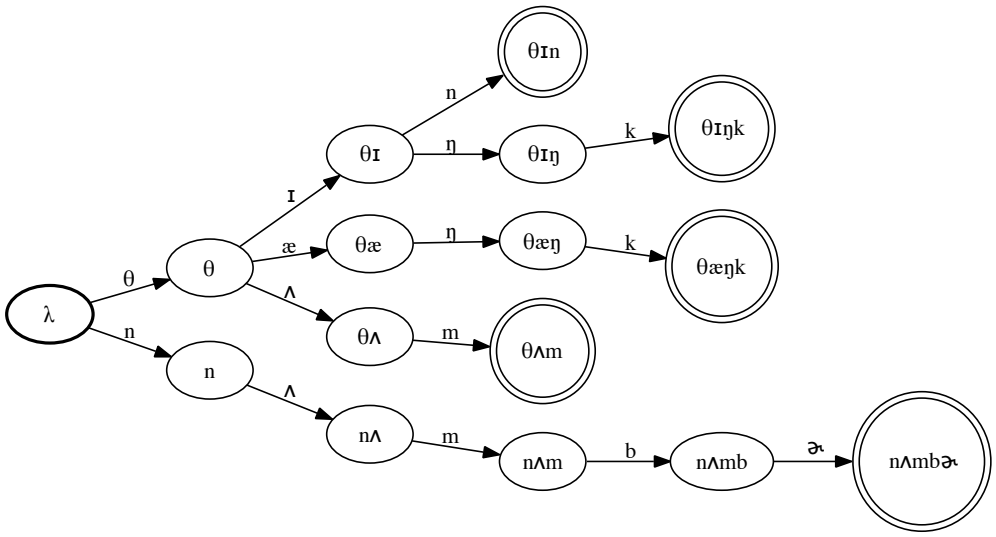

Figure 3: Prefix tree for the data in (3).

FSA on the left of Figure 4 accepts exactly one string: *aa*. Merging all states with the same incoming transition gives the FSA on the right. This FSA accepts all strings of one or more *a*'s.

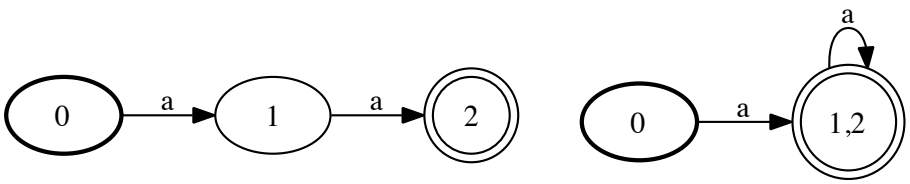

Figure 4: State merging demonstration.

The merging criterion in this example, to merge all states with the same incoming transition, determines the language that is learned. If all strings of one or more *a*'s is indeed the target language, then the merging criterion is a useful one. If the target language is something else, the learner will not be able to learn it. The merging strategy of the SL language learner is to merge states with the same suffix of length $k-1$, with k being the length of the k-factors that define the target language. Recall that the states of the prefix tree are prefixes, which are just strings, so to say two states have the same suffix of length $k-1$ simply means the previous $k-1$ transitions leading into the states have the same symbols. Returning to the homorganic nasal-stop cluster example, the k-value of this phonotactic constraint is 2 (since the illegal factors are of length 2). Applying the state merging strategy to the prefix tree in Figure 3 gives the FSA in Figure 5 (the states have been relabeled with their $k-1$ suffix to clarify the merging that took place).

The acceptor in Figure 5 accepts strings not found in the data set. For example, it includes the hypothetical word θʌmbɚ, which, importantly, obeys the requirement of homorganic nasal-stop sequences. Note also that according to the FSA, all occurrences of ŋ must be followed by a *k*. Though this is not true of English (e.g., *thing, ring, rang*), it is true according to the input data set. This 'error' on the learner's part reflects the fact that it has not received enough data to make the correct generalization.

This section has shown how a two-stage learning algorithm (step one, build a prefix tree; step two, merge states) that assumes its target obeys strictly locality can learn SL languages. For natural

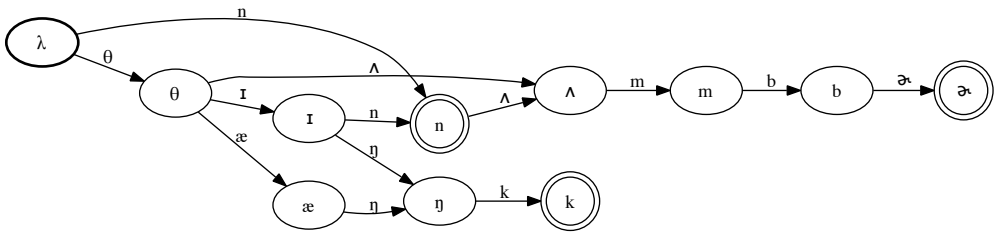

Figure 5: Merging states in Figure 3 with the same $k-1$ suffix.

language, this general strategy (with a different state merging criterion) was applied to the learning of stress patterns in Heinz (2009). For more information on prefix trees and state merging, see de la Higuera (2010). In the next section we adapt this algorithm to the problem of learning local phonological processes.

4 Learning Local Phonological Processes

The learner for SL languages began with a finite state acceptor, but the learner for functions first constructs a finite state *transducer*.[2] The input will be a set of pairs of underlying-surface strings, some of which have undergone the phonological process being learned. Continuing with the example in the previous section, assume the target of the learner is the rule in (4), which assimilates an alveolar nasal to a following velar stop.

(4) $\text{n} \Rightarrow \text{ŋ} \;/\; \_\_\{\text{k, g}\}$

The input data to the learner would include pairs like (θɪnk, θɪŋk), (θaenk, θaeŋk), etc. Our learner's first step is to construct a prefix tree *transducer* of this input data, as shown in Figure 6.

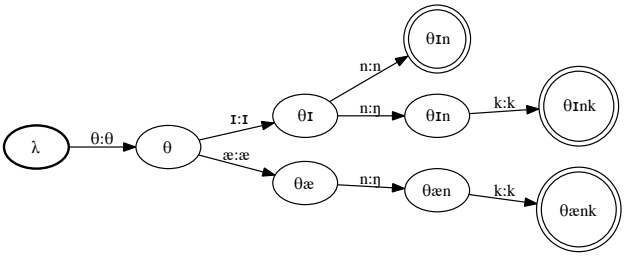

Figure 6: Prefix tree transducer.

The learner's state merging strategy is again to merge states with the same $k-1$ suffix, this time looking only at the *input* side of the transitions leading into a state. Note in Figure 6 that the n:n transition for *thin* and the n:ŋ transition for *think* both lead to a state labeled θɪn. This is because the same input symbols were read in for both of these words; the difference is only that in the latter case the *n* becomes ŋ on the output side of the transition. The learner ignores this difference and merges the states, as well as the state labeled θaen.

Recall again the simple 2-state transducer in Figure 1 that models final devoicing. The reason only two states are needed to capture this process is because the only relevant information is whether the previous symbol read in was a voiced obstruent (state 0 means 'no', state 1 means 'yes'). In the case of Figure 6, it does not matter what input preceded the potential target of the assimilation rule,

[2]Though FSTs can be thought of as acceptors of *pairs* of strings, rather than individual strings.

n, so this information can be discarded. All that matters to determine whether that *n* should become a ŋ is whether the very next symbol is a velar stop.

This section presented the two-stage learning algorithm (modified from the algorithm for learning SL languages) for local phonological processes and illustrated with a trivial example. In the next section we present the results of running the learner on an actual test case with input data from a natural language corpus.

5 Demonstration

As a test for the learner, we gave it a data set of string pairs that reflect the process of final devoicing described above. This data set was derived from the CELEX2 German lemma corpus (Baayen et al. 1996) as follows. Words with a word-final voiced obstruent in the orthography and a word-final voiceless obstruent in the phonetic transcription were assumed to have undergone the process of final devoicing. An input-output pair was constructed for these words that reflected this change. For all other words, a pair was constructed in which the input and output forms are identical. In addition, to make the FST outputted by the leaner more readable, these input-output pairs were recoded with the collapsed alphabet mentioned above and repeated in (5).

(5) D = voiced obstruent
T = voiceless obstruent
N = nasal
V = vowel

The corpus with the full alphabet contains 51,723 pairs; with the collapsed alphabet it contains 18,936 pairs. Given this corpus as input and set with a *k*-value of 2 for final devoicing, the learner converged on the FST shown in Figure 7.

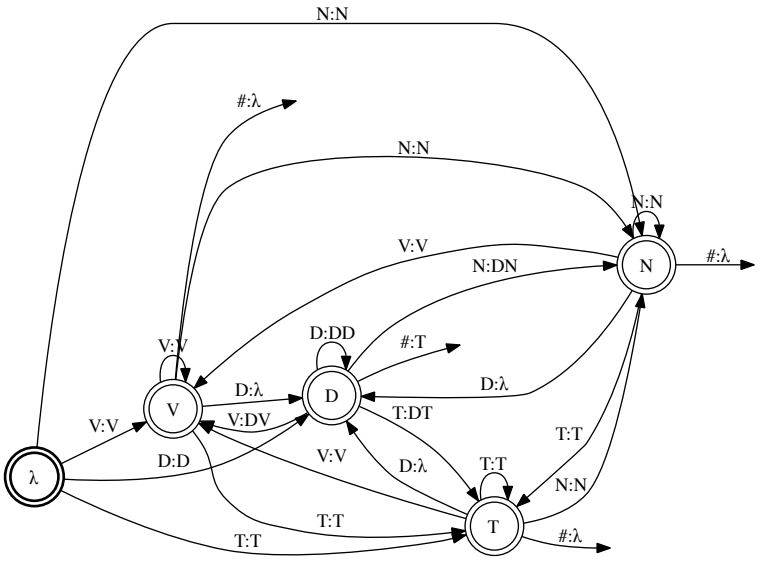

Figure 7: Result with corpus that reflects final devoicing.

The resulting FST has more states than the target (shown in Figure 1), but it describes the same mapping. The larger number of states is in fact entirely expected. Since the learner only merges states that share a suffix of length $k-1$, the smallest FST it will produce will have at least one state per $k-1$ factor. In this example, since $k=2$, the resulting machine, as predicted, has one state for each symbol in the alphabet. In the FST in Figure 1, the N, V, T, and initial state are combined,

because these states have one more thing in common: the FST can only be in them if the previous symbol was not D. Thus state 1 in Figure 1 corresponds to state D in Figure 7 and state 0 in Figure 1 corresponds to the set of states $\{\lambda, \text{V}, \text{T}, \text{N}\}$ in Figure 7.

6 Discussion

The demonstration with final devoicing reveals how a learner with a 'locality bias' can converge on a FST that describes a phonological process that meets the condition of locality. This approach to the learning problem is based on a restriction on the computational complexity of local phonological rules. As stated in the introduction, phonological rewrite rules have previously been shown to be restricted to the class of regular relations, which distinguishes them from the more complex syntactic rules. This restriction, however, does not bear directly on the learning problem, since the regular relations do not appear to be identifiable in the limit from positive data (Gold 1967). It has also been demonstrated that many phonological processes fall into the subsequential class of functions (Mohri 1997, Chandlee et al. 2012, Gainor et al. 2012, Chandlee and Heinz 2012, Luo 2013, Payne 2013, Heinz and Lai 2013), which is a proper subset of the regular relations (Mohri 1997). The subsequential functions *are* identifiable in the limit (Oncina et al. 1993), but it was shown by Gildea and Jurafsky (1996) that an algorithm for the class of subsequential functions cannot learn phonological rules given the data available in natural language corpora.

Gildea and Jurafsky (1996) tested the algorithm called OSTIA (Oncina et al. 1993), which was proven to learn the class of subsequential functions. To succeed, OSTIA needs to see a certain set of data points (called a characteristic sample) that will allow it to make the necessary distinctions among the data and generalize to the correct FST. Some of these data points, however, will never occur in a natural language corpus, which is what Gildea and Jurafsky (1996) discovered when they tested OSTIA on the English flapping rule and found it to be unsuccessful. They went on to modify OSTIA with three learning heuristics (faithfulness, community, and context) and found that their modified algorithm successfully learned the flapping rule, among others. The goal of the current work was likewise to facilitate the learner's ability to converge on the target given the available data, though our approach differs from that of Gildea and Jurafsky (1996). Their community heuristic allowed OSTIA to generalize over the natural classes present in the corpus's alphabet, an issue we instead addressed with the collapsed alphabet of $\{\text{T}, \text{D}, \text{N}, \text{V}\}$. And their notion of context is encoded directly in our learner's state merging strategy of only merging states that share a $k-1$ length suffix.

Our approach thus narrows the learner's hypothesis space so that the available data is sufficient for it to converge on the target function and furthermore allows us to be more definitive about exactly what class of functions the learner can learn. If the learner assumes its target is not just any subsequential function, but one that only requires paying attention to the previous $k-1$ symbols, it automatically rules out a number of candidate functions and may be able to converge on the correct one using the available data. Figure 8 illustrates the contribution of locality.

As noted above, the locality requirement that the learner assumes is not in fact met by all phonological processes. Long-distance phenomena like vowel harmony (with transparent vowels), consonant harmony, nonlocal partial reduplication (Riggle 2003), and dissimilation cannot be modeled with the kind of functions targeted by our learner. However, these processes still appear to be subsequential (Chandlee and Heinz 2012, Heinz and Lai 2013, Luo 2013, Payne 2013), meaning they are still more restrictive than regular. There are in fact other subregular languages that can model the long-distance phonotactics behind these long-distance processes (Heinz 2010). We suspect that our methodology can thus be extended to the learning of the (as yet undefined) functional counterparts to these other subregular languages.

7 Conclusion and Next Steps

This paper has demonstrated that the property of locality (which is inherent to many phonological mappings regardless of the formalism used to describe them) can serve as a useful learning bias, and we demonstrated how a learning algorithm with this bias can generalize a phonological mapping

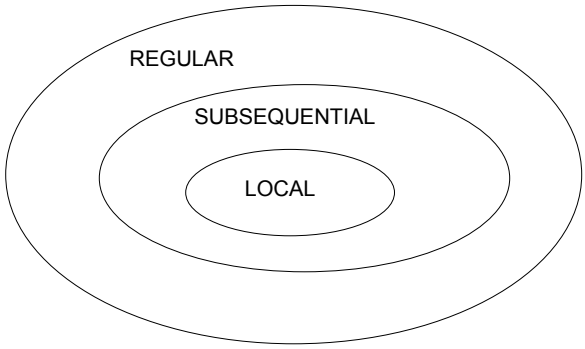

Figure 8: Narrowing the hypothesis space with the locality assumption.

from positive data. This utility of locality in the learning domain suggests that this is not simply a tendency of many phonological processes, but rather a defining property. If the phonological learner assumes its target process applies locally (and therefore will not learn processes that do not have this property) then we have an explanation for why the attested patterns are restricted in this way. The suggestion is that nonlocal processes are likewise learned by a learner that assumes the defining property of the subregular functions that describe them. More work is needed to identify these other functional classes, but the results presented here serve as a crucial first step in what we see as a promising line of research.

Additional work is needed as well to refine the local function learner introduced in this paper. First is a formal definition of the class of functions it can learn, from which should follow a proof of the algorithm's correctness. This proof is necessary to conclude that it learns *all* and *only* the phonological mappings with the necessary property of locality and will address concerns that the learner succeeds based on some other, coincidental property of the particular examples used to test it. Along with the proof is an identification of the algorithm's characteristic data sample. It was briefly discussed in Section 4.1 that a learner needs a certain amount of data in order to converge on the correct generalization. In fact, it is not the amount of data that matters as much as the content of that data. The characteristic sample that an algorithm needs to see in order to succeed includes all of the *crucial* data points, and this needs to be precisely defined to fully understand the conditions under which the learning algorithm succeeds.[3]

References

Albright, Adam, and Bruce Hayes. 2002. Modeling English past tense intuitions with minimal generalization. In *SIGPHON 6: Proceedings of the Sixth Meeting of the ACL Special Interest Group in Computational Phonology*.

Albright, Adam, and Bruce Hayes. 2003. Rules vs. analogy in English past tenses: A computational/experimental study. *Cognition* 90:119–161.

Alderete, John, Adrian Brasoveanua, Nazarre Merchant, Alan Prince, and Bruce Tesar. 2005. Contrast analysis aids in the learning of phonological underlying forms. In *Proceedings of the 24th West Coast Conference on Formal Linguistics*, 34–42. cascadilla.

Baayen, Rolf Harald, Richard Piepenbrock, and Leon Gulikers. 1996. *CELEX2*. Philadelphia: Linguistic Data Consortium.

Boersma, Paul. 1997. How we learn variation, optionality, and probability. In *Proceedings of the Institute of Phonetic Sciences 21*.

Boersma, Paul, and Bruce Hayes. 2001. Empirical tests of the gradual learning algorithm. *Linguistic Inquiry* 32:45–86.

[3]For progress on these additional results, see Chandlee and Jardine (2014) and Chandlee (2014).

Chandlee, Jane. 2014. Strictly Local Phonological Processes. Doctoral dissertation, University of Delaware.

Chandlee, Jane, Angeliki Athanasopoulou, and Jeffrey Heinz. 2012. Evidence for classifying metathesis patterns as subsequential. In *Proceedings of the 29th West Coast Conference on Formal Linguistics*, 303–309. Cascadilla.

Chandlee, Jane, and Jeffrey Heinz. 2012. Bounded copying is subsequential: Implications for metathesis and reduplication. In *Proceedings of SIGMORPHON 12*.

Chandlee, Jane, and Adam Jardine. 2014. Learning phonological mappings by learning strictly local functions. In *Proceedings of the 2013 Meeting on Phonology*, ed. John Kingston, Claire Moore-Cantwell, Joe Pater, and Robert Staubs.

Chomsky, Noam. 1956. Three models for the description of language. *IRE Transactions on Information Theory 113124* IT-2.

Gafos, Diamandis. 1996. The Articulatory Basis of Locality in Phonology. Doctoral dissertation, Johns Hopkins University.

Gainor, Brian, Regine Lai, and Jeffrey Heinz. 2012. Computational characterizations of vowel harmony patterns and pathologies. In *Proceedings of the 29th West Coast Conference on Formal Linguistics*, 63–71. Cascadilla.

García, Pedro, and Enrique Vidal. 1990. Inference of k-testable languages in the strict sense and application to syntactic pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence archive* 12:920–925.

Gildea, Daniel, and Daniel Jurafsky. 1996. Learning bias and phonological-rule induction. *Computational Linguistics* 22:497–530.

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

Hansson, Gunnar. 2001. Theoretical and Typological Issues in Consonant Harmony. Doctoral dissertation, University of California, Berkeley.

Hansson, Gunnar. 2007. On the evolution of consonant harmony: The case of secondary articulation agreement. *Phonology* 24:77–120.

Hayes, Bruce. 2004. Phonological acquisition in Optimality Theory: the early stages. In *Fixing Priorities: Constraints in Phonological Acquisition*, ed. Rene Kager, Joe Pater, and Wim Zonneveld. Cambridge University Press.

Heinz, Jeffrey. 2009. On the role of locality in learning stress patterns. *Phonology* 26:303–351.

Heinz, Jeffrey. 2010. Learning long-distance phonotactics. *Linguistic Inquiry* 41:623–661.

Heinz, Jeffrey, and William Idsardi. 2011. Sentence and word complexity. *Science* 333:295–297.

Heinz, Jeffrey, and Regine Lai. 2013. Vowel harmony and subsequentiality. In *Proceedings of the 13th Meeting on the Mathematics of Language (MoL 13)*, 52–63.

de la Higuera, Colin. 2010. *Grammatical Inference: Learning Automata and Grammars*. Cambridge: Cambridge University Press.

Johnson, C. Douglas. 1972. *Formal Aspects of Phonological Description*. The Hague: Mouton.

Johnson, Mark. 1984. A discovery procedure for certain phonological rules. In *Proceedings of the Tenth International Conference on Computational Linguistics*, 344–347. Stanford.

Kaplan, Ronald M., and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics* 20:371–387.

Kenstowicz, Michael. 1994. *Phonology in Generative Grammar*. Cambridge, MA: Blackwell.

Koskenniemi, Kimmo Matti. 1983. *Two-Level Morphology: A general computational model for word-form recognition and production*. University of Helsinki, Department of General Linguistics.

Luo, Huan. 2013. Long-distance consonant harmony and subsequentiality. Unpublished manuscript, University of Delaware.

Magri, Giorgio. 2010. Complexity of the acquisition of phonotactics in Optimality Theory. In *Proceedings of SIGMORPHON 2010*, 19–27.

Magri, Giorgio. 2012. An approximation approach to the problem of the acquisition of phonotactics in Optimality Theory. In *Proceedings of SIGMORPHON 2012*.

McNaughton, Robert, and Seymour A. Papert. 1971. *Counter-Free Automata*. Cambridge, MA: MIT Press.

Merchant, Nazarre, and Bruce Tesar. 2008. Learning underlying forms by searching restricted lexical subspaces. In *Proceedings of CLS 41*.

Mohri, Mehryar. 1997. Finite-state transducers in language and speech processing. *Computational Linguistics* 23:269–311.

Oncina, José, Pedro García, and Enrique Vidal. 1993. Learning subsequential transducers for pattern recognition interpretation tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15:448–457.

Pater, Joe. 2004. Exceptions and Optimality Theory: Typology and learnability. In *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*, 1777–1180. Barcelona.

Payne, Amanda. 2013. Dissimilation as a subsequential process. Unpublished manuscript, University of Delaware.

Prince, Alan, and Bruce Tesar. 2004. Fixing priorities: constraints in phonological acquisition. In *Fixing Priorities: Constraints in Phonological Acquisition*. Cambridge: Cambridge University Press.

Riggle, Jason. 2003. Nonlocal reduplication. In *Proceedings of the 34th annual meeting of the North Eastern Linguistic Society*.

Riggle, Jason. 2004. Generation, Recognition, and Learning in Finite State Optimality Theory. Doctoral dissertation, UCLA.

Riggle, Jason. 2006. Using entropy to learn OT grammars from surface forms alone. In *Proceedings of the 25th West Coast Conference on Formal Linguistics*. Cascadilla.

Rogers, James, and Geoffrey K. Pullum. 2011. Aural pattern recognition experiments and the subregular hierarchy. *Journal of Logic, Language and Information* 20:329–342.

Rose, Sharon, and Rachel Walker. 2004. A typology of consonant agreement as correspondence. *Language* 80:475–531.

Schieber, Stuart. 1985. Evidence against the context-freeness of natural language. *Linguistics and Philosophy* 8:333–343.

Tesar, Bruce. 1995. Computational Optimality Theory. Doctoral dissertation, University of Colorado at Boulder.

Tesar, Bruce. 1998. An iterative strategy for language learning. *Lingua* 104:131–145.

Tesar, Bruce, and Paul Smolensky. 1993. The learnability of Optimality Theory: An algorithm and some basic complexity results. Ms., University of Colorado, Boulder.

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

Department of Linguistics and Cognitive Science
University of Delaware
Newark, DE 19716
janemc@udel.edu
koirala@udel.edu