Phonological Learning with Output-Driven Maps

Phonological Learning with Output-Driven Maps

Bruce Tesar

Rutgers University, New Brunswick

Department of Linguistics

18 Seminary Place

New Brunswick, NJ 08901-1184 USA

Abstract

The concept of an output-driven map formally characterizes an intuitive notion about phonology: that disparities between the input and the output are introduced only to the extent necessary to satisfy restrictions on outputs. When all of the grammars definable in a phonological system are output-driven, the implied structure provides significant computational benefits to language learners. An output-driven map imposes significant structure on the space of possible inputs for words, which can allow a learner to efficiently learn a lexicon of phonological underlying forms despite the vast number of possible lexica, as well as contend with the challenges of map / lexicon interactions inherent in phonological learning. This paper presents a learning algorithm that exploits the structure of output-driven maps, illustrated with a system of grammars based in Optimality Theory. The algorithm highlights the roles played by contrast and paradigmatic information in phonological learning.

1. Learning Phonologies

A phonological learner must simultaneously learn the phonological map (from linguistic inputs to surface forms) and the lexicon for their language (Hale & Reiss 1997; Tesar & Smolensky 1996). This poses a computational challenge, in part because of the explosive combinatorial growth in the number of possible combinations of maps and lexica. Exhaustively evaluating all possible map/lexicon combinations (Hale & Reiss 1997) is hopelessly intractable. Consider an artificially small (by design) system in which there are 50 morphemes. Each morpheme consists of 3 segments, and each segment possesses 5 binary features. These conditions yield approximately 10^{200} possible lexica alone. By comparison, the observable universe contains approximately 10^{80} atoms. A "universal" computer, in which each atom in the universe evaluated a trillion lexica per second, would still take 10^{100} years to evaluate them all.

The computational literature on simultaneously learning a lexicon and a phonological map is relatively modest; early efforts include (Johnson 1984; Tesar et al. 2003; Tesar & Prince 2007; Tesar & Smolensky 2000: 77-83). The computational challenge described above stalks even more recent work. Some understanding has been gained on the challenges of map/lexicon interaction with work utilizing likelihood maximization (Jarosz 2006) and lexical constraints (Apoussidou 2007), but both approaches require evaluation of very large numbers of lexical hypotheses of one form or another. Work utilizing local lexica over small morpheme sets (Merchant 2008; Merchant & Tesar 2008) significantly reduces the number lexical combinations that need to be considered, but the number of local lexica is still exponential in the number of features.

Processing all underlying forms for even a modest number of morphemes gets computationally 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 section 2, is here proposed as that additional structure.

Any serious account of language acquisition must make theoretical linguistic commitments: an explanation of how knowledge of language is acquired requires some substantive view of what knowledge of language is. The present work commits to the view that knowledge of language includes a phonological grammar, and further that a phonological grammar consists of a lexicon of phonological underlying forms for morphemes, and a mechanism for assigning surface forms to phonological inputs, where phonological inputs are constructed from the underlying forms of the morphemes (Chomsky & Halle 1968).

More specifically, the present work commits to the view that the map mechanism consists of an Optimality Theoretic system (Prince & Smolensky 2004). The linguistic advantages and limitations of Optimality Theory are far too extensive to be rehearsed here, having been the subject of numerous entire volumes; see for example (de Lacy 2007; Kager 1999; McCarthy 2004; Prince & Smolensky 2004). Similarly, the role of Optimality Theory in accounts of phonological learning has a substantial literature with multiple entire volumes; see for example (Boersma 1998; Kager et al. 2004; Tesar & Smolensky 2000). A brief review of the key properties of Optimality Theory is given in section 3. More substantial discussion of recent work

in Optimality Theory and phonological learning that is relevant to the learning of phonological underlying forms and phonological maps can be found in (Tesar 2014).

The present work is part of a research program into the relations between the formal structure of linguistic theory and learnability. As such, it is not attempting to fully model any specific chronological stage of development (but see section 5.2). Instead, this work focuses on a few issues while abstracting away from a number of others. The issues of focus, concerning the simultaneous learning of an interacting map and lexicon from paradigmatically related words, are relevant to multiple chronological stages of development, without being the whole story for any one stage. Further, the intent is to identify formal structures which benefit both linguistic theory (supported by its relevant empirical phenomena) and language learnability (supported by its relevant empirical phenomena). Accounts based on structures which receive support from both linguistic theory and language learnability are more likely to be on the right track.

2. Output-Driven Maps

The concept of output-driven map (Tesar 2014) 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; Optimality Theory, for instance has no theoretical primitive that corresponds to the traditional notion of 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 instead is defined solely at the level of the map: the structured relationships between phonological inputs and their grammatically assigned outputs.

Figure (1) here.

The definition of an output-driven map, given in (1), is stated in terms of candidates and a similarity relation on the candidates. A *candidate* is a phonological representation, consisting of an input, an output, and a correspondence between input and output. The term 'candidate' here alludes to the fact that although there are many possible phonological representations containing a given input (with a variety of different outputs for that input), typically only one of those candidates will be actually included in the map (and thus determine the output for that input). Intuitively, the definition of output-driven map 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. The intuition can be summarized in slightly simpler terms: for every mapping A→X, if B is more similar to X than A is, then B also maps to X.

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→X has greater similarity than B→X if every disparity of B→X has an *identical corresponding disparity* in A→X; B→X has greater similarity than A→X if it has a subset of the disparities of A→X. In the example in (2), candidate A→X has two disparities, while candidate B→X has only one of the disparities of A→X (length in vowel 2) and no others. Therefore, candidate B→X has greater similarity than candidate A→X. Simply having fewer

disparities is inadequate; to be of greater similarity, candidate B→X must only contain disparities with identical corresponding disparities in A→X.

Figure (2) here.

Output-drivenness imposes entailment relations between the presence of candidates in a map. If one candidate is contained in an output-driven map, then any other candidates that have greater similarity must also be contained in the map. Expanding on the example in (2), four candidates are shown in (3). The second through fourth candidates each have greater similarity than the first; in an output-driven map, the presence of the first candidate entails the presence of the other three in the map. Put another way: if the map is willing to overcome certain 'obstacles' to reach an output (tolerate certain disparities between the input to the output), then simply removing some of the obstacles (by changing the input), without adding any new ones, ensures reaching that same output.

Figure (3) here.

3. Output-Drivenness and Optimality Theory

3.1 A System for Illustration: The Stress/Length System

An Optimality Theoretic system consists of a specification of possible representations, labeled GEN, and a set of constraints that evaluate representations, labeled CON. Each phonological representation, commonly referred to as a candidate, consists of an input, an output, and a correspondence relation between the two. Each candidate is evaluated by the constraints: one candidate fares worse than another with respect to a constraint if it incurs more violations of the constraint. However, the constraints can conflict with each other relative to candidates; a candidate which does better than a competitor on one constraint might fare worse that the same

competitor on another constraint. In Optimality Theory, such conflicts are resolved via lexicographic optimization (the same basic concept used when alphabetically ordering a set of words). A grammar imposes an order, commonly referred to as a ranking, on the constraints in CON, so that the constraint at the top of the ranking has priority over the others, the next constraint down has priority over the constraints below it, and so forth. A more illustrated (if somewhat dated) overview of Optimality Theory with an eye toward learnability can be found in (Tesar & Smolensky 2000).

The principle of Richness of the Base (Prince & Smolensky 2004) asserts that the space of possible inputs (the domain of GEN) is universal. The map for a language is the function mapping each input to a grammatical candidate. The lexical entries learned by a learner are for morphemes appearing in words observed by the learner; the ranking learned by a learner must enforce grammatical restrictions on the outputs for all inputs, whether realized in the lexicon or not.

The rest of this paper will use the following linguistic system, labeled the stress/length system, for purposes of illustration (Tesar 2006a). 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 (–). Each candidate has an input, such as /páká:/, and an output, such as [páka]. The example system abstracts away from the possibilities of insertion, deletion, and coalescence, so the input-output correspondence is always an order-preserving bijection: the first segment of the input corresponds to the first segment of the output, and so forth, The constraints are as shown in (4).

Figure (4) here.

GEN and CON are universal: all learners have the same set of possible phonological representations and the same set of constraints. However, the ranking of those constraints can vary cross-linguistically. Crucially, this determines the space of possible map descriptions that a learner must work with. Learning the correct map for their language means learning a correct ranking of the constraints. The space of possible grammars is then the space of possible permutations (rankings) of the set of constraints. The space of rankings has a combinatorial structure, as the number of possible permutations is the factorial of the number of constraints: a set of 6 constraints yields a set of 6! = 720 distinct rankings. The actual typology of distinct languages predicted is typically much less than the number of distinct rankings due to redundancy; various constraints will not interact in certain circumstances (e.g., when both are dominated by some other constraints), so it is common for multiple rankings to generate the very same language. However, the learner is not assumed to have any a priori knowledge of the extensional consequences of interaction between particular constraints. As far as the learner is concerned, their space of possible maps is the space of all possible rankings of the constraints.

The stress/length system defines a typology of 24 languages; for a full presentation of the typology see (Tesar 2014: 237-45). One of the languages is shown in (5); we will call this language L20. It is presented as a paradigmatic table: the word r2s3 (root r2 combined with suffix s3) has input /pa:ká/ (formed by concatenating the underlying forms of the morphemes) and output paká (indicated in the cell in column r2 and row s3). L20 is generated by the constraint ranking given in (6). Briefly stated, L20 has lexical stress, with stress on the initial syllable by default, and long vowels shorten in unstressed position. Note that suffixes s1 and s2 neutralize in all environments, in fact they both surface with a short, unstressed vowel in

combination with all four roots, despite the fact that s1 is short underlyingly while s2 is long underlyingly.

Figure (5) here.

Figure (6) here.

There are a total of 8 morphemes in each language of this system: 4 roots and 4 morphemes. Each morpheme has two features: stress and length. Thus each morpheme has 4 possible underlying forms. The space of possible lexica that the learner can choose from has size 4^8 = 65,536. As with the rankings, typologically there is often a significant amount of redundancy among lexica, but what underlying forms are equivalent is highly dependent on the ranking.

The space of possible grammars for the stress/length system consists of all possible combinations of rankings and lexica: 720 rankings * 65,536 lexica = 47,185,920 grammars. Any learning algorithm is going to have to escape this rapid combinatorial growth if it is to have any hope of scaling up to human grammars. Fortunately, the efficiency with which a space can be searched is dependent on the structure of the space, not its size. The structure of output-driven maps will be shown below to provide great power in efficiently learning in the stress/length system, and by extension, other linguistic systems with similar structure.

3.2 Relation to Other Work on Stress Learnability

The stress/length system has several methodological virtues for present purposes. It is compact, with only two types of features, making it easier to exhaustively study, while still complex enough to contain the key properties of interest. The two features can both be contrastive or neutralized in different languages of the typology, they can be conditionally neutralized in some

languages, and they interact with each other with sufficient freedom that in some languages length phenomena are driven by stress, while in other languages stress phenomena are driven by length. Furthermore, stress is culminative in this system: every word must have exactly one main stress. The mere presence of main stress does not determine the fate of underlying stress features; for the interesting cases, paradigmatic relations between words must be made use of.

Metrical stress has been the empirical focus of substantial prior work on language learnability. Some work has been done using Optimality Theory (Apoussidou & Boersma 2003; Boersma 2003; Jarosz 2013; Tesar 2004; Tesar & Smolensky 2000), some using the Principles and Parameters (Chomsky 1981) framework (Dresher 1999; Dresher & Kaye 1990; Pearl 2011), and some using still other theoretical commitments (Goldsmith 1994; Gupta & Touretzky 1994; Heinz 2009). What all of the work just mentioned have in common is a focus on predictable stress patterns. What they all lack is an investigation of lexically specified stress systems with underlying stress specifications. Much of that work is concerned with structural ambiguity, in particular the ambiguity of foot structure with respect to overtly observable stress patterns. By contrast, the present work abstracts away from structural ambiguity, and focuses on learning with a system in which both predictable stress systems and lexical stress systems (and in-between systems) are possible, along with the possibility of lexical weight specification in the form of vowel length. Akers (2012), in work that also capitalizes on the structure of output-driven maps, has developed a computational model that addresses both kinds of hidden structure, simultaneously learning the map and the lexicon in the presence of ambiguous foot structure; her work does not involve lexical specification of vowel length, however.

3.3 Relative Similarity in the Stress/Length System

Input forms and output forms both arise from assignments of values to the vowel features. Each word has two vowels, each with two features, so there are a total of 16 possible input forms, and 16 possible output forms. For any given output, there are 16 candidates, one for each pairing of an input with that output. Because two candidates can only have a similarity relation between them if they have identical outputs, we can justifiably think of "the" relative similarity relation for a given output, built on all (and only) the candidates containing that output.

The diagram in (7) shows the relative similarity relation for the output paká: (this subrelation is a lattice). Each oval node represents a candidate; the text within the node indicates the input for that candidate (all candidates have output paká:). Candidate B→X of (2) is the rightmost candidate of the second row down, while A→X of (2) is the third candidate from the left in the third row down. Two candidates are related by similarity if they are directly connected by a line, or if there is a strictly downward-heading path along the lines from one candidate to the other. If two candidates are related, it means that the higher candidate in the lattice has greater similarity than the other. Candidate B→X, paká → paká:, has a downward line directly to candidate A→X, páká → paká:, representing the fact that B→X has greater similarity than A→X.

Figure (7) here.

The top candidate has an input identical the the output. It has zero disparities; candidates of this sort can be referred to as identity candidates. 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

If all available grammars define output-driven maps, the learner can capitalize on the entailment relations between different candidates to learn both underlying forms and the ranking without having to generate and evaluate all of the relevant possible underlying forms. The primary benefit here is computational: a vast reduction in the amount of computational effort needed to jointly learn the ranking and the lexicon.

This section traces through the learning of L20, as executed by a learning algorithm known as the Output-Driven Learner, or ODL (Tesar 2014). The ODL starts with phonotactic learning, which learns certain aspects of the ranking. That is followed by a round of single word learning, where aspects of underlying forms and further ranking information are jointly learned by processing a single word at a time. After the first round of single word learning, contrast pair learning is performed, in which a selected pair of words is processed together to obtain certain crucial information about underlying forms. Contrast pair learning is followed by a second (and final) round of single word learning. Output-drivenness makes a contribution to learning in each major component of learning in this model.

A brief introduction to a number of the building block components of learning used here, including inconsistency detection with Recursive Constraint Demotion (RCD), error-driven learning with Multi-Recursive Constraint Demotion (MRCD), and restrictiveness enforcement with Biased Constraint Demotion (BCD) can be found in (Tesar 2007).

4.1 Phonotactic Learning

Phonotactic learning is the early stage of learning in which the learner has access to the surface forms of words, but has not yet determined any morphological relations between words. A long-

standing view is that a learner can make some progress early on by assuming a phonological input that is featurally identical to the output, and determining what must be true of the constraint ranking in order to map the input to the output (Hayes 2004; Prince & Tesar 2004). This approach is most successful when the maps determined by the possible grammars are idempotent: each generated output, when adopted as an input, in fact maps to "itself", in the sense of featural equivalence of the segments.

Idempotence follows as a consequence of output-drivenness. The candidate in which a form maps to itself is an identity candidate, distinguished by having zero disparities (all output segments have the same feature values as their input correspondents). The identity candidate has greater similarity than any other candidate for the same output. If an output is in the language, then it is generated from some input, and must also be generated by the input of the identity candidate.

If paká: is an output observed by the learner, as is the case for L20, then some input must map to it. The learner knows that if any input maps to paká:, then /paká:/ maps to paká:, whether there is lexical cause to construct the input /paká:/ or not. Whatever is necessary for a ranking to map /paká:/ to paká: must hold of the target ranking.

Thus, phonotactic learning can be performed using Multi-Recursive Constraint Demotion (Tesar 2004), or MRCD, along the following lines. For the output paká:, the learner constructs the identity candidate input, /paká:/, and tests to see if some other candidate (with a different output) for that input might beat the identity candidate. Such an alternative candidate has input /paká:/ and output paká, where the second vowel has been shortened. In order for the identity candidate, /paká:/ paká:, to be generated by a ranking, it must beat the competitor /paká:/ paká

with respect to that ranking. Comparing the two candidates, in the form of a winner-loser pair (Tesar & Smolensky 1998), allows the learner to determine ranking conditions necessary to ensure that the identity candidate (the winner) will beat the competitor (the loser). This winner-loser pair is shown in (8).

Figure (8) here.

The logical content of a winner-loser pair is that at least one of the constraints preferring the winner (labeled with W in the bottom row) must dominate all of the constraints preferring the loser (labeled with L in the bottom row). In (8), the learner has determined that ID[LENGTH] must dominate NOLONG. The representation of this content shown in the bottom row is known as an elementary ranking condition, or ERC (Prince 2002). ERCs are the individual units of ranking information that the learner acquires and works with. The learner stores the winner-loser pair, including the ERC, in a memory structure known as the support (Tesar & Prince 2007). The support is permanently retained, and more ERCs are added to the support as the learner accumulates additional ranking information.

MRCD performs this process repeatedly across observed words. MRCD adopts the general strategy of error-driven learning (Wexler & Culicover 1980), in which the learner attempts to change its grammar when it encounters a form not generated by its current grammar. Each time it encounters a word, MRCD generates a constraint hierarchy consistent with the information stored in the support, and parses the inferred input to see what is optimal. Whenever a candidate different from the observed is optimal, MRCD adopts that candidate as a loser, the observed candidate as the winner, creates a new winner-loser pair, and adds it to the support. Each added winner-loser pair further restricts the space of rankings under consideration.

In this fashion, the learner accumulates a total of three winner-loser pairs of phonotactic ranking information for L20, shown in (9). The leftmost column indicates one of the morphological words of L20 that gives rise to the output of that winner.

Figure (9) here.

The procedure of Biased Constraint Demotion (Prince & Tesar 2004), or BCD, which enforces restrictiveness with a bias towards ranking faithfulness constraints low, when applied to the support in (9), generates the ranking in (10). In language L20, long vowels never appear in unstressed position. By ranking the markedness constraint WSP (which is violated by unstressed long vowels) high and the faithfulness constraint ID[LENGTH] low, the systematic absence of unstressed long vowels is imposed by the resulting grammar.

Figure (10) here.

Linguistically, the significance of this information can be expressed by two observations. The first two pairs contain the outputs páka and paká. Together, they reveal the first observation: stress must be contrastive in at least some environments. This is revealed when the two pairs are combined: the faithfulness constraint ID[STRESS] must dominate both of the markedness constraints MAINLEFT and MAINRIGHT.

The third pair, which was shown in (8), has output paká:, and reveals the second observation: length must be contrastive in at least some environments. This is revealed by only a single output due to a property of the system: there are markedness constraints that can be violated by the presence of long vowels (NOLONG, WSP), but no markedness constraints that are violated by the presence of short vowels. Thus, for any output in a language containing a long vowel, there must also be an output in the language containing a short vowel in place of the long

vowel that is otherwise the same. The presence of paká: in a language implicitly entails the presence of paká in that language.

The term "phonotactic learning" suggests that the learner is directly learning the phonotactics of the language. This is a bit misleading. What the learner is actually learning is that certain contrasts must be preserved, based on the observation of contrasts between entire words. It is representing, via the accumulated ranking information, what the phonotactics *cannot* be. Stress cannot be neutralized to a predictable position, because contrasting stress realizations are observed. Length cannot be neutralized to only short vowels, because long vowels have been observed. Phonotactic learning results in ranking information regarding the preservation of underlying contrast.

The phonotactic ranking information in (9) is genuine progress, but falls well short of the entire grammar. It does not determine the default location of stress, which requires knowing the relative ranking of MAINLEFT and MAINRIGHT. With identity mappings, the input always contains exactly one stress (matching the output), and because ID[STRESS] dominates both MAINLEFT and MAINRIGHT, faithfulness to the input will always decide stress location for these inputs. In order to observe the interaction of MAINLEFT and MAINRIGHT, and learn the location of default stress, the learner will need knowledge of unfaithful mappings, where the input contains no stresses, or possibly multiple stresses. Such knowledge of unfaithful mappings in turn requires learning at least some underlying feature values.

4.2 From Phonotactic to Non-Phonotactic Learning

The view of acquisition adopted here is that the advancement to non-phonotactic learning occurs as the learner hypothesizes morphemes as components of words, and starts inferring

morphological relations between words, for example that the words “cats” and “dogs” share a morpheme marking plural. When a morpheme has different surface realizations in different contexts, such as the plural being realized as [s] in “cats” but as [z] in “dogs”, the learner has evidence that an unfaithful mapping is being realized in (at least) one of those cases: the underlying form of the plural cannot be identical to both surface realizations, because the surface realizations are not identical to each other. Morphemic identity across environments is necessary for the learner to access evidence of unfaithful mappings, and it is precisely what is lacking in pure phonotactic learning.

The advent of morphological awareness is modeled very crudely in the ODL as currently conceived: once phonotactic learning has completed, the learner is simply provided with the morphological constituency of each word, including information on which segments of the surface form of a word are affiliated with each morpheme in the word. In the long run, an adequate account of language acquisition will need to include a theory of how the learner determines the morphemic composition of words, a non-trivial activity which will likely include analysis of semantic as well as phonological relations between words. The present work is focused more narrowly on the question of how a learner could use whatever morphological knowledge it has acquired in the process of learning the phonology of the language.

Non-phonotactic learning involves learning both underlying feature values and those aspects of the ranking requiring evidence from unfaithful mappings. The two are directly connected. The learner needs knowledge of underlying feature values in order to determine where and how features are unfaithfully realized on the surface, in order to gain non-phonotactic ranking information. However, there are features whose underlying values may only be inferred

given the knowledge of certain non-phonotactic ranking information. The two kinds of information must be learned in tandem. The learner learns a few underlying feature values using purely phonotactic ranking information. Those underlying feature values then permit the learning of some non-phonotactic ranking information, which in turn permits the learning of additional underlying feature values, and so forth.

4.3 Learning Underlying Feature Values

The definition of an output-driven map is based on entailment between mappings: the presence of A→X in a map entails the presence of B→X in the map when B→X has greater similarity than A→X. When examining the learning of underlying feature values, it helps to think of the entailment relation in its logically equivalent, contrapositive form: the non-presence of B→X in the map entails the non-presence of A→X in the map. If B cannot map to X, then no strictly less similar input can map to X.

The use of this relation for the learning of underlying forms can be illustrated by considering the length feature of suffix s4 in L20. The observed output of r1s4 is [paká:]. In this environment, the surface realization of suffix s4 is [ká:], with the surface value of the length feature +long. What is the underlying value of the length feature for s4? The learner can test the length feature of s4 by constructing a candidate with just a single disparity relative to that output, a disparity in the length feature of the suffix. That candidate is /paká/→[paká:], with the output of r1s4 and only a disparity in the suffix length. The full relative similarity lattice for [paká:] is given in (11). The shaded sublattice consists of all the candidates that have a disparity with the output for the suffix length feature: they all have s4 underlyingly –long. The single disparity

candidate /paká/→[paká:] has a subset of the disparities of all of the other candidates in the sublattice, and thus has greater similarity than all of them.

Figure (11) here.

The payoff comes if the learner is able to determine that the candidate /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 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 inconsistency of the single disparity candidate, /paká/→[paká:], is easily detected using MRCD, similar to what the learner does in phonotactic learning. The learner attempts to find a ranking that is consistent with the learner's support, and that makes the candidate optimal. If no such ranking exists, then the candidate is inconsistent with what the learner's existing knowledge. The tableau in (12) has the single-disparity candidate as the winner; the indicated loser was selected because it is optimal given a ranking generated from the learner's support. The ERC in the bottom row shows that the winner is in fact harmonically bound by the loser; the loser will beat the winner no matter what the ranking is, because two constraints prefer the loser, and none prefer the winner.

Figure (12) here.

An underlying feature may be set to a value when the alternative value is unworkable. The linguistic interpretation of this is that an underlying feature value may be set with reference

to an environment where that feature is contrastive: changing the feature to a different value changes the resulting output. Features are set when they express contrast.

The benefit from output-drivenness here is one of computational efficiency. The single disparity candidate, /paká/→[paká:], acts as a proxy for the entire sublattice of candidates with an underlying value of –long for s4. Even though half of the possible candidates have the suffix feature value –long underlyingly, the learner does not need to evaluate all of them, 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 candidates is exponential in the number of features, but the number of candidates to actually be tested is linear in the number of features. For a word in the stress/length system with 4 unset features, the number of possible candidates is $2^4 = 16$, while the number of candidates that actually need to be tested is 4. Referring back to the beginning of section 1, for a word consisting of 2 morphemes, each morpheme having three segments, each segment having 5 binary features (all unset), that word has a total of 30 unset features. The number of candidates for that word is $2^{30} = 1{,}073{,}741{,}824$; the number of candidates to be tested is 30.

4.4 Learning Non-Phonotactic Ranking Information

Once a feature has been set for a morpheme, 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 L20, such a word is r3s4 [páka].

The lattice of candidates for the output of r3s4 is shown in (13). Because s4 has been set to be +long, none of the inputs that have s4 as –long are still under consideration; they are not viable, and are marked with shaded diamonds. The viable inputs, the ones with s4 as +long, are the non-shaded ovals. Notice that the viable candidates 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 output of the word. This can be called the minimum disparity candidate. In (13), the minimum disparity candidate, /páka:/→[páka], has only one disparity, the one involving the length of s4.

Figure (13) here.

Although the significance of the minimum disparity candidate is best understood in terms of the similarity lattice, it is not at all necessary for the learner to computationally construct the entire lattice to obtain the minimum disparity candidate. In fact, the concept of minimum disparity candidate is also present in phonotactic learning: when no underlying features have been set, the minimum disparity candidate has every underlying feature match its surface feature, resulting in an identity candidate. In the logic of output-driven maps, identity candidates in phonotactic learning are one special case of the general concept of minimum disparity candidates.

As with phonotactic learning, the minimum disparity candidate has greater similarity than any other candidate in the sublattice. The learner doesn't know which candidate of the sublattice is the "true" one for this word (r3s4), but the learner does know that the minimum disparity (top) candidate is grammatical. This candidate 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 candidate by (again) using MRCD. This is summarized in (14). The known non-faithful candidate, /páka:/→[páka], is adopted as the winner; the loser is generated by parsing the input /páka:/. Comparing the winner and loser yields the ERC listed in the fourth row of (14).

Figure (14) here.

We can get a sharper sense of what new information has been obtained by combining this new ERC with a previously obtained phonotactic ERC, shown in the fifth row. The phonotactic ERC indicates that ID[LENGTH] dominates NOLONG. Taking the fusion of the two ERCs (Prince 2002), shown in the last row, reveals what the learner has obtained: WSP must dominate both NoLong and ID[LENGTH]. The conclusion that the faithfulness constraint ID[LENGTH] must be dominated relies on the unfaithful element of the winner, the disparity in suffix length.

The new ERC, along with the prior phonotactic ERC, combine to provide a partial picture of the desired ranking: WSP ≫ ID[LENGTH] ≫ NOLONG. The linguistic significance of this ranking information is that vowel length is neutralized in unstressed position; long vowels are shortened when unstressed. Phonotactically, the learner could (in principle) make the distributional observation that it has never seen an unstressed long vowel on the surface. This new, non-phonotactic, ranking information indicates how that pure surface dictum is enforced in the map: vowels that might otherwise surface as long in unstressed environments are shortened.

Non-phonotactic ranking information is obtained when the learner gets evidence of unfaithful mappings. In an unfaithful mapping, a feature is neutralized, by having its underlying value changed on the surface. If we changed the underlying value, the surface value would not be changed, and the very same output would result: the output will be the same no matter which

value of the feature is used underlyingly (in that environment). The linguistic interpretation of this is that non-phonotactic ranking information comes from evidence of neutralization.

The learner was able to obtain the new ranking 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), just as it was able to set the length feature for s4 without knowing the entire ranking. This is the key to learning the map and the lexicon together. Output-driven map structure allows these interleaving steps to be performed with great efficiency.

4.5 Contrast Pairs

The previous two subsections showed how the structure of output-driven maps can be exploited to learn underlying feature values and non-phonotactic ranking information. The described process used a single word at a time, and is named single word learning. The ranking information that can be obtained about L20 from the combination of phonotactic learning and single word learning is shown in (15), and the corresponding lexical information that can be so learned is shown in (16).

Figure (15) here.

Figure (16) here.

While much has been learned, both ranking and lexicon are incomplete. The relative ranking of MAINLEFT and MAINRIGHT, which determines the default location of stress, has still not been determined. In the lexicon, most of the length features have been set, but none of the stress features have been set. Further progress requires that this learner process more than one word simultaneously. Specifically, the learner will process a contrast pair (Tesar 2006a; Tesar 2006b). A contrast pair consists of two words, both of which contain a particular morpheme,

such that 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 in both words; no matter what underlying value is chosen, it will create a disparity with one of the words of the set. If the feature that alternates has not yet been set, then the contrast pair makes it possible to force a disparity for that feature (despite not yet knowing what the underlying value actually is). That is the source of the power of a contrast pair: the additional disparity forced by the alternation allows the learner to peer more deeply into the structure of the language.

At this point in learning, an informative contrast pair is r1s1, páka, paired with r1s3, paká, as shown in (17). The learner's lexical entries for the relevant morphemes are also shown, with four of the six features still unset. 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 (the alternating feature) is unset in the current lexicon. The suffixes, s1 and s3, are what distinguish the two words morphologically. Because the two words have different surface forms, they must come from different inputs. The only thing that differs between the inputs are underlying forms for s1 and s3; therefore, the difference in the surface forms must be the result of underlying contrast between s1 and s3. The suffixes s1 and s3 are the contrasting morphemes of the contrast pair.

Figure (17) here.

The value of a contrast pair comes in being able to set the underlying value of one (or more) of the features of the contrasting morphemes. The contrast in the surface realizations of the two words must be a consequence of underlying contrast between the contrasting morphemes. When the learner tests an underlying feature value to see if it is inconsistent, it is

looking for a ranking in which both words surface correctly. The greater demands on the ranking increase the chances that the feature disparity being tested will prove to be inconsistent with any viable ranking (allowing the learner to set the tested feature).

Testing the value of an unset feature in a contrast pair, for example the stress feature for s3, is slightly more complicated than with only a single word. The learner already knows that the value matching s3's surface realization in r1s3, +stress, will be consistent. Setting the stress feature for s3 here requires showing that the other value, –stress, is inconsistent. Because there is no single faithful value of the stress feature for r1 (it alternates across the two words), the learner needs to test the –stress value for s3 with *both* values for the stress feature of r1. This means testing two pairs of inputs: the inputs {r1s1 /paka/ r1s3 /paka/} for r1 with –stress, and {r1s1 /páka/ r1s3 /páka/} for r1 with +stress (note that both pairs have s3 set to –stress in the second word). The corresponding lexical hypotheses being tested by the two pairs of words are shown in the two rows of (18).

Figure (18) here.

The justification for testing these two hypotheses (and only these two) lies, again, in the structure imposed by output-drivenness. Because the two words of the contrast pair are being processed simultaneously, the learner is implicitly relying on a joint similarity order on pairs of candidates for the two words. As shown in (19), each node of this order contains the inputs for two candidates: the top form is the input for r1s1, and the bottom for r1s3. The order divides into two connected suborders, each a lattice. The left suborder has candidates in which r1 has the value –stress underlyingly, while the right suborder has candidates in which r1 has the value +stress underlyingly. The top nodes of the suborders contain the minimal number of disparities

for each suborder respectively: each has only a single disparity, the one forced by the adopted underlying value if stress for r1. A candidate one level down thus has two disparities, the one forced by stress on r1 plus one other one. The shaded nodes in each suborder are the hypotheses in which s3 is assigned –stress underlyingly. The top shaded nodes of the two suborders are precisely the two hypotheses to be tested, the ones represented in (18). Because of output-drivenness, each top shaded node can stand for its entire shaded sublattice. If the two top shaded nodes are both inconsistent, then all shaded nodes must be inconsistent, and therefore all possible hypotheses with s3 assigned –stress are inconsistent.

Figure (19) here.

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. It shouldn't be hard to see why both pairs prove to be inconsistent. In both pairs, the underlying forms for s1 and s3 are identical. If the underlying forms are identical, then they cannot cause the necessary contrast between the surface forms of the two words. In the environment created by root r1, s1 must surface unstressed, and s3 must surface stressed. Output-drivenness entails that because s1 surfaces unstressed in r1s1, an underlying value of –stress for s1 will work for that word. A underlying value of –stress for s3 eliminates the necessary contrast between s3 and s1, regardless of the underlying form for r1. Thus, s3 must be underlyingly +stress. Correspondingly, s1 must be underlyingly –stress, and testing that feature will successfully set it as well. Setting a feature with a contrast pair makes the role of contrast more apparent: the feature being set expresses a necessary contrast between the two words of the pair.

Computational complexity becomes an issue when there are multiple alternating features within a contrast pair. Both 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 contrast pair* under consideration.[1]

4.6 More Ranking Information

Having set the underlying value of stress for the suffix s3, the learner can again seek non-phonotactic ranking information by hunting for a word in which s3 surfaces as unstressed. Such a word is r3s3, which surfaces as páka, thus unfaithfully realizing s3's +stress feature. The winner-loser pair that is generated for r3s3 is shown in (20).

Figure (20) here.

The learner has now determined that MAINLEFT ≫ MAINRIGHT, indicating that default stress is on the initial syllable. The minimum disparity input for r3s3 has both input syllables stressed; this neutralizes faithfulness to stress (IDENT[STRESS]), leaving it to the markedness constraints to distinguish winner from loser.

This additional piece of ranking information allows the learner to set the stress features for all of the other morphemes via a second round of single word learning. This again highlights the map/lexicon learning interaction. The contrast pair allowed the learner to set one stress feature, which made it possible to learn the crucial ranking information regarding default stress, which then allowed the learner to set the other stress features.

[1] For further discussion of an algorithm for selecting contrast pairs, see (Tesar 2014: 334-36).

4.7 The Learned Grammar

Once the learner has set the rest of the stress features, the learning of L20 is complete. The final support is shown in (21), and a final ranking of the constraints generated from the support is shown in (22). The final learned lexicon is shown in (23).

Figure (21) here.

Figure (22) here.

Figure (23) here.

Notice that the length features of s1 and s2 have not been set. This is because those features are not contrastive. Recall that, in L20, s1 and s2 are homophonous: they surface identically in every environment. This is because stress can only be moved from the default location (initial) onto the suffix if the suffix is stressed underlyingly. Suffixes s1 and s2 are set to –stress underlyingly, so they will never be stressed. Long vowels are shortened in unstressed position, so if s1 or s2 is underlyingly +long, it will always shorten anyway. Either value of the length feature will produce identical results, for both s1 and s2.

5. Discussion

5.1 Paradigmatic Information in Learning

Paradigmatic information is essential for learning the non-phonotactic aspects of a grammar. The ODL makes crucial use of both fundamental types of paradigmatic information, and each plays a distinct role.

Morphemic contrast, when two morphemes produce different outputs when put in the same environment, provides information about underlying feature values. If two words contrast

on the surface, their linguistic inputs must be distinct. If the two words differ in only a single morpheme, then the underlying forms of those morphemes must be distinct.

Morphemic alternation, in which the same morpheme surfaces non-identically in different environments, provides non-phonotactic ranking information. When a morpheme alternates, at least one of its surface realizations must be unfaithful in some way to the morpheme's underlying form. Evidence for unfaithful mappings is what allows the learner to determine the ranking relations responsible for patterns of neutralization.

The two types of paradigmatic information are related. Phonotactic learning determines the ranking information necessary to realize the basic contrasts between entire words. That ranking information can be used to determine the underlying values of those features that are unambiguously contrastive in at least one word. Whenever contrast succeeds in setting the value of a feature, alternation of that feature creates an opportunity to learn non-phonotactic ranking information by examining a word in which the feature surfaces unfaithfully. Determining that a feature is contrastive in one environment enables the learner to confidently conclude that the feature is neutralized in a different environment. Knowledge of neutralization patterns allows the learner to pin down which features are responsible for additional contrasts.

5.2 Contrast, Unset Features and Acquisition

In the view presented here, "contrastive" or "noncontrastive" is a property of individual features in phonological context. A specific feature in a specific input is contrastive if changing the value of the feature (with the rest of the input unchanged) changes the output for that word. We extend this view to the lexicon by saying that a specific feature in a specific underlying form for a

morpheme is contrastive if there is at least one environment for that underlying form (one input that includes the underlying form) in which the specific feature is contrastive for that input.

A key observation here is that contrast is a property of individual instances of features, not necessarily of all feature instances of a certain type. In L20, length is noncontrastive in the suffixes s1 and s2, as noted above: on the surface, the length of the vowels in s1 and s2 neutralize to short in all environments. However, the length feature is contrastive for the other morphemes: for each other morpheme, there is at least one environment (one input) in which the length feature of that morpheme is contrastive. It is underinformative to declare that length is "contrastive" in the language. This is directly reflected in the final lexicon returned by the ODL: features are set if they are contrastive in at least one environment that is observed by the learner in its data, and they remain unset otherwise.

Under this view, it is not significant linguistically if, in a mature grammar, truly non-contrastive features are assigned a default value, as the same outputs will result regardless of what value is used for those features. The situation would be more complicated if a morpheme has not been observed in a contrastive environment for a contrastive feature. The present work has little to say about how this would be handled in language production, but such situations have been studied, for example the examination of final devoicing in Dutch by Ernestus & Baayen (2003).

Perhaps more interestingly, the situation would be expected to arise during the course of acquisition. The present work, combined with a specific linguistic analysis, potentially has a lot to say about what features are like to be set, and when. Connecting this to acquisition data, such as work on devoicing in Dutch learners (Kerkhoff 2007), will be quite non-trivial, involving a

combination of commitments on both linguistic analysis and the performance production treatment (both adult and child) of unset features.

6. Conclusion

Output-driven maps provide 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. The speed-up exhibited by the ODL appears in both the learning of underlying forms and the learning of ranking information. The ODL successfully leverages both forms of paradigmatic information, contrast and alternation, in learning. The theory of output-driven maps provides the kind of structure necessary to account for the efficiency of child language learning.

Acknowledgements

I would like to thank the organizers and participants of GALANA 6, at the University of Maryland in 2015, for helpful comments and feedback. Over the past several years, this work has benefited from conversations with Crystal Akers, Eric Baković, Karen Campbell, Paul de Lacy, Jane Grimshaw, Fernando Guzman, Brett Hyde, Gara Jarosz, John McCarthy, Nazarré Merchant, Alan Prince, Jason Riggle, and Paul Smolensky. Some of the figures and examples in this paper are reprinted with the permission of Cambridge University Press, having previously appeared in (Tesar 2014), and are © Bruce Tesar 2014.

References

Akers, Crystal. 2012. Simultaneous Learning of Hidden Structures by the Commitment-Based Learner. New Brunswick: Rutgers University dissertation.

Apoussidou, Diana. 2007. The Learnability of Metrical Phonology. Amsterdam: University of Amsterdam dissertation.

Apoussidou, Diana & Paul Boersma. 2003. The learnability of Latin stress. *Proceedings of the Institute of Phonetic Sciences* 25. 101-48.

Boersma, Paul. 1998. *Functional Phonology*. The Hague: Holland Academic Graphics.

—. 2003. Review of Tesar & Smolensky (2000). *Phonology* 20. 436-46.

Chomsky, Noam. 1964. *Current Issues in Linguistic Theory*. The Hague: Mouton.

—. 1981. *Lectures on Government and Binding.* Dordrecht: Foris.

Chomsky, Noam & Morris Halle. 1968. *The Sound Pattern of English.* New York City: Harper & Row.

de Lacy, Paul (ed.). 2007. *The Cambridge Handbook of Phonology*. Cambridge: Cambridge University Press.

Dresher, B. Elan. 1999. Charting the learning path: Cues to parameter setting. *Linguistic Inquiry* 30. 27-67.

Dresher, B. Elan & Jonathan Kaye. 1990. A computational learning model for metrical phonology. *Cognition* 34. 137-95.

Ernestus, Mirjam & Harald Baayen. 2003. Predicting the unpredictable: Interpreting neutralized segments in Dutch. *Language* 79. 5-38.

Goldsmith, John. 1994. A dynamic computational theory of accent systems. In Jennifer Cole & Charles Kisseberth (eds.), *Perspectives in Phonology*, 1-28. Stanford: CSLI.

Gupta, Prahlad & David Touretzky. 1994. Connectionist models and linguistic theory: Investigations of stress systems in language. *Cognitive Science* 18. 1-50.

Hale, Mark & Charles Reiss. 1997. Grammar Optimization: The simultaneous acquisition of constraint ranking and a lexicon. Ms. Concordia University, Montreal.

Hayes, Bruce. 2004. Phonological acquisition in Optimality Theory: The early stages. In René Kager, Joe Pater & Wim Zonneveld (eds.), *Constraints in Phonological Acquisition*, 158-203. Cambridge: Cambridge University Press.

Heinz, Jeffrey. 2009. On the role of locality in learning stress patterns. *Phonology* 26. 303-51.

Jarosz, Gaja. 2006. Rich Lexicons and Restrictive Grammars - Maximum Likelihood Learning in Optimality Theory. Baltimore, MD: The Johns Hopkins University dissertation.

—. 2013. Learning with hidden structure in Optimality Theory and Harmonic Grammar: beyond Robust Interpretive Parsing. *Phonology* 30. 27-71.

Johnson, Mark. 1984. A Discovery Procedure for Certain Phonological Rules. In *The Tenth International Conference on Computational Linguistics / Twenty-Second Annual Conference of the Association for Computational Linguistics*, 344-47. Association for Computational Linguistics.

Kager, Rene. 1999. *Optimality Theory.* Cambridge: Cambridge University Press.

Kager, Rene, Joe Pater & Wim Zonneveld (eds.). 2004. *Constraints in Phonological Acquisition*. Cambridge: Cambridge University Press.

Kerkhoff, Annemarie. 2007. Acquisition of Morpho-Phonology: The Dutch Voicing Alternation. Utrecht: Utrecht Institute of Linguistics dissertation.

Kiparsky, Paul. 1971. Historical linguistics. In W. O. Dingwall (eds.), *A Survey of Linguistic Science*, 576-649. College Park: University of Maryland Linguistics Program.

—. 1973. Abstractness, opacity and global rules (Part 2 of "Phonological representations"). In O. Fujimura (eds.), *Three Dimensions of Linguistic Theory*, 57-86. Tokyo: TEC.

Kisseberth, Charles. 1970. On the functional unity of phonological rules. *Linguistic Inquiry* 1. 291-306.

McCarthy, John J. (ed.). 2004. *Optimality Theory in Phonology*. Malden, MA: Blackwell.

—. 2007. Derivations and levels of representation. In Paul de Lacy (eds.), *The Cambridge Handbook of Phonology*, 99-117. Cambridge: Cambridge University Press.

McCarthy, John J. & Alan Prince. 1993. Generalized alignment. In Geert Booij & Jaap Van Marle (eds.), *Yearbook of Morphology*, 79-154. Dordrecht: Kluwer.

—. 1995. Faithfulness and Reduplicative Identity. In Jill Beckman, Laura Walsh Dickey & Suzanne Urbancyzk (eds.), *University of Massachusetts Occasional Papers 18: Papers in Optimality Theory*, 249-384. Amherst, MA: GLSA, University of Massachusetts.

Merchant, Nazarré. 2008. Discovering underlying forms: Contrast pairs and ranking. New Brunswick: Rutgers University dissertation.

Merchant, Nazarré & Bruce Tesar. 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.

Pearl, Lisa. 2011. When unbiased probabilistic learning is not enough: acquiring a parametric system of metrical phonology. *Language Acquisition* 18. 87-120.

Prince, Alan. 1990. Quantitative consequences of rhythmic organization. In Karen Deaton, Manuela Noske & Michael Ziolkowski (eds.), *CLS26-II: Papers from the Parasession on the Syllable in Phonetics and Phonology*, 355-98. Chicago, IL: Chicago Linguistics Society.

—. 2002. Entailed Ranking Arguments. Ms. Rutgers University, New Brunswick.

Prince, Alan & Paul Smolensky. 2004. *Optimality Theory: Constraint interaction in generative grammar*. Malden, MA: Blackwell.

Prince, Alan & Bruce Tesar. 2004. Learning phonotactic distributions. In René Kager, Joe Pater & Wim Zonneveld (eds.), *Constraints in Phonological Acquisition*, 245-91. Cambridge: Cambridge University Press.

Rosenthall, Sam. 1994. Vowel/glide alternation in a theory of constraint interaction. Amherst: University of Massachusetts dissertation.

Tesar, Bruce. 2004. Using inconsistency detection to overcome structural ambiguity. *Linguistic Inquiry* 35. 219-53.

—. 2006a. Faithful contrastive features in learning. *Cognitive Science* 30. 863-903.

—. 2006b. Learning from paradigmatic information. In Christopher Davis, Amy Rose Deal & Youri Zabbal (eds.), *Proceedings of the 36th Meeting of the North East Linguistics Society*, 619-38. GLSA.

—. 2007. Learnability. In Paul de Lacy (eds.), *The Cambridge Handbook of Phonology*, 555-74. New York City: Cambridge University Press.

—. 2014. *Output-Driven Phonology*. Cambridge: Cambridge University Press.

Tesar, Bruce, John Alderete, Graham Horwood, Nazarré Merchant, Koichi Nishitani & Alan Prince. 2003. Surgery in language learning. In G. Garding & M. Tsujimura (eds.), *Proceedings of the Twenty-Second West Coast Conference on Formal Linguistics*, 477-90. Somerville, MA: Cascadilla Press.

Tesar, Bruce & Alan Prince. 2007. Using phonotactics to learn phonological alternations. In Johnathon E. Cihlar, Amy L. Franklin, David W. Kaiser & Irene Kimbara (eds.), *Proceedings of the Thirty-Ninth Conference of the Chicago Linguistics Society (2003), vol. II: The Panels*, 209-37.

Tesar, Bruce & Paul Smolensky. 1996. *Learnability in Optimality Theory (long version).* The Johns Hopkins University Technical Report JHU-CogSci-96-3.

—. 1998. Learnability in Optimality Theory. *Linguistic Inquiry* 29. 229-68.

—. 2000. *Learnability in Optimality Theory.* Cambridge, MA: MIT Press.

Wexler, Kenneth & Peter Culicover. 1980. *Formal Principles of Language Acquisition.* Cambridge, MA: MIT Press.

Figure Captions

Figure (1): The definition of an output-driven map.

Figure (2): Candidate B→X has greater similarity than A→X.

Figure (3): If the first candidate is present, the other three must be also.

Figure (4): The constraints (McCarthy & Prince 1993; McCarthy & Prince 1995; Prince 1990; Rosenthall 1994)

Figure (5): L20

Figure (6): The constraint ranking for Language L20.

Figure (7): The relative similarity relation for output paká: (higher in the graph means greater similarity).

Figure (8): The ranking information content of paká: (each '*' indicates a constraint violation).

Figure (9): The phonotactic ranking information for L20.

Figure (10): The restrictive constraint ranking generated for the phonotactic ranking information of L20.

Figure (11): The relative similarity lattice for the output of r1s4, [paká:]. The shaded sublattice contains all candidates with s4 underlyingly –long.

Figure (12): /paká/→[paká:] is inconsistent.

Figure (13): The similarity lattice for r3s4. The viable inputs are the non-shaded ovals.

Figure (14): Additional ranking information.

Figure (15): Ranking information after the initial round of single word learning.

Figure (16): Lexical information after the initial round of single word learning. The listed feature order is /stress,length/, with '?' indicating an unset feature.

(1) Figure 1

A map is output-driven if, for every grammatical candidate A→X of the map,

if candidate B→X has greater similarity than A→X,

then B→X is also grammatical (it is part of the map).

(2) Figure 2

B→X	paká → paká:	Disparities: length in vowel 2
A→X	páká → paká:	Disparities: length in vowel 2, stress in vowel 1

(3) Figure 3

páká → paká:	2 disparities
paká → paká:	1 disparity
páká: → paká:	1 disparity
paká: → paká:	0 disparities (identity mapping)

(4) Figure 4

MAINLEFT	main stress on the initial syllable
MAINRIGHT	main stress on the final syllable
NOLONG	no long vowels
WSP	long vowels are stressed (weight-to-stress principle)
ID[STRESS]	IO correspondents have equal stress value
ID[LENGTH]	IO correspondents have equal length value

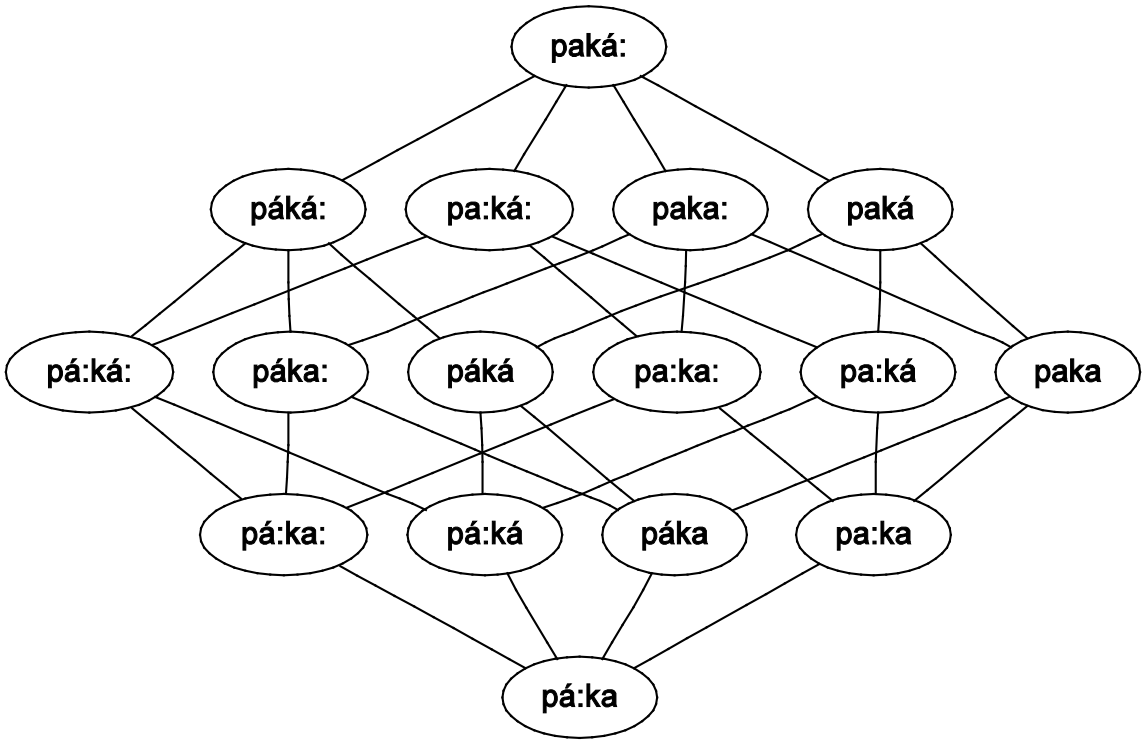
(5) Figure 5

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:/

(6) Figure 6

WSP ≫ ID[STRESS] ≫ MAINLEFT ≫ MAINRIGHT ≫ ID[LENGTH] ≫ NOLONG

(7) Figure 7


paká:
páká:
pa:ká:
paka:
paká
pá:ká:
páka:
páká
pa:ka:
pa:ká
paka
pá:ka:
pá:ká
páka
pa:ka
pá:ka

(8) Figure 8

/ paká:/	WSP	MAINL	MAINR	NOL	ID[STRESS]	ID[LENGTH]
paká: (winner)		*		*		
paká (loser)		*				*
ERC paká: ~ paká				L		W

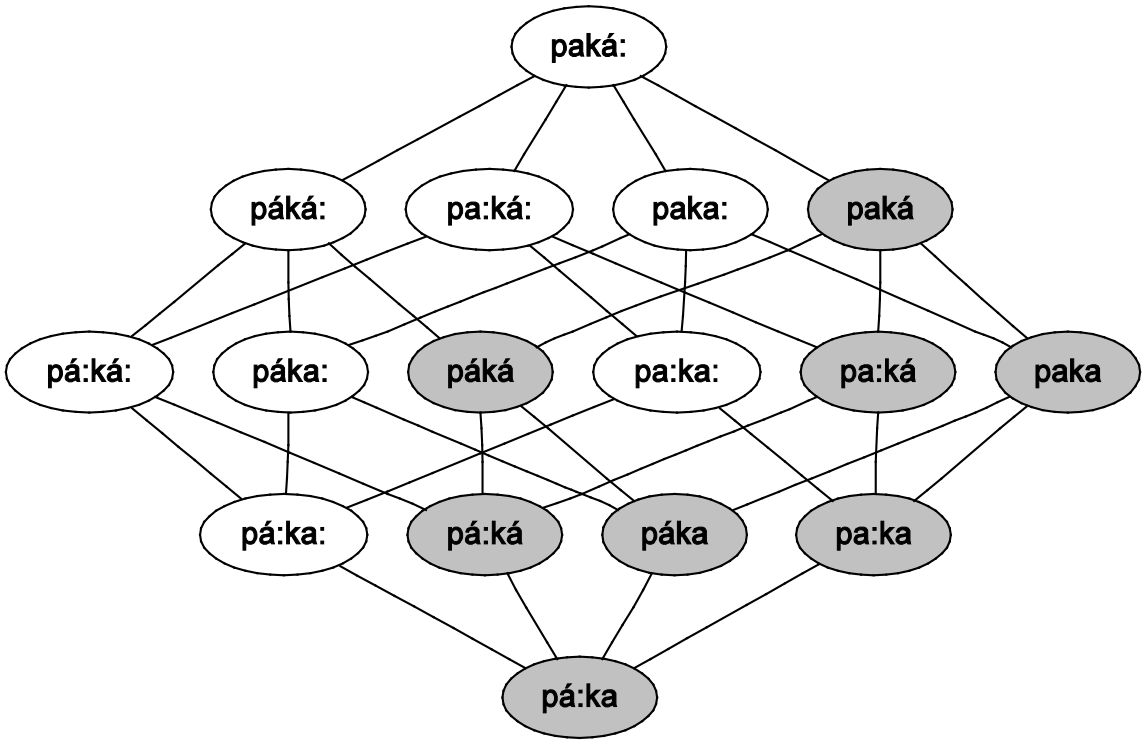
(9) Figure 9

	winner ~ loser	WSP	MAINL	MAINR	NOL	ID[STRESS]	ID[LENGTH]
r1s1	páka ~ paká		W	L		W	
r1s3	paká ~ páka		L	W		W	
r1s4	paká: ~ paká				L		W

(10) Figure 10

WSP ≫ ID[STRESS] ≫ {MAINLEFT, MAINRIGHT} ≫ ID[LENGTH] ≫ NOLONG

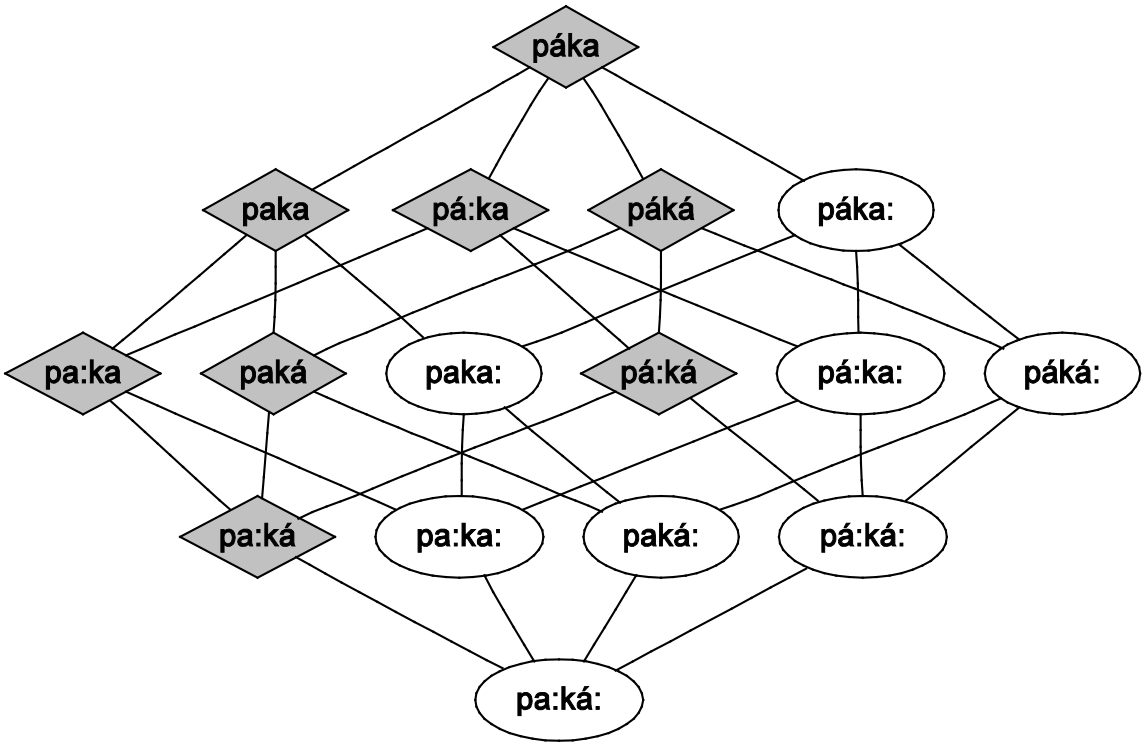
(11) Figure 11


paká:
páká:
pa:ká:
paka:
paká
pá:ká:
páka:
páká
pa:ka:
pa:ká
paka
pá:ka:
pá:ká
páka
pa:ka
pá:ka

(12) Figure 12

/paká/	WSP	MAINL	MAINR	NOL	ID[STRESS]	ID[LENGTH]
paká: (winner)		*		*		*
paká (loser)		*				
ERC paká: ~ paká				L		L

(13) Figure 13


páka
paka
pá:ka
páká
páka:
pa:ka
paká
paka:
pá:ká
pá:ka:
páká:
pa:ká
pa:ka:
paká:
pá:ká:
pa:ká:

(14) Figure 14

/ páka:/	WSP	MAINL	MAINR	NOLONG	ID[STRESS]	ID[LENGTH]
páka (winner)						*
páka: (loser)	*			*		
ERC	W			W		L
Phonotactic ERC				L		W
Fusion	W			L		L

(15) Figure 15

	winner ~ loser	WSP	MAINL	MAINR	NOL	ID[STRESS]	ID[LENGTH]
r1s1	páka ~ paká		W	L		W	
r1s3	paká ~ páka		L	W		W	
r1s4	paká: ~ paká				L		W
r3s4	/páka:/ páka ~ páka:	W			W		L

(16) Figure 16

r1	/?,–/	r2	/?,+/	r3	/?,–/	r4	/?,+/
s1	/?,?/	s2	/?,?/	s3	/?,–/	s4	/?,+/

(17) Figure 17

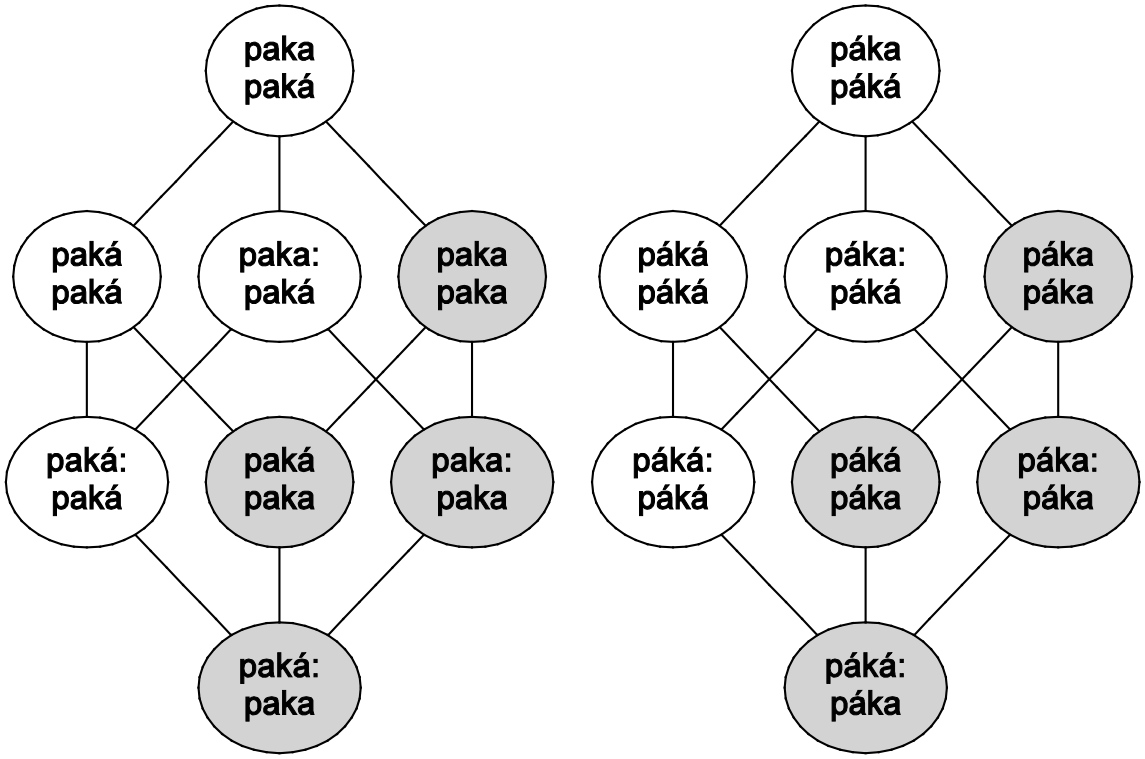
Contrast Pair:		r1s1 [páka]	r1s3 [paká]
Current Lexicon:	r1 /?,–/	s1 /?,?/	s3 /?,–/

(18) Figure 18

r1 /–,–/ s1 /–,–/ s3 /–,–/

r1 /+,–/ s1 /–,–/ s3 /–,–/

(19) Figure 19


paka paká
paká paká
paka: paká
paka paka
paká: paká
paká paka
paka: paka
paká: paka
páka páká
páká páká
páka: páká
páka páká
páká: páká
páká páká
páka: páká
páká: páka

(20) Figure 20

/páká/	WSP	MAINL	MAINR	NOLONG	ID[STRESS]	ID[LENGTH]
páka (winner)			*		*	
paká (loser)		*			*	
ERC		W	L			

(21) Figure 21

	winner ~ loser	WSP	MAINL	MAINR	NOL	ID[STRESS]	ID[LENGTH]
r1s1	páka ~ paká		W	L		W	
r1s3	paká ~ páka		L	W		W	
r1s4	paká: ~ paká				L		W
r3s4	/páka:/ páka ~ páka:	W			W		L
r3s3	/páká/ páka ~ paká		W	L			

(22) Figure 22

WSP ≫ ID[STRESS] ≫ MAINLEFT ≫ MAINRIGHT ≫ ID[LENGTH] ≫ NOLONG

(23) Figure 23

r1	/–,–/	r2	/–,+/	r3	/+,–/	r4	/+,+/
s1	/–,?/	s2	/–,?/	s3	/+,–/	s4	/+,+/