

Contrast Analysis in Phonological Learning*

Bruce Tesar
Dept. of Linguistics, Rutgers University
11/23/04

1 Introduction

It has been common, at least since the appearance of SPE (Chomsky and Halle, 1968), to idealize core phonology as a functional mapping (in the mathematical sense of function) from underlying forms to surface forms. It follows from this assumption that underlying forms realize contrasts between differing lexical items: if two surface forms are phonologically distinct, it must be the case that they have distinct underlying forms. While it is definitely not the case that any distinction between possible underlying forms is guaranteed to result in a surface distinction (neutralization is possible), any surface disparity must result from some underlying distinction.

This property generalizes from the underlying forms of entire words to the underlying forms of individual morphemes in a straightforward way. Two morphemes must have distinct phonological underlying forms if there exists at least one environment in which the morphemes have differing surface realizations. The proposals of this paper exploit the preceding observation in a specific way, toward the goal of learning phonological underlying forms for morphemes from surface data.

The idea of using surface distinctions to indicate underlying ones is hardly novel: linguists have used variations on this idea for decades in constructing analyses. Even in language learning, the idea that the learner might use surface contrasts to guide acquisition is a natural one.¹ This paper presents a specific proposal for the use of observations about surface contrasts in the inference of underlying forms, and investigates the strengths and weaknesses of that proposal. The main formal result presented here is that, if a set of assumptions are satisfied by the grammatical system, then whenever two distinct morphemes contrast on the surface in a particular environment, at least one of the underlying features on which the two differ must be

* The author has benefited greatly from discussions of this work with John Alderete, Adrian Brasoveanu, John McCarthy, Nazarré Merchant, and Alan Prince. All errors in either fact or judgment are the sole responsibility of the author. This material is based upon work supported by the National Science Foundation under Grant No. BCS-0083101 to Bruce Tesar and Alan Prince. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

¹ This idea has clear similarities to proposed principles for the acquisition of lexical semantics: Clark's Principle of Contrast (Clark, 1987), to name one, asserts that if two words differ on the surface, they must have distinct *semantic* representations. The present paper is concerned solely with phonological representations: surface phonological contrasts are indicative of distinct underlying *phonological* representations. Clark's principle is a heuristic that a learner might follow when acquiring lexical semantics; the purely phonological contrast position is a formal consequence of standard assumptions about generative grammar (whether or not learners choose to make use of it).

realized faithfully in each of the morphemes in that environment. To put it another way, at least one of the surface features distinguishing the two surface realizations must faithfully reflect a distinction between the underlying forms of the two morphemes. This property is called the Faithful Contrastive Feature property.

The significance of this property is that at least one contrast-causing feature must be overt, in the sense that its underlying values are faithfully presented in the surface forms, where the learner can observe them. This result forms the basis for an algorithm, Contrast Analysis, which starts with a number of features in underlying forms unset (not yet specified), and specifies them for a morpheme only when that morpheme contrasts with another morpheme, and the contrast can only be attributed to a single unset feature. Because the contrast-causing feature must be faithfully realized, the learner can set that feature in the underlying form of each morpheme to match its surface realization for that morpheme.

The contribution of Contrast Analysis will emerge in its role in a larger theory of language learning. We illustrate Contrast Analysis by using it as part of a procedure for initial lexicon construction, designed to construct a working lexicon for use by a subsequent procedure for jointly learning the lexicon and the phonological mapping (a constraint ranking). Initial lexicon construction depends solely on morphologically analyzed surface forms, and does not make reference to phonological mappings. Initial lexicon construction cannot determine all feature values for underlying forms in all cases. In fact, it is likely that in complex cases it will be unable to set a significant number of values for many underlying forms. If common beliefs about the close interrelation between underlying forms and grammatical mappings are correct, it is not possible in general to determine all underlying forms for the morphemes of a language independent of consideration of the grammatical mapping for the language. However, using Contrast Analysis in initial lexicon construction does offer the possibility of setting the values of some features in some morphemes early on. Those values, in turn, could do much to constrain the search involved in subsequent processes of inferring both the mapping and the rest of the underlying forms. Further, Contrast Analysis is defined such that it could be invoked at several points during learning, opening the possibility that it could be invoked during the joint learning of the lexicon and the phonological mapping.

2 The Problem: Interacting Features

Using contrasting surface forms to construct underlying forms is not transparently simple because features and feature values can interact via the grammar. Here is an example. Assume we have two monosyllabic roots, $r1 = /pa/$ and $r2 = /pa:/$, and a monosyllabic suffix, $s1 = /-ka/$. The underlying forms for $r1$ and $r2$ differ in the length of the vowel. Assume also a grammatical mapping in which stress appears by default on the final syllable of the word, but can be pulled away from the right edge by a long vowel. The two words formed by combining suffix $s1$ with each of the roots $r1$ and $r2$ have the following surface forms (stressed syllables have accent marks):

- (1) $r1s1: /pa+ka/ \rightarrow paká$
 $r2s1: /pa:+ka/ \rightarrow pá:ka$

The two words contrast in the location of stress, as well as the length of the first vowel. The key point here is that, although the roots contrast in the realization of stress in the environment of preceding s1 (r1 is unstressed, r2 is stressed), this contrast is not the consequence of a difference in the underlying specification of stress for the two roots. The stress on root r2 is a consequence of preservation of the long vowel in r2 combined with the obligation to stress long vowels. Roots r1 and r2 contrast in their underlying forms with respect to vowel length, a difference which results in surface contrasts in both vowel length and stress. The attraction of stress to long vowels causes the features to interact.

Determining what underlying distinctions should be posited in order to account for surface distinctions is not a simple matter, because of surface feature interaction. When two morphemes differ on the surface in a given environment, it is clear that the underlying forms for the morphemes must be different somehow. But the learner will have to work in order to determine which of the surface differences are direct realizations of underlying form differences (such as the difference in vowel length between r1 and r2), and which are the result of surface feature interaction (such as the difference in stress between r1 and r2).

2.1 Contrast is Context-Sensitive

It is possible, within a single grammar, for a feature to be contrastive in some environments and not in others. We can demonstrate this with the example in (2). In this example, we again have two roots with differing underlying vowel length, r1 = /pa/ and r2 = /pa:/, and two suffixes with differing underlying vowel length, s1 = /-ka/ and s2 = /-ka:/. The phonological mapping has word-final stress by default, and ranks the imperative to stress long vowels high enough that unstressed long vowels never occur on the surface. Because s2 is long and always appears in word-final position (the default position for stress), it will attract main stress, forcing any co-occurring vowels to be short.

- (2) r1s1: /pa+ka/ → paká
 r2s1: /pa:+ka/ → pá:ka
 r1s2: /pa+ka:/ → paká:
 r2s2: /pa:+ka:/ → paká:

In the environment of preceding s1, r1 and r2 surface differently, reflecting the underlying contrast in length. In the environment of preceding s2, r1 and r2 surface the same, as short and unstressed. The underlying contrast in length between r1 and r2 is not a simple global surface fact: it is subject to neutralization by the grammar. In this example, underlying length is contrastive in some environments, and not others. The learner determines which features are contrastive in which environments as part of the learning of the grammatical mapping (the constraint ranking). The learner must learn the underlying feature values for all features that are potentially contrastive in some environment (features that could possibly affect the morpheme's surface behavior).

Throughout this paper, our concern is the identification of features that serve in particular environments to realize contrast between particular forms, and we intentionally avoid any simplistic notions of a feature being contrastive in any binary, language-wide sense. This is reflected in the algorithm in the fact that all inferences about underlying forms are based on comparisons of surface realizations of morphemes in particular environments.

3 The Result: The Faithful Contrastive Feature Property

The key result of this paper is a property that holds of grammatical systems meeting certain assumptions. The property is here named the Faithful Contrastive Feature property (FCF). In systems with this property, any pair of morphemes surfacing differently in the same environment must faithfully map at least one feature value on which they differ on the surface. In other words, if two morphemes contrast phonologically, they must differ underlyingly in at least one *contrastive feature*, and that feature's values must be *faithfully* preserved in the outputs of the morphemes in the contrasting environment.

- (3) **Faithful Contrastive Feature Property (FCF)**: for any pair of morphemes surfacing differently in the same environment, there exist corresponding segments between the output realizations of the two morphemes in that environment such that: (a) there is a feature f such that the corresponding output segments have different values for f ; (b) each output segment's value for f is identical to that of its respective input correspondent.

3.1 Characterizing Environments

Key to the notion of contrast is the notion of environment: two morphemes contrast if they surface differently in the same environment. In this paper, we use a purely morphological characterization of environment: the environment of a morpheme in a word is the set of other morphemes occurring in that word. In the words r_1+s_1 and r_2+s_1 , roots r_1 and r_2 occur in the same environment, the environment of being combined with suffix s_1 . We also assume that the type of a morpheme (root, suffix, prefix) is immutable (a morpheme which is lexically specified as a suffix cannot be changed to a root via unfaithfulness).

3.2 Correspondence: Notation and Terminology

The term 'underlying form' is used in this paper to refer to underlying lexical forms for specific morphemes. The term 'input' is here used to refer to the form for an entire word, fed as input to the grammar. An input for a word is formed by the concatenation of the underlying forms for the morphemes constituting the word, with the order of concatenation determined by morphological structure.

Correspondence plays an important role in the discussion of this paper. Correspondence will be denoted with a double-arrow ' \leftrightarrow '. Multiple types of correspondence will be referred to, specifically input-output and output-output. A candidate in Optimality Theory consists of an input, an output, and a correspondence relation between the input and the output. We will use $in \leftrightarrow out$ to denote a candidate with input 'in' and output 'out'. Such candidates are defined with input-output correspondence relations. This use of the

double-arrow is distinct from the use of a heavy single arrow ‘ \rightarrow ’, which indicates the optimal output assigned by a grammar to a given input. The notation $\text{in} \rightarrow \text{out}$ asserts that, for the language under discussion, the candidate $\text{in} \leftrightarrow \text{out}$ is optimal.

The work presented here also makes use of output-output correspondence. In this paper, output-output correspondences are not used in the context of evaluating output-output correspondence constraints (Benua, 1997); such constraints are not discussed in this paper. Output-output correspondences are used when comparing the output realizations of different morphemes in the same environment. Such correspondences make it possible to identify where the output realizations of two morphemes differ, allowing the learner to identify concrete and coherent surface contrasts. Output-output correspondences are also constructed between the surface realizations of the same morpheme in different environments, allowing the learner to identify what elements of the morpheme may alternate across contexts. A correspondence relation between output forms out_1 and out_2 will be denoted $\text{out}_1 \leftrightarrow \text{out}_2$.

As will be discussed further in section 3.3, the cases discussed in this paper are restricted in such a way that only a single correspondence relation can hold between a given input and a given output, or between two given outputs. Thus, there is no need in this paper to distinguish different correspondences between two forms in the notation. In more complex linguistic systems, it is necessary to allow multiple distinct correspondences between a pair of forms, a situation discussed further in section 3.5.

3.3 Sufficient Conditions on the Grammatical System

The result presented here only guarantees that FCF holds when some strong conditions about the linguistic system are satisfied. One concerns the nature of the constraints (CON), one concerns the nature of the candidates (GEN), and one concerns the nature of segmental features.

The condition on CON requires that the only constraints that make reference to inputs are input-output segmental identity constraints, constraints which require segments in correspondence to be identical with respect to the value of some feature, in other words, IDENT constraints in the original sense (McCarthy and Prince, 1995). All other constraints must be markedness constraints, meaning that they refer only to the output, and are insensitive to the content of underlying forms. This is a very strong condition, certainly too strong for the ultimate analysis of human phonology. It eliminates traditional segmental MAX and DEP (McCarthy and Prince, 1995), constraints requiring any sort of independent feature correspondence relation, antifaithfulness (Alderete, 1999), and a variety of other input-referring constraints that have been proposed.

The condition on GEN requires that candidate outputs differ from the input only in terms of feature values, i.e. no insertion or deletion of segments in the mapping from input to output. It is assumed that for all candidates, the input – output correspondence relation is a relation between segments, and that it is an order-preserving bijection (1-to-1 and onto). GEN is also required to absolutely preserve the fixed input ordering on the morphemes themselves. The conditions on CON and GEN are related: the condition on GEN rules out

certain kinds of input/output relations and the condition on CON bans constraints that would regulate those kinds of input/output relations.

The condition on the features requires that they be binary. This turns out to be an important condition. The appendix to this paper includes an example of a system that satisfies the conditions on CON and GEN, but includes a supra-binary feature (a 3-valued feature in the example), and does not exhibit FCF. The appendix also presents some discussion of how the binary feature condition relates to a deeper understanding of how contrast is realized.

It follows from the restriction of only comparing morphemes that occur in the same environment that we will only be comparing morphemes of the same morphological type (suffixes with suffixes, roots with roots). In fact, the pairs of outputs that can be compared with each other are restricted even more severely by the requirement that the input-output correspondence relation be a bijection. If two morphemes surface with a different number of segments, it follows that they have a different number of segments underlyingly (because the input-output mapping can't change the number of segments). Therefore, two morphemes that are of the same morphological type but have different numbers of segments automatically contrast in their underlying forms in terms of the number of segments they have. The number of segments in the underlying form is a kind of "feature" that is preserved faithfully in the output realization of each (here, that faithful preservation is mandated by fiat via the sufficient conditions assumed for FCF). The intuition underlying the FCF won't tell the learner anything further. Effectively, then, the only possibly beneficial comparisons are between morphemes of the same morphological type that have identical numbers of segments.

In order to compare two outputs in the way described below, it is necessary to establish a segmental correspondence relation between them. When comparing *paká* to *pá:ka*, a segment-by-segment correspondence is constructed between them: $p_1a_2k_3á_4 \leftrightarrow p_1á:2k_3a_4$. This allows the learner to analyze differences between the output realizations of different morphemes into differences in the feature values of segments: the learner compares the value of the same feature in corresponding segments of the two outputs.

Because two morphemes are only compared in the same environment, and the input-output relation does not permit deletion or insertion of segments, the output-output correspondence between the morphemes characterizing the environment for a comparison is quite straightforward: segments in the respective outputs correspond if they both correspond to the same underlying segment for that morpheme. In comparing *paká* to *pá:ka*, the environment is defined by the suffix */-ka/*, and the occurrences of *k* in each output correspond to each other in the output-output correspondence precisely because each of them corresponds to the 'k' in the underlying form */-ka/*.² The same basis cannot be established for the morphemes being compared, because they are distinct morphemes and thus do not have the same underlying form. For present purposes, we will rely again

² The same is true of the occurrences of the vowel in the suffixes; both correspond to the underlying vowel 'a'. The fact that the vowel surfaces as stressed in *paká* and unstressed in *pá:ka* is irrelevant to the correspondence.

on the condition that faithfulness constraints are only segmental feature identity constraints, and presume that the output-output correspondence between the morphemes being compared is a simple matter of having the first segment of one morpheme correspond with the first segment of the other, the second segment corresponding with the second, and so forth.

Ongoing and future research will determine if the result initially obtained under these strong conditions will generalize in useful ways to systems permitting more complex input-output correspondence relations and input-referring constraints.

3.4 Contrast Implies Faithfulness to Some Contrastive Feature

The Faithful Contrastive Feature property holds when the conditions given above are satisfied. The significance of FCF is that a contrast-causing feature is *overt*: the learner can restrict its consideration of possible sources of a given contrast to those features that differ on the surface for that contrast.

We can see this in example (1) above. In the environment of suffix *s*, *r1* surfaces as short and unstressed, *pa*, while *r2* surfaces as long and stressed, *pá:*. They differ in two feature values: vowel length and stress. FCF implies that the surface contrast must result from a difference in the underlying specification for at least one of vowel length and stress. Under the working assumptions, the contrast could not result from a difference in the underlying specification of the voicing of the initial consonant, for instance, because *r1* and *r2* do not differ on that feature in this environment (both begin with a voiceless consonant). Note that this does not mean that initial consonant voicing must not be contrastive in the language; it certainly could be. It only claims that underlying initial consonant voicing cannot be responsible for the contrast between *r1* and *r2* in the environment *s*.

Below is a proof that FCF holds of any linguistic system meeting the conditions given above. The sections of the proof are labeled for reference in later discussion.

3.4.1 Proof

Part 1: Consider two morphemes, *m1* and *m2*, which surface differently in some environment. We'll refer to the overall word containing *m1* and the morphemes of the environment as *w1*; likewise *w2* is the word containing *m2*. Morpheme *m1* has underlying form *u1*, and Morpheme *m2* has underlying form *u2*. The input for *w1* is *i1* (*u1* combined with the underlying forms for the morphemes defining the environment), and the input for *w2* is *i2*. The output form for *w1* is *o1*, and the output form for *w2* is *o2*. By definition, *i1* and *i2* can only differ in the portions corresponding to *u1* and *u2*; the rest of the input comes from the underlying forms of morphemes that are common to both words. However, it is possible that *o1* and *o2* differ both in segments that correspond to the contrasting morphemes (*m1* and *m2*) and in segments that correspond to the environmental morphemes.

- | | | | |
|-----|---|---|-----------------------|
| (4) | <i>m1</i> has underlying form / <i>u1</i> / | <i>i1</i> = / <i>u1</i> / + /environment/ | <i>i1</i> → <i>o1</i> |
| (5) | <i>m2</i> has underlying form / <i>u2</i> / | <i>i2</i> = / <i>u2</i> / + /environment/ | <i>i2</i> → <i>o2</i> |

For ease of exposition, we will refer to corresponding segments in the output-output correspondence, as well as corresponding segments in input-output correspondences, as “a segment”, and refer to the feature values of that segment in the respective forms. By hypothesis, the output forms $o1$ and $o2$ are different by at least the value of one feature on one segment (there is at least one pair of corresponding segments in the correspondence $o1 \leftrightarrow o2$ that have different values for a feature). Given the constraint ranking defining the grammar, when input form $i1$ is used, the candidate with output $o1$ beats the candidate with output $o2$ (‘beats’ as in ‘is more harmonic than’). We will denote with $i_x \leftrightarrow o_y$ the candidate with input i_x and output o_y . When input $i1$ is used, candidate $i1 \leftrightarrow o1$ beats candidate $i1 \leftrightarrow o2$. Similarly, when input $i2$ is used, $i2 \leftrightarrow o2$ beats $i2 \leftrightarrow o1$.

Part 2: We have two winner-loser comparisons of interest: $i1 \leftrightarrow o1$ beats $i1 \leftrightarrow o2$, and $i2 \leftrightarrow o2$ beats $i2 \leftrightarrow o1$. Now consider constraints that have a preference in either of those two comparisons. A key observation is that markedness constraints only evaluate the outputs. Thus, a markedness constraint that prefers $o1$ to $o2$ will do so regardless of the input. If the highest-ranked constraint with a preference in either comparison were a markedness constraint, it would prefer the same output in both comparisons, contradicting our assumptions that $o1$ and $o2$ are distinct and optimal for $i1$ and $i2$ respectively. Thus, at least one of the comparisons must be decided by a faithfulness constraint; call that constraint F . Note that it is not mandatory that both comparisons be decided by F ; one of the comparisons could be indeterminate on F , and be decided by a constraint lower-ranking than F (see the Appendix, section 8, for a case similar to this). But of the deciding constraints in the two comparisons, F must be ranked higher than any other.

Part 3: Without loss of generality, assume that F prefers $i1 \leftrightarrow o1$ over $i1 \leftrightarrow o2$. By assumption, F evaluates identity of the value of a feature, f , for corresponding input and output segments. Any segment in which $o1$ and $o2$ share the same value of f will be irrelevant to the comparison; F will evaluate $o1$ and $o2$ the same for those segments (either both match $i1$, or both mismatch and are assessed a violation of F). For each segment on which $o1$ and $o2$ have different values for f , exactly one of them will match $i1$, and the other will be assessed a violation of F . In order for F to prefer $i1 \leftrightarrow o1$ over $i1 \leftrightarrow o2$, $o1$ must agree with $i1$ on a majority of the segments for which $o1$ and $o2$ have differing values (so that $i1 \leftrightarrow o1$ incurs fewer violations of F than $i1 \leftrightarrow o2$). Let f_diff be the set of segments³ on which $o1$ and $o2$ have conflicting values of f , and let f_diff_1 be the subset of f_diff on which $o1$ and $i1$ have the same value of f (and, by implication, on which $o2$ and $i1$ have different values). To restate, the size of f_diff_1 must be more than half the size of f_diff .

Now consider the comparison with respect to input $i2$, the comparison between $i2 \leftrightarrow o1$ and $i2 \leftrightarrow o2$. Candidate $i2 \leftrightarrow o2$ beats $i2 \leftrightarrow o1$, so faithfulness constraint F must either prefer $i2 \leftrightarrow o2$ over $i2 \leftrightarrow o1$, or be neutral (leaving it to a lower-ranked constraint to decide in favor of $i2 \leftrightarrow o2$). Let f_diff_2 be the subset of f_diff on which $o2$ and $i2$ have the same

³ Technically, f_diff contains pairs of segments that correspond in the output-output correspondence between $o1$ and $o2$, as do the sets f_diff_1 and f_diff_2 that follow.

value of f , that is, the segments for which o_1 and o_2 disagree on f and o_2 and i_2 agree (and, by implication, o_1 and i_2 disagree). The size of f_diff_2 must be at least half that of f_diff . If the size of f_diff_2 is exactly half the size of f_diff , then F assesses the same number of violations to the two candidates and does not decide. If the size of f_diff_2 is more than half the size of f_diff , then F prefers $i_2 \leftrightarrow o_2$ over $i_2 \leftrightarrow o_1$.

Part 4: Since f_diff_1 and f_diff_2 are both subsets of f_diff , and f_diff_1 is more than half the size of f_diff , and f_diff_2 is at least half the size of f_diff , it follows that f_diff_1 and f_diff_2 have a non-empty overlap: there is at least one segment that is an element of both f_diff_1 and f_diff_2 . Call such a segment $seg_contrast$. Because $seg_contrast$ is in f_diff , o_1 and o_2 differ on it. Because $seg_contrast$ is in f_diff_1 , it is a case where o_1 faithfully reflects a specification of feature f in i_1 . Because $seg_contrast$ is in f_diff_2 , it is a case where o_2 faithfully reflects a specification of feature f in i_2 . Thus, $seg_contrast$ faithfully maps its input correspondents in i_1 and i_2 .

It remains to show that $seg_contrast$ is affiliated with morphemes m_1 and m_2 , as opposed to one of the morphemes of the environment. This follows from the fact i_1 and i_2 must differ on feature f for $seg_contrast$. For each of the environmental morphemes, the input specifications are identical in i_1 and i_2 , reflecting a single underlying form for each morpheme. Thus, $seg_contrast$ must be affiliated with the contrasting morphemes m_1 and m_2 . This proves the result.

3.5 The Roles of the Sufficient Conditions

The condition requiring that faithfulness constraints be restricted to evaluating feature value identity between segmental correspondents has a couple of motivations. The restriction of input-referring constraints to those requiring identity preservation (of any kind) between input and output is necessary for the nature of FCF. If a system has a constraint that requires any input segment with feature value $+long$ to have an output correspondent with feature value $+stress$, then FCF likely will not be true in any simple sense: outputs which differ only in stress could result from underlying forms which differ only in length. In the proof, this reflected in part 3 where, having already established the existing of a deciding faithfulness constraint F (a result which does not itself depend on the faithfulness constraint condition), the proof presumes that F evaluates identity for input-output correspondents.

Beyond restricting faithfulness constraints to identity preservation, the further restriction of faithfulness constraints to only constraints evaluating feature value identity between corresponding segments (IDENT constraints) is motivated primarily by simplicity of the analysis for purposes of the proof (in other words, analyzing cases without this restriction will be extremely complicated). MAX and DEP are constraints that seek to preserve identity between input and output, but they are not IDENT constraints.⁴ Recognizing that two morphemes have different surface realizations in a given environment is a simple matter of identity of the surface realizations in the relevant output forms; either they are identical or they are not. Locating the actual disparities between two output forms

⁴ The same is true of constraints such as LINEARITY and CONTIGUITY (McCarthy and Prince, 1995).

requires establishing a correspondence between the output forms, identifying which segments are the “same” segment in the outputs. This follows from the essence of FCF itself: to claim anything (such as faithful mapping) about a feature on which two outputs differ requires a correspondence between the segments such that correspondents have different values of the feature. Without such an output-output correspondence, FCF isn’t saying anything at all. The restriction of CON to only IDENT constraints, and the co-attendant restriction of GEN banning insertion and deletion, means that there is only one correspondence between the outputs to be considered, a bijectional segmental correspondence. This has the consequence that either two outputs have the same number of segments, or they cannot be compared for contrast. The restriction also means that, for each input-output pair, there is only a single input-output correspondence to be considered, again a bijection. This condition figures heavily in part 3 of the proof, where unique output-output and input-output correspondences are assumed.

If the ban on insertion and deletion is relaxed, neither the output-output correspondence nor the input-output correspondences are guaranteed to be bijections. This turns one possible analysis of the comparison into many. In determining the underlying form for a morpheme, the learner not only has to consider possible feature values for a set of segments, it also has to consider different possible numbers of segments, as well as different possible correspondences between the hypothesized underlying form for a morpheme and each of that morpheme’s surface realizations. The learner also has to consider different output-output correspondences. If two morphemes surface in an environment as *spa* and *pat* respectively, should the learner assume a bijectional correspondence, with wildly differing feature values, or should the learner assume that the final two segments of the first output correspond to the initial two segments of the second, with the contrast being localized to the initial segment of the first output and the final segment of the second?

Proving that an FCF-like property applies even when insertion and deletion are permitted (if in fact there is such a property) would require a definition that applied to non-bijectional output-output correspondences. For instance, the existence of a segment in an output might be taken to be a faithful realization of a “root node” if that segment had a correspondent in the input. If two outputs had a non-bijectional correspondence, each segment in one output that did not have a correspondent in the other output would count as a disparity between the two outputs, one possibly attributed to differences in the inputs for the contrasting morphemes. The observations about contrast in such cases are not necessarily trivial. For instance, if two roots, *r1* and *r2*, combine with a suffix */-ka/*, to form the outputs *paka* and *spaka*, the learner could infer from an FCF-like property that the initial *s* in the second word cannot be epenthetic, and must instead have a correspondent in the underlying form for *r2*.

Proving a result for an FCF-like property along those lines would require at least demonstrating that the property holds when the correct input-output correspondences are assumed (the ones actually employed by the grammar in the analysis of the forms), and that for any pair of morphemes that occur in the same environment there exists an output-output correspondence for which the property holds. Exploiting such an FCF-like

property during learning might require the learner to be able to construct the correct output-output correspondence for the morphemes being compared.⁵ Extending the reach of FCF to linguistic systems permitting other complex input-output relationships (e.g., metathesis, coalescence) would require contemplation of an even wider range of possible correspondence relations.

The requirement that features be binary is necessary to ensure FCF (see section 8). In the proof, this requirement has an impact in part 3, where it is the basis for the assertion that “For each segment on which o_1 and o_2 have different values of f , exactly one of them will match i_1, \dots ”. If there are only two possible values for f , there can be no cases where o_1 and o_2 have different values of f and neither one matches i_1 's value for f . Otherwise, it would be possible to have an instance of feature f on which o_1 and o_2 differed, but both violated F , so that F preferred neither candidate on that instance of that feature. In that contrary case, f_diff_1 would no longer need to be more than half the size of f_diff , it would only need to be larger than the number of instances on which o_2 matched i_1 ; similarly, f_diff_2 would no longer need to be at least half the size of f_diff . All of part 4 relies on f_diff_1 being more than half the size of f_diff , and f_diff_2 being at least half the size of f_diff .

4 Exploiting FCF in Learning

This paper proposes a procedure, Contrast Analysis, for exploiting FCF in language learning, specifically in the learning of underlying forms. Section 4.1 discusses the possibilities of and limits on the use of morphemic contrast in learning. Section 4.2 gives a brief, intuitive-level presentation of the procedure. Section 4.3 discusses the various ways in which Contrast Analysis could be utilized in a larger theory of learning. Section 5 gives a more formal definition of the Contrast Analysis algorithm, along with explanatory discussion of the algorithm's definition. Section 6 gives two illustrations of the algorithm's application.

4.1 Possibilities and Limits of Morphemic Contrast

FCF guarantees that, when two comparable morphemes contrast in a given environment, one of the features on which they differ on the surface faithfully reflects a difference between the underlying forms of the morphemes. The learner can use this to set some feature values for the underlying forms of contrasting morphemes *if* it can figure out, for a given pair of contrasting morphemes, which disparity is “the” one guaranteed by FCF. If the morphemes differ in several feature values, FCF guarantees that at least one of them faithfully reflects a contrast in the underlying forms, but does not tell which ones.

This naturally leads to an interest in pairs of morphemes that differ in only one feature. If two morpheme output realizations differ in only a single feature, then that feature must be the one faithfully reflecting the underlying differences. The idea of using “minimal pairs”

⁵ Note that, unlike input-output correspondences, the output-output correspondences discussed here are purely a learning phenomenon (setting aside output-output constraints). There is nothing in the linguistic theory itself requiring that such correspondences exist or are well-defined. However, see section 7.2 for arguments for the inevitability of such correspondences in language learning.

to identify phonologically meaningful contrasts is doubtless familiar to anyone who has taken an introductory phonology course (and to many who haven't, as well). The proposal made here bears a resemblance to that idea, but goes beyond it.

For one thing, the “minimal pairs” of interest here are actually pairs of morphemes, not necessarily entire words. If a pair of roots differ by only one feature when preceding a given suffix, then they are useful for present purposes, even if the suffix is realized differently on the surface after each of the roots. This can easily happen: if you have two roots, one of which is stressed before a given suffix, the roots differ on the surface on stress, but so do the surface realizations of the suffix, which is stressed when the root is not, and vice-versa.

If the learner can identify a minimal pair, then it not only knows that the differing feature is contrastive in the given environment, but that the underlying feature specification for each morpheme matches its surface realization. The learner can thus use the minimal contrast to set the underlying value of the relevant feature for each of the morphemes.

The learner can also draw inferences about underlying forms from a contrast between morphemes with differences in more than one feature value, if the learner has independent knowledge that all but one of the features on which the surface realizations of the morphemes differ do not result from contrasts in the underlying forms. Such situations are demonstrated in Illustration 2 below.

Given the current formulation of FCF, once the learner determines (by whatever means) that a feature on which two morphemes differ in a given environment is faithfully mapped for each, the learner cannot use FCF to infer anything further on the basis of the observation of contrast between those two morphemes in that environment alone. The learner does not immediately know which other disparities (if there are any) between the surface realizations of the two morphemes in the relevant environment result from surface interactions with the already-identified differing feature. This ignorance follows from the lack, within the formulation of FCF, of any restrictions on the kinds of interactions between output features imposed by markedness constraints. This allows the use of FCF to apply to linguistic systems with a wide variety of markedness interactions, at the cost of never being able to set the underlying value for more than one feature on the basis of a contrast between two morphemes in a given environment. Of course, this does not rule out subsequently setting another feature on the basis of different surface realizations of the morphemes in other environments, or subsequently setting an underlying feature for one of the morphemes on the basis of a surface contrast with another, third morpheme.

The possible existence of linguistically motivated restrictions on surface interactions between feature values is, of course, well worth investigating, especially in the hope that such restrictions could permit a learner to confidently identify more than one faithfully mapped contrastive feature for a given pair of contrasting morphemes in a given environment.

4.2 An Intuitive Description of Contrast Analysis

Contrast Analysis is a procedure for exploiting the FCF. It accepts, as input, a working lexicon of underlying forms for morphemes, in which each feature is marked as either already permanently set or unset. It also accepts, as input, morphologically analyzed surface forms for words. It returns, as output, a working lexicon. If the algorithm has been effective, then some features marked as unset in the input lexicon will be marked as permanently set in the output lexicon; the learner will have learned more of the content of the underlying forms for the language.

This section gives a brief description of the steps of the algorithm. Illustrations of each of the cases mentioned here can be found in section 6.

The algorithm searches the morphologically analyzed words for pairs of words in which different morphemes appear in the same environment. If the words are non-identical on the surface, then the learner constructs an output-output correspondence between the words, and computes the disparities (differing feature values) between the output realizations of the two morphemes being contrasted in the two words. For each differing feature value for the two output realizations, the learner checks that feature in the underlying forms of the two morphemes to see if they are permanently set and faithful to the value of the feature in the surface realizations of the respective morphemes. If each underlying form has the feature permanently set and faithful to its surface realization, then that feature satisfies FCF, and there is nothing more for the learner to learn from the pair of words. In this instance, the underlying forms for the two morphemes would be said to faithfully map a surface-differing feature.

On the other hand, if none of the features on which the surface realization differ are faithfully mapped by the current lexicon, the learner checks to see which of the surface-differing features could possibly be set in the lexicon so that they were faithfully mapped. If a feature is permanently set in the underlying form for a morpheme to a value that does not match the surface realization for that morpheme, then that feature cannot possibly be faithfully mapped.

FCF guarantees that there will be at least one surface-differing feature that can possibly be faithfully mapped. If the learner finds only one, then by the FCF it can permanently set the value of that feature in each of the underlying forms to match its surface realization. If the learner finds more than one feature is possibly faithfully mapped, then it cannot know at this stage which one (if not both) is actually faithfully mapped. The learner conservatively declines to set any of the unset features based on this comparison, and moves on to the next pair of contrasting morphemes.

4.3 Situating it in the Larger Learning Task

Contrast Analysis works on the basis of disparities between the surface forms of morphemes in particular environments. It also works from a working lexicon provided to the algorithm. That lexicon contains underlying forms for each morpheme, marked in a specific way: each feature for each segment in each underlying form is marked as being either set (permanently) or not. In other words, it is possible that some underlying feature

values have already been set and committed to, prior to the execution of Contrast Analysis.⁶ It is also the case that Contrast Analysis, when it attributes a contrast to a particular feature in a pair of morphemes, sets the value of that feature and marks it as set permanently in the underlying forms of each of those morphemes. If Contrast Analysis is productive, then more features in the working lexicon will be set after it has applied than were set before it was applied.

Defined in this way, Contrast Analysis could be applied at any of a number of points during the learning process. It could be applied more than once during learning, and multiple applications could imaginably be productive. The illustrations in this paper will presume that Contrast Analysis takes place at a particular stage of the learning of phonological alternations. The illustrations will assume that phonotactic learning has already taken place (Prince and Tesar, 2004), and that the learner has analyzed words into their constituent morphemes, so that it knows which segments are affiliated with which morphemes in the words. The learner will then go through a stage of preliminary lexicon construction, in which it constructs a working lexicon of underlying forms for the morphemes based upon basic analysis of the differing realizations of the output forms. Contrast Analysis is a part (but not all) of this stage. Once the preliminary lexicon is constructed, the learner executes an algorithm that jointly refines both the lexicon and the ranking in tandem, such as the surgery learning algorithm (Tesar et al., 2003). One would expect the surgery learning algorithm to benefit from a lexicon that has been enhanced by Contrast Analysis, as it reduces the number of underlying features that are alterable by surgery, thus reducing the lexical search space.⁷

The initial lexicon construction stage consists of the following parts. First, the learner examines, for each morpheme, all of its output realizations in different environments, and checks to see which features alternate and which do not. Any feature which does not alternate (it bears the same feature value across all environments) will be set in the underlying form to that value, and marked as permanently set. This embodies an assumption that it is always safe (although not always necessary) to map invariant features faithfully, and was used as the principal means of initial lexicon construction in previous work with the surgery learning algorithm (Tesar et al., 2003). Features which alternate are left unset. That working lexicon is then passed to Contrast Analysis.

The two parts of the initial lexicon construction stage refer to the two basic kinds of paradigmatic information, and both require direct output-output correspondences between forms. The determination of which features alternate and which do not concern one basic

⁶ For instance, if phonotactic learning determined that a given feature was completely predictable in the target language, and thus not the source of any contrast in the language, the learner might mark all occurrences of that feature as permanently set to some default value. The learner, during Contrast Analysis, could then avoid trying to determine settings for that feature, and would also not consider attributing a surface contrast to that feature.

⁷ A previous grammatical space investigated with surgery, one involving only underlying specification of stress, is completely solvable by contrast analysis combined with error-driven constraint demotion (see (Tesar et al., 2003) for the linguistic system in question). This is because each morpheme has only one feature (stress). When two morphemes of the same type differ on the surface, the learner immediately knows for certain what feature is responsible (there is only one), and can set it appropriately.

kind of paradigmatic information: the different surface forms that a morpheme can take in different environments. It requires output-output correspondences between the surface realizations of a particular morpheme in different environments. Contrast Analysis concerns the other basic kind of paradigmatic information: the different surface forms that different morphemes take in the same environment. It requires output-output correspondences between the surface realizations of contrasting morphemes in the same environment.

It bears emphasizing that Contrast Analysis does not make reference to any constraint ranking. It is not itself engaging in error-driven learning; it is not testing any of its constructed underlying forms to see if they surface correctly under some mapping.

5 Contrast Analysis

5.1 *The Algorithm in Pseudocode*

The algorithm is given here in very rough pseudocode. This does not approach the level of detail necessary for anything like an actual computer program, it provides just enough detail to make the main algorithm clear.

Given: an input working lexicon in which some features may be marked as permanently set, and a paradigm of word output forms with morphological affiliation of each segment specified.

```
Repeat until no more features can be set
  For (each morpheme with an unset feature)
    For each other morpheme of the same morphological type
      For each environment in which they differ on the surface
        If (UFs don't already faithfully map a surface-differing feature)
          If (only one surface-differing feature could possibly be faithfully mapped)
            Set each UF to match its surface form for that feature
          End-If
        End-If
      End-For
    End-For
  End-For
End-Repeat
Return(lexicon)
```

5.2 *Explanation of the Algorithm*

The algorithm has a rather simple structure. It has four concentric layers of control loops (the Repeat and three For loops). Within that loop structure are a set of conditions and the actual operations for setting underlying feature values. The outer control loops serve largely to search for pairs of contrasting morphemes to compare, and are necessary but not terribly interesting. The heart of the algorithm, including its connection to FCF, resides in the inner If and Set statements.

5.2.1 **The Control Loops**

The outermost loop allows the learner to test contrasts of a morpheme more than once. This is useful because it is possible for a learner to be unable to learn from a comparison between two morphemes at an earlier point, because too many of the features are unset,

but be able to learn from the same comparison at a later point, because some features have been set on the basis of other comparisons.

The For-loop testing each morpheme for contrast is restricted to testing only morphemes with at least one unset feature. This is purely for efficiency, to prevent the algorithm from wasting effort comparing the output realizations of two morphemes with underlying forms that are already fully specified.

Each morpheme with an unset feature is compared, in turn, with every other morpheme of the same morphological type. In the present illustration, the theory of morphological types is restricted to roots and suffixes, but the algorithm is consistent with any linguistic theory providing a set of morphological types such that contrast between morphemes within a type may be attributed solely to phonological factors.

When a pair of morphemes is compared, they are successively compared in each environment in which they both occur; this is accomplished by the inner-most For loop. If the surface realizations of the two morphemes in a given environment are not identical, then they contrast in that environment, and there is potential for learning something from the contrast.

The efficiency of the algorithm could be improved somewhat by putting additional conditions in the inner two loops to terminate immediately if the morpheme being evaluated becomes fully specified, avoiding the wasted effort of continuing to compare that morpheme to remaining others. This would have no impact on the output returned by the algorithm, and we have chosen to leave such details out of the pseudocode description for purposes of clarity.

5.2.2 The Contrast Conditions

The outer If conditional checks to see if the underlying forms of the two morphemes are set faithfully for a feature on which the surface realizations of the morphemes differ. If they are, that feature already is capable of accounting for the contrast, so far as FCF is concerned, and the learner cannot use the contrast between the output forms to infer anything about any of the unset features in the underlying forms. Note that this condition does not merely check to see if the underlying forms for the two morphemes are specified differently for some feature, as such a feature could not account for the contrast in a particular environment if it is not faithfully realized for each of the morphemes in that environment. The condition set forth in this If statement can only be evaluated relative to a given environment in which the two morphemes occur.

The inner If conditional checks to see if there is more than one feature that could possibly satisfy FCF. A surface-differing feature is possibly faithfully mapped if, for each morpheme, the underlying feature either is not set or matches the surface value for that morpheme. If the output realizations of the morphemes differ on more than one feature, and more than one of those features are possibly (with the further setting of underlying feature values) the source of the contrast, then the learner does not yet know which to attribute to contrast to. In the current algorithm, the learner declines to guess, and does

not set any underlying features unless it has identified a unique feature to which the contrast may be attributed. Note that if there are *no* features that could potentially faithfully map to contrasting surface values, then there is an inconsistency in the data; the conditions ensuring FCF have been violated.

Each surface-differing feature evaluated by the inner If statement must either have an unset value in one of the morphemes or have an unfaithful mapping in one of the morphemes, because if the underlying value in both morphemes were set and faithfully mapped, then the comparison would have been screened out by the outer If statement.

If both conditions are met, then there is no feature already set in both forms that accounts for the contrast, and there is only one settable feature that could possibly account for the contrast. In that case, the one feature is then set, by setting the unset underlying feature value to match the surface realization for each morpheme.

6 Illustrations of the Algorithm

6.1 *The Linguistic System*

The illustrations use data generated from grammars with monosyllabic roots and suffixes. Each vowel can have two features specified underlyingly: vowel length (-length for short vowel, +length for long vowel) and stress (-stress for unstressed, +stress for stressed). Each word has exactly one stress on the surface, regardless of the number of syllables that are stressed underlyingly (this is enforced by GEN). Because each morpheme is monosyllabic, the discussion can be simplified by assigning each morpheme an underlying form consisting of a stress feature and a length feature for the vowel (leaving out any details concerning any consonants of the syllable, which are not of interest here). The illustrations below will thus refer, somewhat abstractly, to roots and suffixes that are either long or short, and stressed or unstressed.

We can textually represent a form (surface or underlying) for a morpheme with an ordered pair of values inside square brackets, one for stress and one for length, in that order. Binary features are assumed, so each specified feature has values '+' and '-'. [+,-] represents a form that is stressed and short. A surface realization of a morpheme will always be fully specified, with a plus or minus for both values. Underlying forms, however, can have features that have not yet been set. When a feature has not yet been set by the learner, the slot for that feature's value will contain 'x'. [x,+] represents an underlying form that is long and has the stress feature unset.

It should be emphasized that we are assuming that the marking of a feature as unset is assumed to be a device meaningful only to the learning algorithm, not to the grammar. We are assuming for present purposes that all proper underlying forms in adult languages are fully specified. In particular, it is assumed that a grammar cannot generate distinct surface behaviors for a morpheme on the basis of an unset feature value in the form (distinct from an otherwise identical underlying form in which the feature has one of the values specified for the feature).

The languages used in the illustrations were generated using Optimality Theoretic grammars involving six constraints. We won't focus on the constraint rankings involved, because Contrast Analysis does not itself refer to constraints or rankings.

6.2 Illustration 1

The first illustration uses a paradigm with four roots and three suffixes. In the grammar used to generate the surface data, surface long vowels are always stressed. Stress is initial by default, but underlying stress overrides default stress placement. Underlying vowel length is preserved only in syllables that are stressed on the surface.

The system only has three suffixes because the underlying suffix forms /-s/ and /-s:/ never contrast; they are indistinguishable on the surface. This is because suffixes only receive surface stress when they are underlyingly stressed (because stress appears initially by default, on the root), and length only surfaces in stressed position. /-s/ and /-s:/ aren't underlyingly stressed, and so are never stressed on the surface, so the underlying length distinction never surfaces. We have chosen to list the correct underlying form as short, because the morpheme invariably surfaces as short, in keeping with lexicon optimization (Prince and Smolensky, 1993).

Freely combining roots and suffixes gives the following paradigm, represented in a more traditional notation:

Table 1 The language for illustration 1.

	$r_1=/r/$	$r_2=/r:/$	$r_3=/r'/$	$r_4=/r':/$
$s_1=-s/$	řs	ř:s	řs	ř:s
$s_2=-s'/$	rś	rś	řs	ř:s
$s_3=-s':/$	rś:	rś:	řs	ř:s

The row and column headings give the correct underlying forms for the morphemes. The internal cells of the table show the resulting surface forms for each word (root+suffix pair).

We can represent the surface forms used by the learner with the +/- feature specifications as shown in the table below, with the stress feature listed first, followed by the length feature.

Table 2 The surface forms for Illustration 1. Features: [stress,length]

	r1	r2	r3	r4
s1	r[+,-] s[-,-]	r[+,+] s[-,-]	r[+,-] s[-,-]	r[+,+] s[-,-]
s2	r[-,-] s[+,-]	r[-,-] s[+,-]	r[+,-] s[-,-]	r[+,+] s[-,-]
s3	r[-,-] s[+,+]	r[-,-] s[+,+]	r[+,-] s[-,-]	r[+,+] s[-,-]

The working lexicon that serves as input to Contrast Analysis is constructed by setting the value of each feature which does not alternate on the surface to match its solitary surface realization. This yields the working lexicon in (6).

(6) The lexicon with only non-alternating features set. Features: [stress,length]
r1[x,-] r2[x,x] r3[+,-] r4[+,+]
s1[-,-] s2[x,-] s3[x,x]

For instance, morpheme r1 is stressed in environment s1, but unstressed in environments s2 and s3; the stress feature alternates, so it remains unset in the underlying form for r1 (r1's first feature is shown as 'x'). However, r1 has a short vowel (-length) in all three environments; the length feature for r1 does not alternate, so it is set to -length.

Contrast Analysis begins by choosing a morpheme with an unset feature, and then processes that morpheme by comparing it successively to each of the other morphemes of the same morphological type. The algorithm as given above leaves completely open the order in which the morphemes are considered; our illustration will proceed by considering morphemes in the order r1 to r4 and then s1 to s4.

6.2.1 Processing r1

Root r1, with underlying form [x,-], has an unset stress feature.

The learner compares r1 with r2. They differ in environment s1, on only one surface feature, length. The learner sets the length feature of each to match their surface forms: r1 is already -length, r2 is set to +length. The revised lexicon is given in (7).

(7) The lexicon after comparing r1 and r2 in environment s1. Features: [stress,length]
r1[x,-] r2[x,+]
s1[-,-] s2[x,-]

Note that although the learner is currently “processing” r1, this comparison actually changed the underlying form of r2, not r1. This is to be expected, as contrast is a symmetric relationship. Circumstances will vary as to which (if not both) of the morphemes will be previously unset for the feature responsible for the contrast.

r1 and r2 do not differ in environments s2 and s3. That concludes the comparison of r1 and r2. The sole surface difference between them, in environment s1, has been accounted for with an underlying contrast in length.

The learner then compares r1 with r3. They do not differ in environment s1, but they differ in environment s2, on only one feature, stress. The learner sets the underlying stress feature of each to match their surface forms: r1 is set to -stress, r3 is already +stress. The revised lexicon is shown in (8).

(8) The lexicon after comparing r1 and r3 in environment s2. Features: [stress,length]
 r1[-,-] r2[x,+]
 s1[-,-] s2[x,-]

r1 and r4 are now both fully specified, so comparing them doesn't change anything. This concludes the processing of morpheme r1.

6.2.2 Processing r2

Root r2, with underlying form [x,+], has an unset stress feature. Note that it no longer has an unset length feature, as that was set following the comparison with root r1.

The learner compares r2 with r1. r1 and r2 differ on the surface in length in environment s1, but that feature is already faithfully mapped for each (r1 is -length, r2 is +length). The surface contrast is already accounted for. This is not surprising; the length feature was set when r1 and r2 were previously compared, during the processing of r1.

The learner compares r2 with r3. They differ on the surface in environment s2, on only one feature, stress. The learner sets the underlying stress feature of each to match their surface forms: r2 is set to -stress, while r3 is already +stress. The revised lexicon is now

(9) The lexicon after comparing r2 and r3 in environment s2. Features: [stress,length]
 r1[-,-] r2[-,+]
 s1[-,-] s2[x,-]

Root r3 is now fully specified. In fact, all of the roots are now fully specified. r4 will not be selected for processing because it is fully specified, and none of the roots will be selected on any subsequent rounds of processing, for that reason.

6.2.3 Processing s2

Suffix s2, with underlying form [x,-], has an unset stress feature.

The learner compares s2 with s1. They differ in environment r1, on only one feature, stress. The learner sets the underlying stress feature of each to match their surface forms: s1 is already -stress, s2 is set to +stress. The revised lexicon is shown in (10).

(10) The lexicon after comparing s2 and s1 in environment r1. Features: [stress,length]
 r1[-,-] r2[-,+]
 s1[-,-] s2[+,-]

s2 is now fully specified. We will have the learner stop processing s2 at this point, and move on to the next morpheme with an unset feature.

6.2.4 Processing s3

Suffix s3, with underlying form [x,x], is has both stress and length unset.

The learner compares s3 with s1. They differ on the surface in environments r1 and r2, but in each environment they differ in two features. s3 is not specified for either feature, and s1 faithfully maps both of its features, so the learner doesn't know if they contrast underlyingly in one feature, the other, or both. No features are set.

The learner compares s3 with s2. They differ in environment r1, by only one feature, length. The learner sets the underlying length feature for each to match their surface forms: s2 is already -length, s3 is set to +length. The revised lexicon is now

(11) The lexicon after comparing s3 and s2 in environment r1. Features: [stress,length]
 r1[-,-] r2[-,+]
 s1[-,-] s2[+,-]

This finishes the processing of s3 for this pass.

6.2.5 Processing s3 (again)

The learner has made one pass through the set of morphemes, processing each with an unset feature. Now the learner performs another pass, because at least one feature was set to a value during the previous pass. Setting features changes the lexicon, and could alter some comparisons to be more informative.

Only one morpheme remains with an unset feature, s3, with underlying form [x,+].

The learner compares s3 with s1. They differ on the surface in environments r1 and r2, but differ in two features. The morphemes already contrast underlyingly in one of the features, length. The previous time s3 and s1 were compared, both features of s3 were unset, and the learner could not determine which feature should account for the surface difference. Now, the contrast in underlying length can account for the surface difference, so the contrast between s3 and s1 cannot determine how to set the stress feature for s3.

The learner compares s3 with s2. They differ in environment r1 on length, but the contrast is already accounted for by the faithfully mapped difference in length (s3 is +length, s2 is -length).

No features were set for s3, and s3 is the only morpheme with an unset feature, so the algorithm terminates. The lexicon returned by Contrast Analysis is shown in (12).

(12) The lexicon returned by Contrast Analysis. Features: [stress,length]
 r1[-,-] r2[-,+]
 s1[-,-] s2[+,-]

The stress feature for s3 cannot be set by Contrast Analysis. This can be attributed in part to the fact that s3 is underlyingly long, while s1 and s2 are underlyingly short, so s3 already has a realized contrast with both of the other suffixes. It is worth noting that the failure of Contrast Analysis to set the stress feature of s3 is not a consequence of the feature being inert: in the overall language, s3 must be set to +stress, or s3 will yield the wrong surface outputs (in fact, it will be indistinguishable from s1).

6.2.6 Comments on Illustration 1

Contrast Analysis delivers a lexicon in which all but one feature has been set, and set correctly. This is a significant improvement over the initial lexicon constructed by setting only those features that never alternate, in which six features were unset (out of 14).

This illustration reveals some key properties of the algorithm. When suffix s3 is being processed the first time, and is compared to s1, the two differ on the surface in some environments, but always by two features. Because s3 is not specified for either feature, while s1 faithfully maps both features, either feature could possibly satisfy FCF, which only promises that one differing surface feature is faithfully mapped for both forms. When the algorithm cannot reliably determine which feature must be faithfully mapped, it declines to set a feature, and leaves the surface contrast unaccounted for.

When s3 is compared with s1 again, during the second processing of s3, s3 has had its length feature set, and it faithfully maps length in the environment where it differs from s1. Thus, length satisfies FCF: s1 and s3 differ on the surface in length, and each faithfully maps that feature. That is all FCF promises, so the learner does not draw any further conclusions regarding the fact that the outputs also differ on the surface in stress. FCF does not guarantee this because the difference in stress on the surface could be a consequence of the surface difference in length; the underlying contrast in length could be sufficient to account for all of the surface difference.

The phonotactics of the language conspire to block the existence of a form that could minimally contrast with s3 on stress. The key conspirators are the restriction of long vowels to only appear in stressed position, the default location of stress word-initially, and the simple fact that suffixes never appear word-initially. This means that suffixes can only ever bear surface stress if they are stressed underlyingly, and thus can only ever surface with a long vowel if they are underlyingly stressed. In this language, if a suffix is unstressed, it will always surface with a short vowel, regardless of how its length feature is set: the length feature is only contrastive in the presence of an underlying stress feature (for suffixes). Any suffix differing from s3 in stress will necessarily always surface as short, and appear for all practical purposes to be short underlyingly, thus appearing to contrast with s3 in length.

6.3 Illustration 2

The second illustration uses a paradigm with four roots and four suffixes. Like in the first illustration, the grammar used to generate the surface data has a high-ranking Weight-To-Stress principle, so long vowels only appear in stressed position. Stress is initial by default, but underlying stress overrides default stress placement. However, faithfulness to underlying length outranks faithfulness to underlying stress in this grammar, so vowel length is capable of attracting surface stress away from initial position and from underlyingly stressed syllables, as illustrated in form r3s2.

Freely combining roots and suffixes gives the paradigm in Table 3, represented in a more traditional notation.

Table 3 The language for illustration 2.

	r₁=/r/	r₂=/r:/	r₃= /r̄/	r₄=/r̄:/
s₁=/-s/	řs	ř:s	řs	ř:s
s₂=/-s:/	rś:	ř:s	rś:	ř:s
s₃= /-ś/	rś	ř:s	řs	ř:s
s₄=/-ś:/	rś:	rś:	rś:	ř:s

Switching to the +/- feature notation, the surface forms used by the learner are shown in Table 4.

Table 4 The surface forms for Illustration 2. Features: [stress,length]

	r1	r2	r3	r4
s1	r[+,-] s[-,-]	r[+,+] s[-,-]	r[+,-] s[-,-]	r[+,+] s[-,-]
s2	r[-,-] s[+,+]	r[+,+] s[-,-]	r[-,-] s[+,+]	r[+,+] s[-,-]
s3	r[-,-] s[+,-]	r[+,+] s[-,-]	r[+,-] s[-,-]	r[+,+] s[-,-]
s4	r[-,-] s[+,+]	r[-,-] s[+,+]	r[-,-] s[+,+]	r[+,+] s[-,-]

The initial lexicon that serves as input to Contrast Analysis is again constructed by setting the value of each feature which does not alternate on the surface to match its solitary surface realization. This yields the following initial lexicon:

(13) The lexicon with only non-alternating features set. Features: [stress,length]

r1[x,-]	r2[x,x]	r3[x,-]	r4[+,+]
s1[-,-]	s2[x,x]	s3[x,-]	s4[x,x]

Out of 16 features in the lexicon, 9 are unset (more than half). The learner now uses this initial lexicon as input to Contrast Analysis.

6.3.1 Processing r1

r1, with underlying form [x,-], has an unset stress feature.

The learner compares r1 with r2. They differ in environment s1, on only one surface feature, length. The learner sets the length feature of each to match their surface forms: r1 is already -length, r2 is set to +length. The revised lexicon is now

(14) The lexicon after comparing r1 and r2 in environment s1. Features: [stress,length]

r1[x,-]	r2[x,+]	r3[x,-]	r4[+,+]
s1[-,-]	s2[x,x]	s3[x,-]	s4[x,x]

r1 and r2 also differ in environments s2 and s3, but in both stress and length. r1 and r2 already contrast underlyingly in length, and those feature values are faithfully mapped. Thus, the stress features for r1 and r2 cannot be set on the basis of the contrast.

The learner compares r1 with r3. They differ in environment s3, on only one feature, stress. The learner sets the stress feature of each to match their surface forms: r1 is set to -stress, r3 is set to +stress. The revised lexicon is now

(15) The lexicon after comparing r1 and r3 in environment s3. Features: [stress,length]

r1[-,-]	r2[x,+]	r3[+,-]	r4[+,+]
s1[-,-]	s2[x,x]	s3[x,-]	s4[x,x]

r1 and r4 are now both fully specified, so comparing them won't change anything.

6.3.2 Processing r2

Root r2, with underlying form [x,+], has an unset stress feature.

The learner compares r2 with r1. In all differing environments, r2 and r1 differ in length; the underlying forms already contrast in length, and are faithfully mapped in these surface forms, so the surface difference can be accounted for already.

The learner compares r2 with r3. In all differing environments, r2 and r3 differ in length; the underlying forms already contrast in length, and are faithfully mapped in these surface forms.

The learner compares r2 with r4. They differ in environment s4, in both length and stress. However, the UFs explicitly match on length, denying a length-based contrast. So, they must contrast underlyingly in stress. The learner sets the underlying stress feature of each to match their surface forms: r2 is set to -stress, r4 is already +stress. The revised lexicon is shown in (16).

(16) The lexicon after comparing r2 and r4 in environment s4. Features: [stress,length]
 r1[-,-] r2[-,+]
 s1[-,-] s2[x,x]

r2, and in fact all of the roots, are now fully specified.

6.3.3 Processing s2

Suffix s2, with underlying form [x,x], has both stress and length unset.

The learner compares s2 with s1. They differ in environments r1 and r3, but differ in two features. s2 is not specified for either feature, and s1 faithfully maps both of its features, so the learner doesn't know if they contrast in one, the other, or both. No features are set.

The learner compares s2 with s3. They differ in environment r1, in length alone. The learner sets the length feature of each to match their surface forms: s2 is set to +length, s3 is already -length. The revised lexicon is now

(17) The lexicon after comparing s2 and s3 in environment r1. Features: [stress,length]
 r1[-,-] r2[-,+]
 s1[-,-] s2[x,+]

The learner compares s2 with s4. They differ in environment r2, but in both length and stress. s4 is set for neither feature. s2 is set to +length, but surfaces in environment r2 as short (-length), so length is not faithfully mapped. Therefore, the faithfully mapped contrasting feature must be stress. The learner sets the stress feature of each to match their surface forms: s2 is set to -stress, s4 is set to +stress. The revised lexicon is now

(18) The lexicon after comparing s2 and s4 in environment r2. Features: [stress,length]
 r1[-,-] r2[-,+]
 s1[-,-] s2[-,+]

s2 is now fully specified, and the processing of s2 is complete.

6.3.4 Processing s3

Suffix s3, with underlying form [x,-], is unset for stress.

The learner compares s3 with s1. They differ in environment r1, in stress alone. The learner sets the stress feature of each to match their surface forms: s3 is set to +stress, s1 is already -stress. The revised lexicon is now

(19) The lexicon after comparing s3 and s1 in environment r1. Features: [stress,length]
 r1[-,-] r2[-,+]
 s1[-,-] s2[-,+]

6.3.5 Processing s4

Suffix s4, with underlying form [+x], has an unset length feature.

The learner compares s4 with s1. They differ in environments r1, r2 and r3, but always by both features, and s1 and s4 faithfully map their set stress feature in each differing environment. Nothing is set.

The learner compares s4 with s2. They differ in environment r2, but by both features, and s1 and s4 faithfully map their set stress feature in the environment. Nothing is set.

The learner compares s4 with s3. They differ in environment r1, by length alone. The learner sets the length feature of each to match their surface forms: s4 is set to +length, s3 is already -length. The revised lexicon is now

(20) The lexicon returned by Contrast Analysis. Features: [stress,length]
r1[-,-] r2[-,+]
s1[-,-] s2[-,+]

The entire correct lexicon has now been determined.

6.3.6 Comments on Illustration 2

Contrast Analysis succeeds in determining the entire lexicon despite the fact that more than half of the underlying features alternate, and were initially unset.

In this case, the algorithm likely benefited from two facts about the simulation. First, the fact that the phonological mapping used to generate the data maximally exploits the contrastive possibilities of the base of underlying forms (given the restriction to monosyllables): every difference in underlying specification results in a distinct surface alternation pattern, for both roots and suffixes. Second, the fact that the dataset provided to the learner included the outputs for all combinatorial possibilities (there were no gaps for possible but unrealized forms).

This illustration reveals some additional key properties of the algorithm. When the learner compares r2 with r4, it observes that in the environment s4 they differ in both stress and length. Thus, r2 and r4 are not minimal pairs in the featural sense; they differ in more than one feature. However, the learner then observes that both morphemes are already specified underlyingly as long, and length is not faithfully mapped for r2 (it surfaces as short). So the surface difference between them cannot be a consequence of an underlying length contrast (r4 was set long because it always surfaces as long, while r2 was set long to realize a contrast with r1). The learner concludes that the surface difference must be a consequence of an underlying contrast in stress, and sets the underlying values for stress to match the surface realizations of the morphemes for stress. This shows how Contrast Analysis goes beyond a simple search for featural minimal pairs; it can learn from pairs of morphemes that differ in more than feature if it can rule out all but one feature as a possible source of the contrast.

While the fact that r2 and r4 are both specified the same way for length makes it particularly obvious that surface differences between them cannot be the result of a

length contrast, the true basis for the judgment is that the length feature is not faithfully mapped for r2. FCF ensures the existence of at least one surface-differing feature that is faithfully mapped for both morphemes. Thus, a feature can be ruled out as responsible for the surface difference between two morphemes if it is not faithfully mapped for one of the morphemes, even if the underlying value for that feature has not yet been set for the other morpheme. This is illustrated when the learner is processing suffix s2, and compares s2 with s4. s2 and s4 differ on the surface in both stress and length in the environment r2. s4 is not yet specified for either stress or length. s2 is specified long, but surfaces as short in the environment. One cannot rule out length as a source of the surface difference solely by observing that s2 is underlyingly long while s4 has unknown underlying length; what if s4 is underlyingly short? But the learner can rule out length as a source of the surface difference based on the fact that s2 is underlyingly long but surfaces as short. This allows the learner to conclude that stress must be the feature that is faithfully mapped for each, and the learner sets the underlying stress features (neither of which had yet been set) to match their surface realizations.

7 Discussion

7.1 *Dependence on Minimally Contrastive Morphemes*

Contrast Analysis rests upon the Faithful Contrastive Feature Property. FCF guarantees that there will be a faithfully realized contrastive feature, but it does not guarantee that there will be only one. Indeed, forms may well have a number of underlying contrasts between them, all of which are essential to determining the correct surface forms for the morphemes. Contrast Analysis is only able to capitalize on FCF when it can limit the source of a contrast between the surface realizations of two morphemes in a given environment to a single feature. If more than one surface-differing feature is potentially faithfully mapped, the algorithm declines to set any underlying forms. This conservative approach is motivated by the possibility of feature interaction; it is possible that one of the surface-differing features is determined not by faithfulness to underlying specification, but by markedness-driven interaction with another of the surface-differing features.

In the illustrations above, the lexical base is limited to a very small space, and that entire space is realized in the data: every possible form is included in the data. That means that, for each morpheme, every minimally contrasting possible morpheme is included in the actual data. This will not in general be true of realistic linguistic data encountered by learners; many possible morphemic forms simply won't be used as actual morphemes by the language. The effectiveness of Contrast Analysis could be limited by the availability of minimally contrastive pairs of morphemes. Also, phonotactic restrictions can serve to block full complements of minimal pairs even with a full base, as is shown by the stress feature of suffix s3 in Illustration 1. In general, one might expect such surface gaps in places where the grammar creates neutralizations.

It is cause for neither surprise nor concern that initial lexicon construction, as described above, will in general fail to completely determine the correct lexicon all on its own. If it could fully determine the lexicon in all cases, independent of any consideration of

constraint ranking, then learning would be far less mysterious than it currently is. A learner could construct the lexicon and then use error-driven constraint demotion to determine the constraint ranking. There is good reason to believe, however, that the lexicon and the ranking are sufficiently interrelated that they must be learned together. In more complex cases, the best one can hope for from initial lexicon construction is the determination of meaningful portions of the lexicon, enough to significantly assist the subsequent navigation of the joint lexicon/ranking hypothesis space.

However, Contrast Analysis could contribute a great deal even if there are significant numbers of morphemes lacking minimally contrastive counterparts. In initial lexicon construction, if there is even one densely packed lexical neighborhood, the learner should be able to use Contrast Analysis to determine the underlying forms of a number of morphemes in that neighborhood. That will greatly constrain and inform the subsequent learning of the constraint ranking. If the processing of the forms in the dense lexical neighborhood, with largely determined underlying forms, determines most or all of the ranking, then the ranking may in turn be effectively used to determine the underlying forms of other morphemes. During the joint learning of the rest of the lexicon and the constraint ranking, Contrast Analysis might be utilized by the learner to direct the consideration of lexical hypotheses. If the learner contemplates changing its hypothesized underlying form for morpheme, it might focus first on changing the underlying feature values for features that are capable of accounting for surface contrasts that exist between that morpheme and other morphemes. Under this approach, the learner would attempt to identify the features most likely to be responsible for determining contrasts between one morpheme and others, even when it cannot reduce the list to a single feature occurrence, and then focus its efforts on testing different values of those features.

It would be natural to attempt to extend the power of Contrast Analysis by localizing the types or domains of contrast. When linguists compare two output forms on, say, the voicing of the initial consonant, they often do so with an implicit theory stating that initial consonant voicing never interacts with the height of a vowel occurring three syllables later. Thus, even if two output forms are different in that later vowel, and it is accounted for by faithfully mapped vowel height features, the linguist might fairly infer that the voicing features of the initial consonant must be faithfully mapped also; the linguist would presume that the height contrast must be independent of the voicing contrast. If such an extension could be accomplished, it could greatly enhance the power of Contrast Analysis, by allowing it to separate out independent contrasts between two forms. However, it is not entirely apparent how to achieve this in a plausible way. The learner would somehow need to know what feature values could interact under what circumstances. Furthermore, this could not be information gleaned in full knowledge of the language being learned, because Contrast Analysis is part of the learning process, so the learner does not yet know for certain what the correct grammar is. The information on non-interacting features would need to be based either on knowledge of what features could not possibly interact in any possible language (presumably without the learner itself engaging in a computationally expensive algorithmic analysis of Universal Grammar), or else on hypotheses about the (language-specific) interactivity of features constructed by the learner during a prior stage of learning. The latter option could be effective in

boosting the power of Contrast Analysis if it could be accomplished via, for example, a computationally efficient analysis of the constraint ranking returned by phonotactic learning.

7.2 The Role of Output-Output Comparison

Contrast Analysis commits to the idea that learners actively compare the output realizations of different morphemes. The claim that learners can compare outputs does not attribute any major new computational capacity to the learner. It is virtually inevitable that learners compare the output forms of different realizations of the same morpheme as they attempt to fully analyze and account for alternations, as testified by extensive prior work utilizing correspondences between output forms, both in morphological/phonological learning (see Albright and Hayes (2002) for an example) and in the literature on similarity (for example, Frisch, Broe, and Pierrehumbert (1997)). Further, the learner cannot avoid engaging in an extensive amount of output-output comparison between larger utterances when engaging in morpheme discovery in the first place. The idealized learning situation used here assumes that the learner suddenly “knows” the identity of the language’s morphemes, and what segments are affiliated with what morphemes in different words, before learning the underlying forms. In fact, it is plausible that a healthy amount of underlying form and ranking learning occurs in tandem with morpheme discovery. The commitment to the existence of a given morpheme most likely follows the hypothesizing and testing of output correspondences among words believed to contain the morpheme.

As described in the section above on exploiting FCF in learning, initial lexicon construction uses both kinds of output-output correspondence. The determination of which features alternate and which do not requires output-output correspondence between realizations of the same morpheme in different environments. Contrast Analysis requires output-output correspondences between realizations of different morphemes in the same environment. As currently characterized, the Surgery learning algorithm does not use output-output correspondence of either type. Surgery relies on error-driven learning (the need to get the correct output for each word) combined with the assumption that each morpheme has only a single underlying form, and manipulates both the constraint ranking and the underlying forms in an effort to produce the correct outputs. It remains an open question as to whether output-output correspondences, and procedures like Contrast Analysis, could be profitably exploited in the process of surgery-type learning.

7.3 Beyond FCF

Contrast Analysis could conceivably be profitably used even with linguistic systems that do not support FCF, at least in the simple form described here. One way to do this would be to identify special circumstances, for a linguistic system, where FCF is true, even if it isn’t true in all circumstances, and then apply Contrast Analysis only to forms occurring in those circumstances.

Another way Contrast Analysis might be used in systems that do not support FCF would be to apply Contrast Analysis initially as if the FCF were true, and then detect later when FCF was incorrectly assumed and change the underlying forms accordingly. Presumably

this approach would need to be justified by a demonstration that the consequences of incorrectly assuming FCF are detectable, and that a learner has a clear course of recovery.

A potentially interesting observation about the proof of FCF is that it relies entirely on comparisons between candidates with outputs matching the words that contrast. When comparing *paká* with *paká:*, the reasoning about the underlying form for the suffix -ka relies entirely on the comparison between two candidates, the two with the outputs matching those two forms, without any need to consider the many other possible candidates. Being able to restrict attention to such comparisons, which are easily constructed from the surface data, could help make it computationally efficient for the learner to employ contrast-based observations in the process of inferring relationships between the ranking and the underlying forms.

8 Appendix: Supra-Binary Features

One of the conditions given to ensure that a linguistic system has FCF is that all of the segmental features be binary. This condition turns out to be important: there exist linguistic systems which satisfy all of the other conditions, but do not have FCF. We give a simple example in this appendix.

Assume we have a system in which consonants have a three-valued place feature, taking values coronal, labial, and velar. Assume also that the system includes a root, *r*, with underlying form /pa/, and two suffixes, *s1* with underlying form /-ka/ and *s2* with underlying form /-pa/. This gives rise to two words, *r+s1* and *r+s2*. Obviously, our focus will be on contrast between *s1* and *s2* in the environment *r*.

Assume this system has the following constraints:

IDENT(place): correspondents should have identical values for place.

OCP(place): consecutive consonants should have non-identical place.

NOVELARSUFF: suffix consonants should not be velar.

NOLABIALSUFF: suffix consonants should not be labial.

The FCF-defying effect can be achieved with the ranking

OCP(place) \gg IDENT(place) \gg NOVELARSUFF \gg NOLABIALSUFF

Under this ranking, we get the following mappings.

r+s1: /pa+ka/ \rightarrow *paka*

r+s2: /pa+pa/ \rightarrow *pata*

Suffixes *s1* and *s2* are realized differently on the surface in environment *r*. They differ in only one feature, place. Yet, while *s1*'s place feature is mapped faithfully, *s2*'s place feature is not. The constraint OCP(place) will not permit both *r* and *s2* to surface faithfully as labial, IDENT(place) will only permit one of the morphemes to be unfaithful in order to satisfy OCP, and the suffix-specific markedness constraints both determine that the suffix will be unfaithful (rather than the root), and that the unmarked place value coronal will be the surface place value for *s2*. Note that the underlying place specification for *s2* is not arbitrary: it could easily be supported by combining the suffix with a non-

labial root like /ga/: /ga+pa/ → *gapa*, not **gata*. Absent the intervention of OCP(place), IDENT(place) preserves place in both roots and suffixes.

This example mirrors the possibility explicitly discussed in the proof of the FCF result. Given comparison c1, between /pa+ka/ ↔ *paka* and /pa+ka/ ↔ *pata*, and comparison c2, between /pa+pa/ ↔ *pata* and /pa+pa/ ↔ *paka*, the highest-ranked constraint having a preference in either comparison is a faithfulness constraint, IDENT(place), which makes the decision for comparison c1. In comparison c2, the candidates tie on IDENT(place), and the choice between them falls to a lower-ranked constraint, NOVELARSUFF.

Table 5 The ranking arguments resulting from the two comparisons.

		OCP(place)	IDENT(place)	NOVELARSUFF	NOLABIALSUFF
/pa+ka/	W: <i>paka</i>				
	L: <i>pata</i>	e	W	L	e
/pa+pa/	W: <i>pata</i>				
	L: <i>paka</i>	e	e	W	e

This example helps clarify what the FCF result is really saying. It is really guaranteeing the existence of a feature on which the two morphemes differ on the surface, which is mapped faithfully for one of the morphemes, and which is specified differently underlyingly for the other morpheme. In other words, the contrast has to originate from an underlying difference in feature specification where that difference matters: the underlying difference results in a difference on the surface, and that can only happen by faithfully mapping at least one of the underlying values. When the features are forced to be binary, the ‘other’ morpheme ends up necessarily faithfully mapping the underlying value as a consequence: since both the underlying and surface values of the different feature for the ‘other’ morpheme have to be different from the underlying/surface value of the first morpheme, and there is only one other value (due to binarity), the underlying and surface values of the other morpheme must match. The above example shows that when a feature is not binary, it is possible for both the underlying and surface values of the ‘other’ morpheme, in this case *s2*, to be different from the faithfully mapped feature value of the first morpheme (*s1*), but not identical to each other.

One might interpret this outcome as another argument for binary features: if all features are binary, then at least one must be faithfully mapped in both contrasting forms (given that the other conditions for FCF are satisfied). A different reaction would be to attempt to construct a weaker FCF-type result for supra-binary features. As illustrated in the example above, for the crucial surface-differing feature, one morpheme has the feature faithfully mapped. The other morpheme must have a different value for the underlying feature. This does not tell the learner for certain what the underlying feature value is, but it does tell the learner what it isn’t: it isn’t the same value as the faithfully mapped value of the first morpheme. In the example, the learner doesn’t know if the underlying place value for the consonant in suffix *s2* is coronal or labial, based upon the comparison with *s1*, but the learner does know it is not velar. A learner could use this kind of information to rule out certain values of certain features based on contrast information. If a learner can rule out all but one of the possible values for a feature in this fashion, it can

confidently set the feature. Of course, this would require knowing, for the feature determining the contrast, which one of the morphemes is faithfully mapping its underlying value. That could be determined directly if one of the morphemes has its underlying value for the feature already set, but not if neither morpheme has its underlying value set.

9 References

- Albright, Adam, and Hayes, Bruce. 2002. Modeling English past tense intuitions with minimal generalization. In *Proceedings of the Sixth Meeting of the ACL Special Interest Group in Computational Phonology*, ed. by Michael Maxwell, 58-69. Association for Computational Linguistics.
- Alderete, John. 1999. Morphologically Governed Accent in Optimality Theory. PhD. dissertation, University of Massachusetts, Amherst.
- Benua, Laura. 1997. Transderivational Identity: Phonological Relations between Words. PhD. dissertation, University of Massachusetts, Amherst.
- Chomsky, Noam, and Halle, Morris. 1968. *The Sound Pattern of English*. New York City: Harper & Row.
- Clark, Eve. 1987. The principle of contrast: A constraint on language acquisition. In *Mechanisms of Language Acquisition*, ed. by Brian MacWhinney, 1-33. Hillsdale, NJ: Lawrence Erlbaum.
- Frisch, Stefan, Broe, Michael, and Pierrehumbert, Janet. 1997. Similarity and phonotactics in Arabic. Ms., Psychology Dept., Indiana University, and Linguistics Dept., Northwestern University. ROA-223.
- McCarthy, John, and Prince, Alan. 1995. Faithfulness and Reduplicative Identity. In *University of Massachusetts Occasional Papers 18: Papers in Optimality Theory*, ed. by Jill Beckman, Laura Walsh Dickey, and Suzanne Urbanczyk, 249-384. Amherst, MA: GLSA, University of Massachusetts.
- Prince, Alan, and Smolensky, Paul. 1993. Optimality Theory: Constraint interaction in generative grammar. Ms., Rutgers University, New Brunswick, and the University of Colorado, Boulder. ROA-537.
- Prince, Alan, and Tesar, Bruce. 2004. Learning phonotactic distributions. In *Constraints in Phonological Acquisition*, ed. by René Kager, Joe Pater, and Wim Zonneveld, 245-291. Cambridge: Cambridge University Press.
- Tesar, Bruce, Alderete, John, Horwood, Graham, Merchant, Nazarre, Nishitani, Koichi, and Prince, Alan. 2003. Surgery in language learning. In *Proceedings of the Twenty-Second West Coast Conference on Formal Linguistics*, ed. by G. Garding and M. Tsujimura, 477-490. Somerville, MA: Cascadilla Press. ROA-619.