

Learning Phonological Grammars for Output-Driven Maps

Rutgers University has made this article freely available. Please share how this access benefits you.
Your story matters. <https://rucore.libraries.rutgers.edu/rutgers-lib/40802/story/>

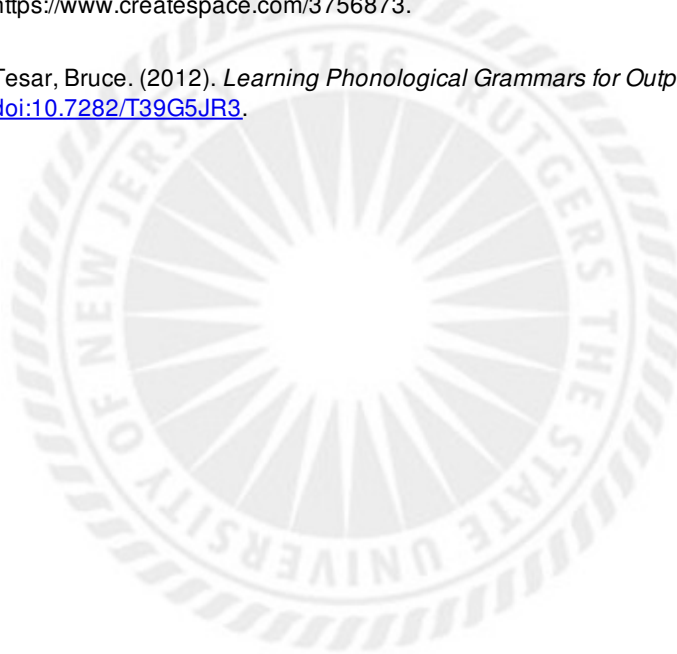
This work is an **ACCEPTED MANUSCRIPT (AM)**

This is the author's manuscript for a work that has been accepted for publication. Changes resulting from the publishing process, such as copyediting, final layout, and pagination, may not be reflected in this document. The publisher takes permanent responsibility for the work. Content and layout follow publisher's submission requirements.

Citation for this version and the definitive version are shown below.

Citation to Publisher Tesar, Bruce. (2012). *Learning Phonological Grammars for Output-Driven Maps*.
Version: <https://www.createpace.com/3756873>.

Citation to *this* Version: Tesar, Bruce. (2012). *Learning Phonological Grammars for Output-Driven Maps*. Retrieved from [doi:10.7282/T39G5JR3](https://doi.org/10.7282/T39G5JR3).



Terms of Use: Copyright for scholarly resources published in RUcore is retained by the copyright holder. By virtue of its appearance in this open access medium, you are free to use this resource, with proper attribution, in educational and other non-commercial settings. Other uses, such as reproduction or republication, may require the permission of the copyright holder.

Article begins on next page

Learning Phonological Grammars for Output-Driven Maps*

Bruce Tesar

Rutgers University, New Brunswick

1. Introduction

This paper proposes that a new characterization of phonological systems, output-driven maps, provides the kind of formal structure needed to overcome computational difficulties in the learning of constraint rankings and underlying forms.

1.1 Terminology

A **candidate** is an input, an output, and a correspondence relation between them. An **input** for a word is constructed from the underlying forms for the morphemes of the word. A candidate has a set of (zero or more) disparities. A **disparity** is a difference between the input and the output of a candidate, for example when corresponding segments differ in the value of a feature. The candidate shown in (1) has two disparities; the subscripts are IO correspondence indices. The corresponding segments with index 2 disagree in stress: the input segment is unstressed, while the output segment is stressed. The corresponding segments with index 4 disagree in length: the input segment is long, while the output segment is short.

(1) / p₁a₂k₃a:₄ / → [p₁á₂k₃a₄]

A **mapping** is a grammatical candidate (the actual assignment of an output to that input in the language). A **map** is a set of grammatical candidates, the well-formed structural descriptions of a language.

1.2 A System for Illustration

* I would like to acknowledge Crystal Akers for helpfully working through an example with an early version of the proposal. Valuable discussion and comments were provided by Alan Prince, and audiences at NELS39 and NECPhon2 (NorthEast Computational Phonology Meeting).

Each word consists of a root and a suffix (both monosyllabic). Each vowel has two features. The length feature has the values long (+) and short (-). The main stress feature has the values stressed (+) and unstressed (-). The constraints are as shown in (2).

- (2) The constraints (McCarthy and Prince 1993, McCarthy and Prince 1995, Prince 1990, Rosenthal 1994)

MAINLEFT	main stress on the initial syllable
MAINRIGHT	main stress on the final syllable
*V:	no long vowels
WSP	long vowels must be stressed (weight-to-stress principle)
FAITHSTRESS	IO correspondents have equal stress value
FAITHLENGTH	IO correspondents have equal length value

This system defines a typology of 24 languages. One of the languages is shown in (3); we will call this case Language A. It is generated by the constraint ranking given in (4). Briefly stated, Language A has lexical stress, with stress on the initial syllable by default, and long vowels shorten in unstressed position.

- (3) Language A

r1 = /pa/	r2 = /pa:/	r3 = /pá/	r4 = /pá:/	
páka	pá:ka	páka	pá:ka	s1 = /-ka/
páka	pá:ka	páka	pá:ka	s2 = /-ka:/
paká	paká	páka	pá:ka	s3 = /-ká/
paká:	paká:	páka	pá:ka	s4 = /-ká:/

- (4) WSP ≫ FAITHSTRESS ≫ MAINLEFT ≫ MAINRIGHT ≫ FAITHLENGTH ≫ *V:

2. Learning Phonologies

A phonological learner must simultaneously learn the ranking and lexicon (Hale and Reiss 1997, Tesar and Smolensky 1996). This poses a computational challenge, in part because of the explosive combinatorial growth in the number of possible combinations of constraint rankings and lexica. Exhaustively evaluating all possible lexicon-ranking combinations (Hale and Reiss 1997) is hopelessly intractable. Even modest assumptions lead to large numbers. In a system where all segments possess 10 binary features and all underlying forms have four segments, a morpheme has $2^{40} \approx 10^{12}$ possible underlying forms. For a lexicon of 1000 morphemes, that yields $2^{40 \times 1000} \approx 10^{12041}$ possible lexica alone (Tesar 2007).¹ The number of possible rankings grows factorially in the number of constraints, and the number of possible lexicon-ranking combinations is the product of the number of each.

¹ By comparison, the number of atoms in the universe is commonly estimated to be about 10^{80} .

Versions of this computational challenge appear in recent work on phonological learnability. Jarosz (2006), investigating an approach based on likelihood maximization, is able to bias the learner towards more restrictive grammars, but the algorithm separately evaluates each possible underlying form for each morpheme (as well as each possible constraint ranking). Apoussidou (2007), investigating an approach based on lexical constraints against possible underlying forms (Boersma 2001), is able to avoid exhaustive search of all possible rankings, but the algorithm separately evaluates each possible underlying form for each morpheme.

Merchant (Merchant 2008, Merchant and Tesar 2008) proposes evaluating **local lexica** for a small morpheme set, in an approach that determines underlying forms by setting one feature at a time (rather than treating underlying forms monolithically). A local lexicon is a possible assignment of feature values to **unset** underlying features, so the number of possible local lexica goes down as more features are set. This is better than exhaustive search of all possible underlying forms, but the number of local lexica is still exponential in the number of unset features.

Each of these recent lines of research has advanced the field in various ways, but computationally the techniques are still implausibly slow. Processing all underlying forms for even a modest number of morphemes gets expensive very quickly. The claim motivating the present work is that faster (and more cognitively plausible) learning will require additional posited structure in the space of possible grammars, structure that can be exploited by a learner to effectively search the space without exhaustively evaluating all (or even most) of the possibilities in the space. The concept of output-driven maps, described in the next section, is here proposed as that additional structure.

3. Output-Driven Maps

The concept of output-driven map (Tesar 2008) originates in an old issue in phonological theory, the extent to which phonological generalizations can be expressed in terms of restrictions on the output (Chomsky 1964, Kiparsky 1971, Kiparsky 1973, Kisseberth 1970, for an overview see McCarthy 2007). For over three decades, this issue has commonly been discussed in terms of the transparency/opacity of phonological processes (Kiparsky 1971, Kiparsky 1973). However, the notion of phonological process does not fit with some theoretical frameworks as well as it does with others. For example, in Optimality Theory (Prince and Smolensky 1993/2004) there is no natural theoretical primitive analogous to a phonological process.

The concept of output-driven map formally characterizes the intuitive notion of “determined by restrictions on the output” without any reference to phonological processes. It refers to the representational elements of structural descriptions (inputs, outputs, and the correspondences between them), and can apply equally to analyses expressed within SPE (Chomsky and Halle 1968), Optimality Theory, or any other framework formalizing phonology in terms of an input-output relation.

The definition of an output-driven map, given in (5), is stated in terms of candidates and a similarity relation on the candidates. Intuitively, it can be interpreted as saying that if an input A maps to X , and an input B is more similar to X than A is, then B must also map to X .

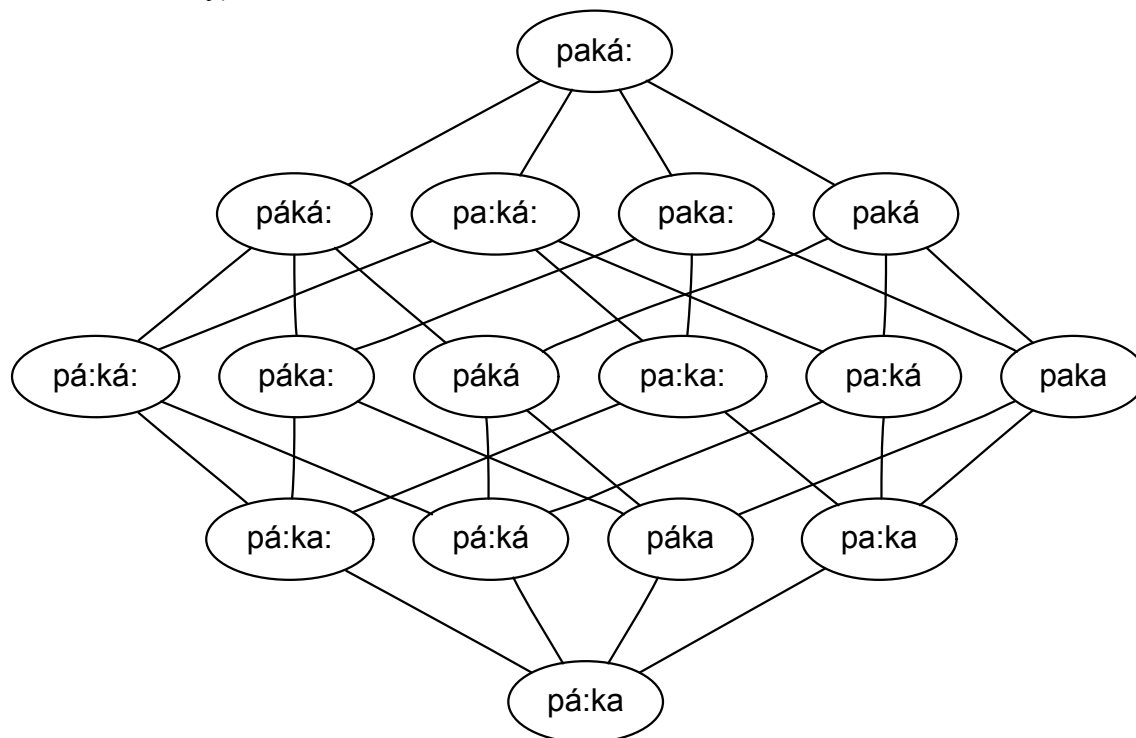
- (5) A map is output-driven if, for every grammatical candidate $A \rightarrow X$ of the map,
 if candidate $B \rightarrow X$ has greater similarity than $A \rightarrow X$,
 then $B \rightarrow X$ is also grammatical (it is part of the map).

The notion of similarity is here characterized in terms of disparities. For present purposes, two candidates can only be compared for relative similarity if they have the same output form (if two candidates have different output forms, neither can have greater similarity than the other). Candidate $A \rightarrow X$ has greater similarity than $B \rightarrow X$ if every disparity of $B \rightarrow X$ has an identical corresponding disparity in $A \rightarrow X$; $B \rightarrow X$ has greater similarity than $A \rightarrow X$ if it has a subset of the disparities of $A \rightarrow X$.

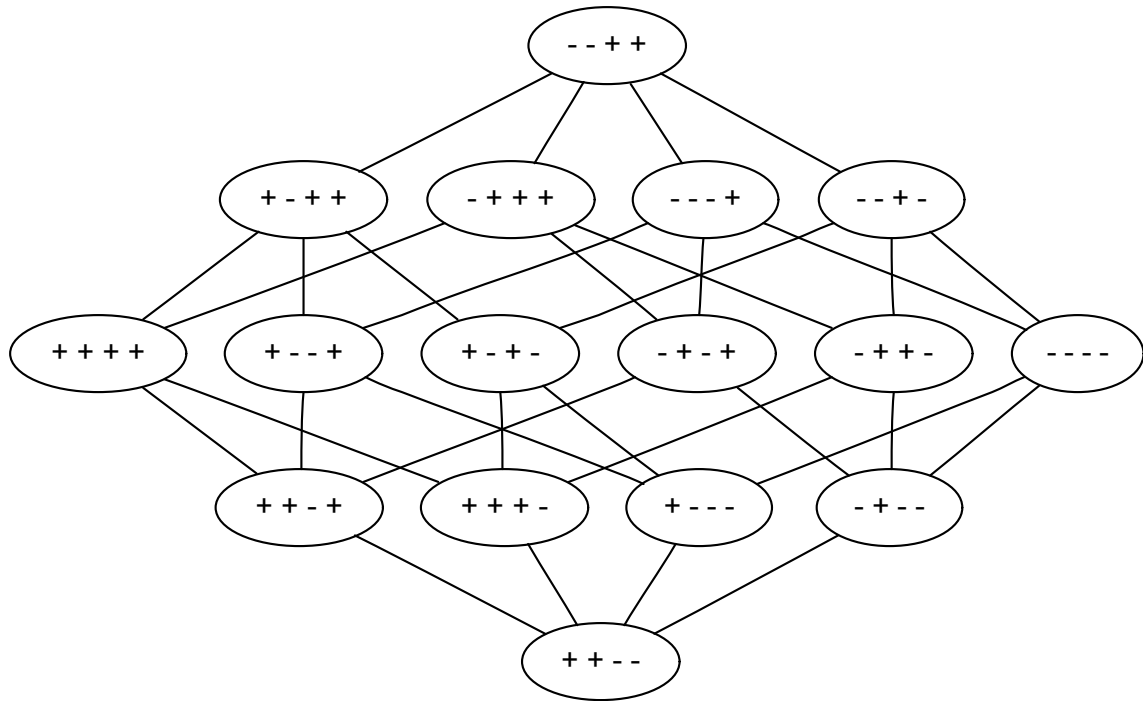
- (6) Candidate $B \rightarrow X$ has greater similarity than $A \rightarrow X$.
 $B \rightarrow X$ $\text{paká} \rightarrow \text{paká:}$ Disparities: length in seg. 4
 $A \rightarrow X$ $\text{páká} \rightarrow \text{paká:}$ Disparities: stress in seg. 2, length in seg. 4

The diagram in (7) shows the relative similarity relation, which is in fact a lattice, for the output paká: . Each node represents a candidate, with the text within each node

- (7) Relative similarity relation for output paká: : (higher in the graph means greater similarity)



(8) Relative similarity relation for output paká: (features version)



indicating the input for that candidate (all candidates have output paká:). Candidate $B \rightarrow X$ of (6) is the last (rightmost) candidate of the second row down, while $A \rightarrow X$ of (6) is the third candidate from the left in the third row down. If one candidate is above another, it means that the first (higher) candidate has greater similarity than the other.

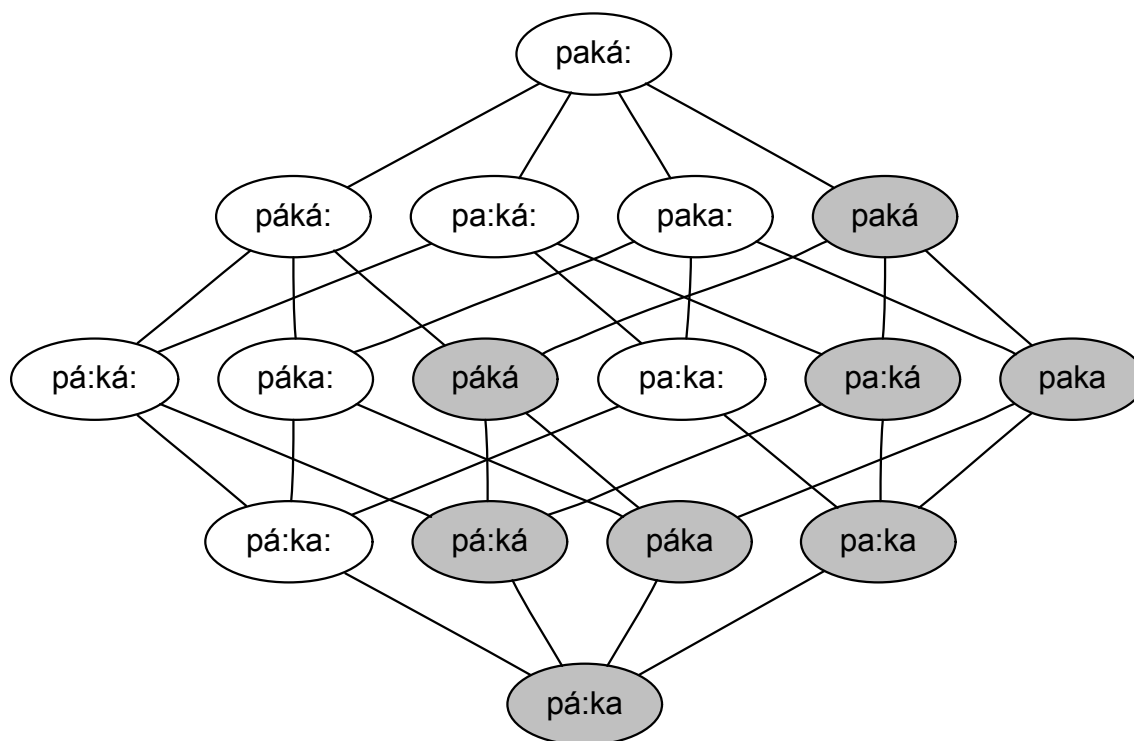
It can be easier to visually process the relative similarity order if feature matrices are used in place of textual representations of the words. Such a diagram is given in (8), where the four features are given in the order [root-stress root-length suffix-stress suffix-length].

The top node in the similarity relation represents the input with the greatest similarity to the output: the output itself. The top candidate has zero disparities. The candidates immediately below the top one each have a single disparity with the output (one such candidate for each feature). This continues down the order until the bottom is reached: the candidate in which the input differs on every feature from the output.

4. Exploiting ODM Structure in Learning

A learner can benefit from the knowledge that the language being learned is output-driven. Specifically, the learner can capitalize on the entailment relations between different candidates to arrive at conclusions about both underlying forms and the ranking without having to generate and evaluate all of the relevant possible underlying forms.

- (9) Setting s_4 to +long. This is the relative similarity lattice for the output of $r1s_4$, [paká:]. The shaded sublattice contains all candidates with s_4 underlyingly –long.



Output-driven maps are defined by an entailment relation: $A \rightarrow X$ entails $B \rightarrow X$. It is easier to see how this can be exploited in learning if one considers the logically equivalent contrapositive form: $\text{NOT}(B \rightarrow X)$ entails $\text{NOT}(A \rightarrow X)$. If a learner can determine that B is not the correct input for X , then the learner may automatically conclude that A cannot be the correct input for X either.

4.1 Learning Underlying Forms

The use of the output-driven map structure for the learning of underlying forms can be illustrated by considering the length feature of suffix s_4 in language A . The output of word $r1s_4$ is [paká:]. The learner can test the length feature of s_4 by constructing an input with just a single disparity relative to that output, a disparity in the length feature of the suffix. That input is /paká/. The learner then constructs the candidate /paká/→[paká:], with the output of $r1s_4$ and only a disparity in the suffix length. The full relative similarity lattice for [paká:] is given in (9). The shaded sublattice consists of all the candidates that have a disparity with the output for the suffix length feature: they all have s_4 underlyingly –long. The single disparity candidate thus has a subset of the disparities of all of the other candidates in the sublattice; /paká/→[paká:] has greater similarity than any other candidate for this output with a disparity in the length of the suffix.

The payoff comes if the learner is able to determine that the candidate /paká/→[paká:] cannot be optimal. Testing a candidate for consistency with what a

learner has learned thus far can be done with multirecursive constraint demotion (Tesar 1997, Tesar 2004). If the learner determines that /paká/→[paká:] cannot be optimal, then the learner may conclude, via the contrapositive form of the output-driven map property, that none of the candidates in the entire sublattice can be optimal. The only remaining viable candidates for the output have something in common: they all have s4 underlyingly +long. Thus, the learner can conclude that the underlying form for s4 is set to +long.

The benefit is one of computational efficiency. Even though half of the possible underlying forms have the suffix –long underlyingly, the learner does not need to evaluate all of them (as was previously proposed in Merchant 2008, Merchant and Tesar 2008), only the one at the top of the sublattice. The same benefit applies to every other unset feature of the word; only candidates with a single disparity need to be tested. This effectively converts exponential search into linear search: the number of possible underlying forms is exponential in the number of features, but the number of forms to actually be tested is linear in the number of features.

4.2 Simultaneous Consideration of Multiple Words

The strict conversion of exponential to linear is true without qualification when the learner is considering one word at a time, as discussed above. Things get a bit more complicated when multiple words are considered simultaneously, such as with the processing of a contrast pair (Tesar 2006a, Tesar 2006b). Specifically, the wrinkle occurs when more than one word in the set under consideration contains a particular morpheme, and a feature of that morpheme alternates within that set of words. For such a feature, there is no single underlying value that will match the surface everywhere; no matter what underlying value is chosen, it will create a disparity with at least one of the words of the set.

The solution to this is to test with respect to all values of the alternating feature, not just one. This is illustrated with the situation shown in (10), which features the contrast pair consisting of the words r1s1 and r1s3. The current lexicon is also shown, with four of the six underlying features for the relevant morphemes still unset (denoted with a question mark, ‘?’). Both words contain the root morpheme r1, and r1 alternates in the set: it surfaces as stressed in r1s1 and as unstressed in r1s3. Significantly, the stress feature of r1 is unset in the current lexicon (if an alternating feature is set, the learner needn’t consider it further, and there is no complication).

(10)	Contrast Pair:	r1s1 [páka]	r1s3 [paká]
	Current Lexicon:	r1 /?,-/	s1 /?,?/ s3 /?,-/

Testing the stress feature of r1 itself involves adopting the surface realization of each of the other three (non-alternating) features, and testing a pair of inputs for each value of r1: the pair of inputs {r1s1 / - - - - / r1s3 / - - + - /} for r1 with –stress, and {r1s1 / + - - - / r1s3 / + - + - /} for r1 with +stress. If both pairs of candidates prove to be consistent with current ranking information (as is the case in this example), then the

learner will not be able to set the stress feature of *s1* at this point, but can attempt to set one of the other features.

Testing the stress feature for *s3*, for example, is slightly more complicated. The learner already knows that the value matching *s3*'s surface realization in *r1s3*, +stress, will be consistent; it is not an alternating feature in this set. Setting the stress feature for *s3* here requires showing that the other value, –stress, is inconsistent. Because of the potential for interaction between the values of the stress features for *r1* and *s3*, the learner needs to test the –stress value for *s3* with **all** values for the stress feature of *r1*. This means testing two pairs of inputs: the inputs {*r1s1* / – – – – / *r1s3* / – – – – /} for *r1* with –stress, and {*r1s1* / + – – – / *r1s3* / + – – – /} for *r1* with +stress (note that both pairs have *s3* set to –stress in the second word). In this case, both pairs prove to be inconsistent with current ranking information; neither value of the stress feature for *r1* can bail out *s3* set to –stress. Thus, learner can set *s3* to +stress.

Computational complexity becomes an issue when there are multiple alternating features within a set of words being simultaneously considered. All values of each alternating feature must be considered independently, and there will be exponential growth in the number of combinations of values of such features. Fortunately, the potential for such growth is limited: it is only exponential growth in the number of **unset** features that **alternate within the set** of words under simultaneous consideration.

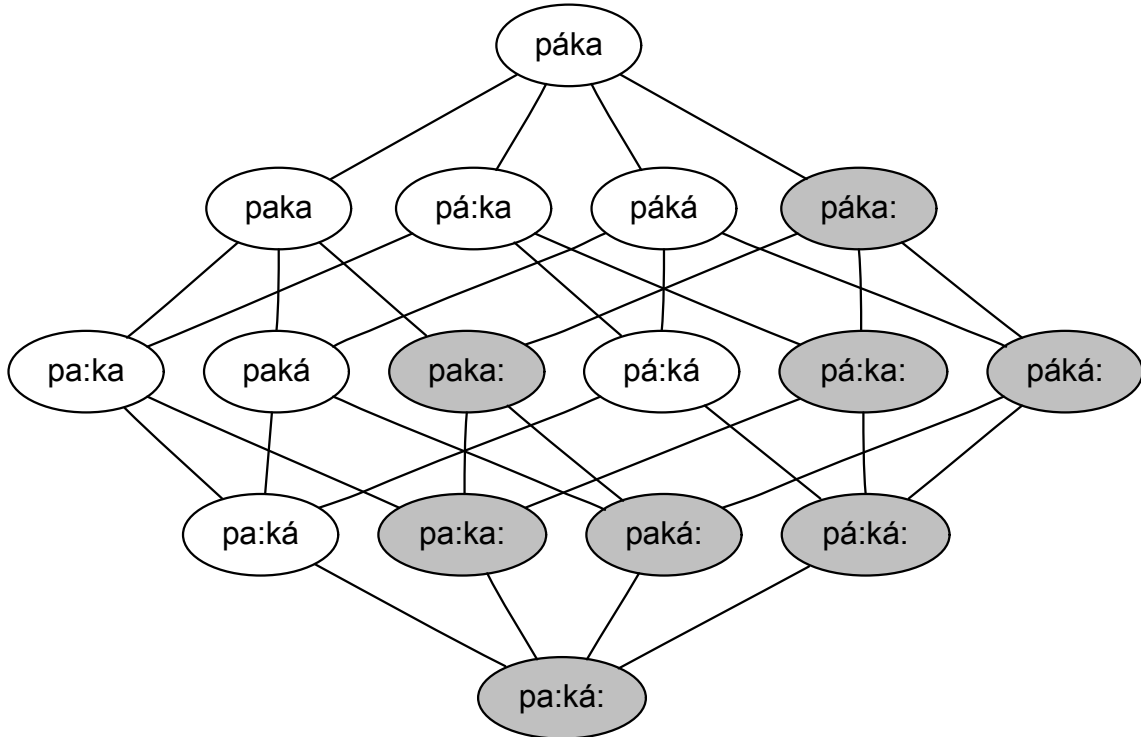
4.3 Learning the Ranking

Once a feature has been set for a given underlying form, the value is fixed for any word containing that morpheme. That fact, combined with output-driven structure, can be exploited to learn further, non-phonotactic ranking information. The key is to find a different word containing the same morpheme (in this case, *s4*), in which the set feature surfaces unfaithfully (Tesar 2006b). In Language A, such a word is *r3s4* [páka].

The lattice of candidates for the output of *r3s4* are shown in (11). Because *s4* has been set to be +long, none of the inputs that have *s4* as –long are still under consideration. The still-viable inputs, the ones with *s4* as +long, are shaded. Notice that these form a sublattice, with a top element. The form of the top element is predictable: it is the candidate in which the only disparities are those resulting from features that have been previously set; all features unset in the lexicon match the surface form of the word in the top element. In (11), the top element of the shaded sublattice, corresponding to the candidate /páka:/→[páka], has only one disparity, the one involving the length of *s4*.

Again, the learner can encapsulate the information from this entire subspace of possible inputs into a single form, the top of the sublattice. The top candidate in the sublattice has greater similarity than any other candidate in the sublattice. Because the sublattice contains all viable candidates, one of them must be grammatical. The learner doesn't know which one is the “true” underlying form for this word (*r3s4*), but the learner does know that the top candidate is a grammatical mapping, because if any viable candidate is grammatical, then the top one is. Thus, the learner can correctly deduce that

(11) The possible inputs for r3s4 (possible inputs are shaded).



the mapping $/páka:/ \rightarrow [páka]$ is part of the map. This mapping is non-phonotactic, because it is not fully faithful, and so it is an opportunity for the learner to obtain non-phonotactic ranking information.

The learner can obtain ranking information from this mapping by (again) using multirecursive constraint demotion. This is summarized in (12). The known non-faithful mapping, $/páka:/ \rightarrow [páka]$, is adopted as the winner. An appropriate loser is then selected via production-directed parsing on the input (Tesar 1998), in this case yielding the candidate with output identical to the input. Comparing the winner and loser yields the ERC listed in the fourth row of (12).

We can get a sharper sense of what new information has been obtained by combining this new ERC with a previously obtained ERC, one obtainable from purely phonotactic information. The relevant phonotactic ERC is shown in the fifth row. The phonotactic ERC simply expresses the observation that long vowels sometimes appear on the surface, and in this system that requires that faithfulness to length dominate the markedness constraint against long vowels. Taking the fusion of the two ERCs (Prince 2002), shown in the last row, reveals what the learner has obtained: WSP must dominate both $*V:$ and FAITHLENGTH. The conclusion that the faithfulness constraint must be dominated relies on the non-faithful element of the winner, the failure to faithfully preserve the underlying length on the suffix.

(12) Additional ranking information

/ páka:/	WSP	MAINL	MAINR	*V:	FAITHSTRESS	FAITHLENGTH
páka (winner)						*
páka: (loser)	*			*		
ERC	W			W		L
Phonotactic ERC				L		W
Fusion	W			L		L

The new ERC, along with the prior phonotactic ERC, combine to provide a partial picture of the desired ranking: WSP \gg FAITHLENGTH \gg *V:. The learner was able to obtain this information from the word r3s4 despite not knowing the complete input for the word (a consequence of not knowing the complete underlying forms for r3 and s4). Whenever a feature has been newly set in the lexicon, the learner can identify possible sources of additional ranking information by finding those words that do not faithfully preserve the feature. Each such word can be tested by constructing an input in which all features not set in the lexicon are taken to have the same value as their output counterparts. Output-driven map structure ensures that these mappings are valid in the target language, and any additional ranking information required to ensure that these mappings are optimal may be adopted by the learner.

5. Discussion

5.1 Initial Lexicon Construction

Initial lexicon construction (Tesar et al. 2003) is a procedure for early setting of some features in the lexicon. It examines the surface realizations of each morpheme, and sets each non-alternating feature to match its (single) surface realization. The original virtue of this was computational: a reduction in the number of unset features when the ‘complex’ learning began. Since in general the space of lexical hypotheses being considered is exponential in the number of unset features, an easy way to set even some features in advance can significantly reduce the space of lexical hypotheses being considered (‘exponential shrinkage’), benefitting learning procedures with computational requirements that are proportional to the number of lexical hypotheses.

Initial lexicon construction has always had a concern attached to it, however: to be fully correct, the learner has to have a representative sample of the possible surface realizations of each morpheme, so that it can accurately determine which features alternate. This would seem to require the learner to hold off on doing much learning until they had seen a healthy collection of forms with at least some morphemes. Further, it makes the learner potentially vulnerable if key surface realizations of some morphemes occur only infrequently.

The adoption of output-driven map structure in learning changes things. In the approach described in this paper, the computational demands of learning are linear, not exponential, in the number of unset features. Therefore, initial lexicon construction does

add much benefit for such a learner. Output-driven map structure has vitiated the virtue, and the learner is likely better off without it (and its attendant concerns).

5.2 The Final Lexicon

Abandoning initial lexicon construction means that features are only set when they are necessary, that is, when setting the feature to some other value would result in an incorrect output for some input containing the relevant morpheme. If the value of a particular feature for a particular morpheme never matters, then that feature will never be set underlyingly. In a sense, one could label a feature that isn't set as non-contrastive: it doesn't play a role in distinguishing the phonological identity of the morpheme from other possible morphemes.

There is a potential confusion to be avoided here. In the sense just described, contrastiveness is a property of feature tokens, not feature types. In general, some occurrences (in underlying forms) of a feature like length can be contrastive and others not, within the same language. Recall Language A, as listed in (3). Observe that suffixes *s1* and *s2* are phonologically indistinguishable on the surface: they surface identically in every morphological context (as do the entire words in which they appear). They differ underlyingly only in the value of the length feature. The length feature is not contrastive for suffixes *s1* and *s2*. Length **is** contrastive for *s3* and *s4*, as evidenced by the context of root *r1*: *r1s3* surfaces as *paká* (with a short final vowel), while *r1s4* surfaces as *paká:* (with a long final vowel). The difference between the sets of suffixes lies in the underlying value of the stress feature. Because default stress position is initial, suffixes can only be stressed on the surface if they are +stress underlyingly. Suffixes *s1* and *s2* are –stress underlyingly, so they will never surface as stressed. Because long vowels are shortened in unstressed position, the length of underlyingly –stress suffixes is predictably short: if underlyingly long, it will be shortened, and if underlyingly short, it will remain short. In Language A, there are only three phonologically distinguishable suffixes: –stress, +stress with –long, and +stress with +long.

The tokenness of feature contrast extends beyond relations with the underlying values of other features within the segment. Consider a language in which stress is predictably word-initial, vowels are shortened in unstressed position, and roots can be multisyllabic, so that a root could contain multiple vowels always surfacing in distinct syllables. In a two-syllable root, the first vowel would be contrastive for length, but the second vowel wouldn't be, because it would never be word-initial, and thus never surface as stressed.²

The concept here of permanently unset features should also not be confused with underspecification, in particular with temporary (input-only) underspecification (Archangeli 1984, Archangeli 1988, Kiparsky 1982, Steriade 1987, as cited in Steriade 1995). An unset feature is a construct of the learner, not the linguistic theory itself. A

² A similar case involves languages with syllable-final / word-final obstruent devoicing, and contrastive obstruent voicing in other obstruent-friendly environments.

feature left unset by the learner described here could be set to any value without effect, whereas an underspecified feature is such that assigning a value to the feature might well have phonological effects.

5.3 Expanding the Scope

Output-drivenness appears to hold for much of basic phonology. But there are a number of phenomena for which the standard analyses result in non-output-driven maps, such as synchronic chain shifts (Tesar 2008). Short of reanalyzing all such cases in purely output-driven terms, how such phenomena could be accommodated in learning will be highly dependent on how one chooses to account for them within core linguistic theory.

One possibility would be to develop a theory of phonological maps that is less restrictive than purely output-driven maps, but retains restrictive properties that could be exploited for learning along the same lines as is done with output-driven maps in this paper. This would be a plausible approach if non-output-driven maps are dealt with by using constraints that can introduce non-output-driven effects into the maps defined by Optimality Theoretic systems.

Another possibility would account for non-output-driven phenomena with Stratal Optimality Theory (Kiparsky 2003), in which the output is derived from the input by a series of OT grammars, each with distinct rankings. Adapting a line of thinking suggested by Bermudez-Otero (2003), one could propose that each of the individual OT grammars be output-driven, with non-output-driven effects resulting solely from interaction between the different grammars. Learning in such a theory could depend to a great extent on the proposals made here for learning each of the individual stratal grammars, and would require further principles for working out the relations between the different grammars as part of the process of learning them.

6. Conclusion

Output-driven maps are a good approximation to the structure of basic phonology. It provides structure in the space of possible grammars that goes beyond the structure provided by Optimality Theory, structure that can be exploited to great effect in learning, in particular in the mutual learning of underlying forms and constraint rankings. The lattice structure imposed on the space of possible inputs allows single input forms to stand in for entire sublattices, reducing an exponential exhaustive search to a linear search of just the key forms. This improvement from exponential to linear computation applies both to the learning of underlying forms and the learning of ranking information based on incomplete input forms. The theory of output-driven maps, in addition to providing insight into the nature of phonology, provides the kind of structure necessary to account for the efficiency of child language learning.

References

- Apoussidou, Diana. 2007. The Learnability of Metrical Phonology, Linguistics, University of Amsterdam: PhD.
- Archangeli, Diana. 1984. Underspecification in Yawelmani phonology and morphology, Linguistics, MIT: PhD.
- Archangeli, Diana. 1988. Aspects of underspecification theory. *Phonology* 5:183-207.
- Bermudez-Otero, Ricardo. 2003. The acquisition of phonological opacity. In *Variation within Optimality Theory: Proceedings of the Stockholm workshop on 'Variation within Optimality Theory'*, eds. Jennifer Spenader, Anders Eriksson and Osten Dahl, 25-36. Stockholm: Dept. of Linguistics, Stockholm University. ROA-593 (expanded version).
- Boersma, Paul. 2001. Phonology-semantics interaction in OT, and its acquisition. In *Papers in Experimental and Theoretical Linguistics*, eds. Robert Kirchner, Wolf Wikeley and Joe Pater, 24-35. Edmonton: University of Alberta. ROA-369.
- Chomsky, Noam. 1964. *Current Issues in Linguistic Theory*. The Hague: Mouton.
- Chomsky, Noam, and Halle, Morris. 1968. *The Sound Pattern of English*. New York City: Harper & Row.
- Hale, Mark, and Reiss, Charles. 1997. Grammar Optimization: The simultaneous acquisition of constraint ranking and a lexicon. Montreal: Concordia University. ROA-231.
- Jarosz, Gaja. 2006. Rich Lexicons and Restrictive Grammars - Maximum Likelihood Learning in Optimality Theory, The Johns Hopkins University: PhD. ROA-884.
- Kiparsky, Paul. 1971. Historical linguistics. In *A survey of linguistic science*, ed. W. O. Dingwall, 576-649. College Park: University of Maryland Linguistics Program.
- Kiparsky, Paul. 1973. Abstractness, opacity and global rules (Part 2 of "Phonological representations"). In *Three Dimensions of Linguistic Theory*, ed. O. Fujimura, 57-86. Tokyo: TEC.
- Kiparsky, Paul. 1982. Lexical phonology and morphology. In *Linguistics in the Morning Calm*, ed. I. S. Yang, 3-91. Seoul: Hanshin.
- Kiparsky, Paul. 2003. Syllables and Moras in Arabic. In *The Syllable in Optimality Theory*, eds. Caroline Féry and Ruben van de Vijver, 147-182: Cambridge University Press.
- Kisseberth, Charles. 1970. On the functional unity of phonological rules. *Linguistic Inquiry* 1:291-306.
- McCarthy, John J. 2007. Derivations and levels of representation. In *The Cambridge Handbook of Phonology*, ed. Paul de Lacy, 99-117. Cambridge: Cambridge University Press.
- McCarthy, John J., and Prince, Alan. 1993. Generalized alignment. In *Yearbook of Morphology*, eds. Geert Booij and Jaap Van Marle, 79-154. Dordrecht: Kluwer.
- McCarthy, John J., and Prince, Alan. 1995. Faithfulness and Reduplicative Identity. In *University of Massachusetts Occasional Papers 18: Papers in Optimality Theory*, eds. Jill Beckman, Laura Walsh Dickey and Suzanne Urbanczyk, 249-384. Amherst, MA: GLSA, University of Massachusetts.
- Merchant, Nazarré. 2008. Discovering underlying forms: Contrast pairs and ranking, Linguistics Department, Rutgers University: PhD. ROA-964.

- Merchant, Nazarré, and Tesar, Bruce. 2008. Learning underlying forms by searching restricted lexical subspaces. In *Proceedings of the Forty-First Conference of the Chicago Linguistics Society (2005)*, vol. II: *The Panels*, 33-47. ROA-811.
- Prince, Alan. 1990. Quantitative consequences of rhythmic organization. In *CLS26-II: Papers from the Parasession on the Syllable in Phonetics and Phonology*, eds. Karen Deaton, Manuela Noske and Michael Ziolkowski, 355-398. Chicago, IL: Chicago Linguistics Society.
- Prince, Alan. 2002. Entailed Ranking Arguments. Ms. Rutgers University, New Brunswick. ROA-500.
- Prince, Alan, and Smolensky, Paul. 1993/2004. *Optimality Theory: Constraint interaction in generative grammar*. Malden, MA: Blackwell.
- Rosenthal, Sam. 1994. Vowel/glide alternation in a theory of constraint interaction, Linguistics, University of Massachusetts: PhD.
- Steriade, Donca. 1987. On class nodes. Ms. MIT.
- Steriade, Donca. 1995. Underspecification and markedness. In *The Handbook of Phonological Theory*, ed. John Goldsmith, 114-174. Cambridge: Blackwell.
- Tesar, Bruce. 1997. Multi-Recursive Constraint Demotion. Ms. Rutgers University, New Brunswick. ROA-197.
- Tesar, Bruce. 1998. Error-driven learning in Optimality Theory via the efficient computation of optimal forms. In *Is the Best Good Enough? Optimality and Competition in Syntax*, eds. Pilar Barbosa, Danny Fox, Paul Hagstrom, Martha Jo McGinnis and David Pesetsky, 421-435. Cambridge, MA: MIT Press.
- Tesar, Bruce. 2004. Using inconsistency detection to overcome structural ambiguity [Spring]. *Linguistic Inquiry* 35:219-253.
- Tesar, Bruce. 2006a. Faithful contrastive features in learning. *Cognitive Science* 30:863-903.
- Tesar, Bruce. 2006b. Learning from paradigmatic information. In *Proceedings of the 36th Meeting of the North East Linguistics Society*, eds. Christopher Davis, Amy Rose Deal and Youri Zabbal, 619-638: GLSA. ROA-795.
- Tesar, Bruce. 2007. Learnability. In *The Cambridge Handbook of Phonology*, ed. Paul de Lacy, 555-574. New York City: Cambridge University Press.
- Tesar, Bruce. 2008. Output-driven maps. Ms. Linguistics Dept., Rutgers University. ROA-956.
- Tesar, Bruce, and Smolensky, Paul. 1996. Learnability in Optimality Theory (long version). Baltimore, MD: The Johns Hopkins University. ROA-156.
- Tesar, Bruce, Alderete, John, Horwood, Graham, Merchant, Nazarré, Nishitani, Koichi, and Prince, Alan. 2003. Surgery in language learning. In *Proceedings of the Twenty-Second West Coast Conference on Formal Linguistics*, eds. G. Garding and M. Tsujimura, 477-490. Somerville, MA: Cascadilla Press. ROA-619.

Department of Linguistics
18 Seminary Place
Rutgers University
New Brunswick, NJ 08901

tesar@rutgers.edu