Multi-Recursive Constraint Demotion*

Bruce B. Tesar
Rutgers University

5/10/97

1 Introduction

A significant source of difficulty in language learning is the presumed "incompleteness" of the overt information available to a language learner, termed here an 'overt form', when they hear an utterance. The complete structural description assigned to an utterance by linguistic analysis includes representational elements not directly apparent in the overt form, but which play a critical role in linguistic theory. Because the central principles of linguistic theory, including those determining the space of possible human grammars, make reference to these elements of 'hidden structure', recovering them is necessary if the overt data are to be brought to bear on the task of determining the correct grammar.

Hidden structure, although not directly perceivable, need not be a great difficulty if it can easily be reconstructed based upon the overt form. Hidden structure becomes a problem when the overt form is ambiguous. If a given overt form is consistent with two or more different full structural descriptions, then the correct structural description cannot be determined from the information in that overt form alone. Presumably, other data, from other overt forms, is necessary to determine the correct structural description.

In Optimality Theory (Prince and Smolensky, 1993), learning a grammar means finding a correct ranking for the universal constraints. The learner, given a collection overt forms (presumed to be the overt reflexes of grammatical utterances), must arrive at a ranking of the constraints such that, for each overt form, there is a matching structural description which is optimal for some input under that ranking. Tesar and Smolensky (Tesar and Smolensky, to appear) have demonstrated that, given the correct full structural descriptions, a constraint ranking can be determined efficiently which makes all of those structural descriptions optimal. Thus, if the problem of hidden structure can be overcome, constraint rankings can be learned.

Recent work by Tesar on language learning in Optimality Theory has used an iterative strategy to approach the problem of determining hidden structure (Tesar, to appear) (Tesar, 1997). The strategy processes overt forms in serial fashion, one at a time. One notable property of that work is that, when the processing of an overt form is complete, the procedure retains as information only a single hypothesized constraint hierarchy. Upon receipt of an overt form, the algorithm modifies its hypothesized constraint ranking as necessary to accommodate the overt form, but then retains only the resulting constraint hierarchy as information when moving on to the next overt form. This behavior is standard practice in what is known as the "language learnability in the limit" framework(Gold, 1967). One motivation for this type of limitation is to avoid learning procedures which remember an unbounded number of utterances. However, limiting the

*The author would like to thank Eric Bakovic and Alan Prince for useful discussions. Any errors are the sole responsibility of the author.

learning procedure to storing only a single grammar hypothesis is an extreme in the opposite direction. In the case of Optimality Theory, a constraint hierarchy does not preserve all of the information that was used to arrive at it. Further, it is possible for modifications to a constraint hierarchy, made in response to a new overt form, to partially obscure domination relations put in place in response to previous overt forms.

The primary proposal of this paper is a data structure, separate from the hypothesized constraint hierarchy, which records information obtained from observed data[1]. Specifically, the learner constructs a list of mark-data pairs; the form and role of mark-data pairs is explained in section 2.2. A constraint hierarchy is easily obtained from this list, but by keeping the list itself information is retained that would otherwise be lost if only the generated constraint hierarchy were retained. The retained information makes it easier for the learner to take account of multiple overt forms simultaneously. Further, the necessary size of the mark-data pair list is quite reasonable; the learner is not required to store an unreasonable amount of data.

The value of the information contained in mark-data pair lists is demonstrated in section 3, where it is shown how a learner can use the information to detect when a set of interpretations of overt forms is mutually inconsistent. A learning strategy using this ability to detect inconsistencies is then presented in section 3.3. The learner, when presented with an overt form, creates a separate grammar hypothesis for each possible interpretation of the overt form, and uses constraint demotion both to infer constraint rankings from the interpretations and to detect and eliminate non-viable interpretations. This approach is not sensitive to the number of possible grammars so much as to the degree of ambiguity in the overt forms. This strategy is guaranteed to find a correct grammar, and an empirical argument will be made, within the domain of metrical stress, that it is far more efficient than brute-force enumeration of all possible grammars.

2 Learning in Optimality Theory

2.1 Metrical Stress

Metrical stress theory has been a domain of focus for several learning investigations. Dresher and Kaye (Dresher and Kaye, 1990) applied cue learning to a system of stress grammars set within the principles and parameters framework (Chomsky, 1981). Approaches less closely tied to any explicit linguistic framework have also been investigated (Gupta and Touretzky, 1994) (Daelemans, Gillis, and Durieux, 1994). Metrical stress is an appealing domain because a lot is known about it, and because it can be treated somewhat in isolation from other aspects of phonology.

Metrical stress was selected for the current investigation because it permits the issue of input/output faithfulness to be set aside. In the present analysis, underlying forms are strings of syllables, and structural descriptions assign stresses to the syllables; no insertion/deletion of syllables is considered (for discussion of learning underlying forms, including relations with child language acquisition work, see (Tesar and Smolensky, to appear), (Smolensky, 1996), (Hale and Reiss, 1996), and the works cited therein). Thus, the underlying form for an utterance can be directly (and correctly) inferred from the overt form; the underlying form is simply the syllables of the overt form (without the stresses).

For purposes of illustration, consider the following simple optimality theoretic system for metrical stress. Each structural description is of a single prosodic word, and all overt forms are of single prosodic words. The overt forms are strings of stress levels, one for each syllable. The overt forms range from 2 to 7 syllables in length. A structural description is a grouping of the syllables (with their stress levels) into feet. Table

[1] It should be emphasized that what is new here is not the idea of a learner storing information apart from a single grammar; this work is preceded by countless others in that regard. What is new is the particular information structure stored, lists of mark-data pairs, along with its application to the problem of hidden structure in learning.

Overt Forms	Descriptions
[1 0]	[(1 0)]
[0 1 0]	[0 (1 0)]
[2 0 1 0]	[(2 0) (1 0)]
[0 2 0 1 0]	[0 (2 0) (1 0)]
[2 0 2 0 1 0]	[(2 0) (2 0) (1 0)]
[0 2 0 2 0 1 0]	[0 (2 0) (2 0) (1 0)]

Table 1: The Warao Stress Pattern: 1 = main stress, 2 = secondary stress, 0 = unstressed, [ ] denote prosodic word boundaries, ( ) denote foot boundaries.

Name	Description
PARSE	a syllable must be footed
MAIN-RIGHT	align the head-foot with the word, on the right edge
MAIN-LEFT	align the head-foot with the word, on the left edge
ALL-FEET-RIGHT	align each foot with the word, on the right edge
ALL-FEET-LEFT	align each foot with the word, on the left edge
IAMBIC	align the head syllable with its foot, on the right edge
TROCHAIC	align the head syllable with its foot, on the left edge

Table 2: The constraints for the simple metrical system.

1 shows the pairings of overt forms and structural descriptions for the stress pattern of Warao (Osborn, 1966) (Hayes, 1980) (Hayes, 1995); the analysis is taken from (McCarthy and Prince, 1993). GEN will only generate descriptions in which each foot has precisely one head syllable, which is the sole stress-bearing syllable of that foot. An unfooted syllable must be unstressed. GEN also requires that a prosodic word have precisely one head foot, whose head syllable bears main stress. If the word has any other (non-head) feet, their head syllables each bear secondary stress. Feet are strictly bisyllabic. The seven constraints, freely rankable, are listed in table 2. Table 3 shows a constraint hierarchy which generates the stress pattern for Warao shown in table 1.

Under the formal definition of Optimality Theory, a grammar requires a *total ranking* the constraints, with the relative ranking determined for every pair of constraints. However, the learning algorithm makes use of a more general space of hypotheses, that of stratified hierarchies (see (Tesar and Smolensky, to appear) for discussion of the role of stratified hierarchies in learning). In a stratified hierarchy, one or more constraints may occupy the same stratum in a hierarchy. The constraints of a stratum are not ranked relative to each other, but all of them dominate all constraints occupying lower strata. The use of stratified hierarchies requires that the definition of an OT mapping be extended. Two candidates are compared on a stratified hierarchy as follows. The two candidates are evaluated on all constraints in the top stratum, and the marks

Stratum 1		Stratum 2		Stratum 3
PARSE MAIN-RIGHT TROCHAIC	$\gg$	ALL-FEET-RIGHT MAIN-LEFT IAMBIC	$\gg$	ALL-FEET-LEFT

Table 3: A Constraint Hierarchy Generating the Warao Stress Pattern

			PARSE	M-R	TROCH	A-F-R	M-L	IAMB	A-F-L
☞	*a*	[0 (2 0) (1 0)]	*			* *	* * *	* *	* * * *
	b	[(1 0) (2 0) 0]	*	* * *		* * * *		* *	* *
	c	[0 0 0 (1 0)]	* * *				* * *	*	* * *
	d	[0 (0 2) (0 1)]	*		* *	* *	* * *		* * * *
	e	[(0 2) (0 1) 0]	*	*	* *	* * * *	* *		* *
	f	[(2 0) 0 (1 0)]	*			* * *	* * *	* *	* * *

Table 4: The grammatical description of 5 syllables in Warao, along with some competitors.

are pooled together. The candidate which has the smaller total number of violations of the constraints in the stratum is the more harmonic (better) one. A constraint violated equally by both candidates makes no contribution to distinguishing the two (as always). If the two candidates fare equally on the top stratum, the decision is passed to the next stratum, and so forth.

2.2 Mark-Data Pairs

Optimality Theory is inherently comparative. A structural description is grammatical not because of how well it satisfies constraints in isolation, but because it better satisfies the ranked constraints of a grammar than every other candidate structural description for the same underlying form. The assertion that [0 (2 0) (1 0)], candidate *a* in table 4, is grammatical in Warao contains information about the constraint ranking for Warao, to the effect that the ranking must make candidate *a* simultaneously more harmonic than all other candidate structural descriptions for a word of five syllables, such as *b* through *f* (also listed in table 4). Such competing descriptions have been termed *implicit negative evidence* (Tesar and Smolensky, to appear): positive evidence for the grammaticality of a description implies the ungrammaticality of its competitors.

The information provided by implicit negative evidence can be made explicit by pairing the constraint violations of a grammatical description, called the *winner*, with the violations of a competitor, called the *loser*. Such a pair of lists of constraint violation marks is termed a *mark-data pair*. It is useful to cancel constraint violation marks in common, so that the only remaining marks indicate which constraints are violated more by the *winner*, and which constraints are violated more by the *loser*. An example can be formed by using candidate *a* of table 4 as the winner, and candidate *b* as the loser. The mark-data pair formed is shown in (1).

Loser Marks	Winner Marks
ALL-FEET-RIGHT MAIN-RIGHT	ALL-FEET-LEFT MAIN-LEFT

(1)

In each column, there is one mark for each constraint violated more by the relevant candidate, in comparison to the other one. Notice that the notation does not indicate the extent of greater violation: the loser, *b*, violates ALL-FEET-RIGHT two more times than the winner, *a*, but only one mark for ALL-FEET-RIGHT appears in the Loser Marks column. This is because the magnitude of greater violation is not relevant information; what a mark-data pair reflects is only which candidate, if either, incurs more violations of a constraint, not how many more. Notice also that constraints PARSE and IAMBIC are violated by both *a* and *b*. No marks occur for these constraints in the mark-data pair, because both candidates violate each constraint an equal number of times (each violates PARSE once, and IAMBIC twice). A constraint is only relevant in distinguishing two candidates if one of the candidates violates the constraint more than the other.

What a mark-data pair implies is that at least one of the constraints violated more by the loser must dominate all of the constraints violated more by the winner. The mark-data pair in (1) contains the information that (ALL-FEET-RIGHT **or** MAIN-RIGHT) must dominate (ALL-FEET-LEFT **and** MAIN-LEFT). Of all the constraints on which the loser and the winner have a differing degree of violation, the highest-ranked one must be violated more by the loser (otherwise it will not lose to the winner).

The information about the grammar provided by a grammatical description can thus be encapsulated in a set of mark-data pairs, where each pair matches the grammatical description with a suitably informative competitor. More generally, the information needed to determine the constraint ranking for a language can be compiled by combining sets of mark-data pairs for several grammatical descriptions into one overall set of mark-data pairs for the language.

2.3 Recursive Constraint Demotion

Recursive Constraint Demotion (RCD) is a procedure for learning a constraint ranking from a set of mark-data pairs (Tesar and Smolensky, 1995). It is based upon the following observation. For any specific grammar, at least one constraint must be undominated, that is, not dominated by any other constraint. That constraint cannot be violated more times by any winner in comparison to any of its losers. Thus, the constraints which can possibly be at the top of the hierarchy can be determined by examining the constraint violation marks of the winners in the mark-data pairs. If a constraint does not appear among the (post-cancellation) marks for the winner of any of the mark-data pairs, it can be ranked at the top. Any constraint appearing among the winner marks of a mark-data pair must be dominated by at least one of the constraints with loser marks in that pair, thus the constraint appearing in the winner marks cannot be top-ranked. RCD is in essence a repeated application of this observation.

RCD will be illustrated with the list of mark-data pairs in (2).

Loser Marks	Winner Marks
ALL-FEET-RIGHT MAIN-RIGHT	ALL-FEET-LEFT MAIN-LEFT
TROCHAIC	IAMBIC
PARSE	ALL-FEET-RIGHT IAMBIC
ALL-FEET-RIGHT	ALL-FEET-LEFT

(2)

The learner begins with all of the constraints unranked, and proceeds as follows. First, all of the constraints with no marks for any of the winners are determined, and placed in the top stratum of the hierarchy. In (2), the constraints PARSE, MAIN-RIGHT, and TROCHAIC do not appear anywhere in the Winner Marks column. None of the winners in the list incurs greater violation than its respective loser of any of these three constraints. Thus, these three constraints may be put in the first (top) stratum of the hierarchy, as shown in (3).

Stratum 1

PARSE
MAIN-RIGHT
TROCHAIC

(3)

Next, the loser's marks for each mark-data pair are examined. If a loser has a mark for a constraint just placed in the top stratum, then that constraint's position ensures that the loser will be less harmonic than its associated winner. Thus, the mark-data pair's conditions are all satisfied, so the mark-data pair may be removed from the list. In (2), the first, second, and third pairs have, in the Loser Marks, an occurrence of at least one of the constraints in the top stratum. Because the constraints in the first stratum are guaranteed

to dominate the rest of the constraints, the winners in the first, second, and third pairs are guaranteed to be more harmonic than their corresponding losers, because those losers each violate a top-ranked constraint more than the winners. Once all such mark-data pairs have been removed, the first pass of RCD is complete. There is now only one mark-data pair remaining in the list, as shown in (4).

Loser Marks	Winner Marks
ALL-FEET-RIGHT	ALL-FEET-LEFT

(4)

The next pass performs the same procedure, but using only the remaining mark-data pairs, and the constraints not already placed into the hierarchy. Due to the removal of mark-data pairs in the previous pass, there will be at least one among the remaining constraints that does not have a violation mark for any of the winners in the remaining mark-data pairs. The list in (4) has no remaining marks in the Winner Marks column for ALL-FEET-RIGHT, MAIN-LEFT, and IAMBIC. Those constraints may be placed into the second stratum of the hierarchy, indicating that they are all dominated by the constraints in the top stratum (output on the previous pass), as shown in (5).

Stratum 1		Stratum 2
PARSE MAIN-RIGHT TROCHAIC	$\gg$	ALL-FEET-RIGHT MAIN-LEFT IAMBIC

(5)

The constraints placed into the second stratum may then be used to remove more mark-data pairs from the list (those pairs whose loser contains a mark for at least one of the constraints just placed into the hierarchy). This procedure is repeated until all of the constraints have been placed into the hierarchy. In the illustration, the placement of ALL-FEET-RIGHT into Stratum 2 causes the subsequent removal of the sole mark-data pair in (4), leaving no remaining mark-data pairs. Thus, on the next pass the remaining constraint, ALL-FEET-LEFT, may be placed into Stratum 3 At this point, all of the constraints have been ranked, so RCD is complete, returning constraint hierarchy (6).

Stratum 1		Stratum 2		Stratum 3
PARSE MAIN-RIGHT TROCHAIC	$\gg$	ALL-FEET-RIGHT MAIN-LEFT IAMBIC	$\gg$	ALL-FEET-LEFT

(6)

One important property of RCD is that it automatically detects when the list of mark-data pairs is inconsistent, that is, when no ranking exists which simultaneously makes each winner more harmonic than its corresponding loser. When given such a list, the algorithm will at some point reach the end of a pass and have no constraints without marks for the remaining winners, even though there remain constraints not yet ranked. This means that some cycle of dominations is implied by the data, contradicting strict domination. This property is further discussed and illustrated in section 3.

2.4 Identifying Informative Mark-Data Pairs

RCD is guaranteed to find a correct constraint hierarchy, when given a suitable list of mark-data pairs. However, this leaves the problem of getting the appropriate mark-data pairs from the actual overt information directly available to the learner. To apply RCD, the learner needs to (a) arrive at the correct structural description (the winner) for each observed overt form; and (b) select appropriate competing descriptions

(losers) for each winner to form informative mark-data pairs. The problem in part (a) will be discussed further in section 3.

A solution to part (b), Error-Driven Constraint Demotion (EDCD), uses parsing to identify informative losers (Tesar, 1996). EDCD requires the learner to at all times have a hypothesized ranking. Given a hypothesized ranking, along with a grammatical full structural description, the learner can compute what structural description is assigned to the underlying form (of the grammatical description) by the hypothesized ranking. The procedure for computing the optimal description of an underlying form for a particular constraint hierarchy is called *production-directed parsing* (Tesar and Smolensky, to appear). If the assigned description matches the given grammatical one, then the hypothesized ranking already makes the grammatical form optimal; given that hypothesized ranking, there is nothing further to be learned from the grammatical description. If the assigned description does not match, then a mark-data pair formed by the grammatical description (as the winner) and the description assigned by the hypothesized ranking (the loser) contains information not reflected in the hypothesized ranking. The ranking can then be changed to reflect the new mark-data pair. This procedure can be repeated until a ranking is arrived at which makes the winner the optimal description.

Because of the error-driven nature of this procedure, a mark-data pair is only formed if it provides useful information not contained in the learner's hypothesized ranking. The number of informative mark-data pairs needed to determine a particular language has been proven to be less than the square of the number of constraints in the system (Tesar and Smolensky, to appear). Thus, the number of learning steps (the number of mark-data pairs formed and used for learning) by the learner will never exceed that limit.

3 Multi-Recursive Constraint Demotion

This leaves the problem of finding the correct structural description for an observed overt form. An *interpretation* of an overt form is a structural description whose overt portion matches the overt form. The overt form [0 1 0], a trisyllabic word with stress on the middle syllable, has (at least) two interpretations: [(0 1) 0] and [0 (1 0)]. These interpretations are full structural descriptions (the foot form is fully specified), and their overt portions (the stress levels assigned to the syllables) match the overt form [0 1 0]. This distinguishes the interpretations of [0 1 0] from other candidate structural descriptions of a trisyllabic word, such as [(1 0) 0] or [0 (0 1)], whose overt portions do not match [0 1 0].

In the fully general situation, part of determining an interpretation of an overt form is determining the underlying form(s). The present paper will avoid the challenging problems of identifying and learning underlying forms, and assume that, for a given overt form, the underlying form is apparent. What are not necessarily apparent are other aspects of the interpretation encompassing an overt form and its underlying form (in the case of metrical stress, the foot structure).

Given the underlying form for an overt form, the set of possible interpretations for the overt form is quite well-defined, due to the structure of Optimality Theory. The set of candidate structural descriptions for the underlying form is defined by GEN. The descriptions among them whose overt portions match the overt form are precisely the candidate interpretations of the overt form. Thus, knowledge of GEN, combined with the overt form and its underlying form, give the learner access to the set of possible interpretations of that overt form.

The challenge for the learner, then, is to select the correct interpretation, the one assigned by the language being learned, for the observed overt form. Perhaps the most obvious strategy is a brute-force approach: given a set of overt forms, generate every possible interpretation of each overt form, and then consider all possible combinations of interpretations, where each combination includes precisely one interpretation of

Interp-X1	Interp-Y1	Interp-Z1
Interp-X1	Interp-Y1	Interp-Z2
Interp-X1	Interp-Y2	Interp-Z1
Interp-X1	Interp-Y2	Interp-Z2
Interp-X2	Interp-Y1	Interp-Z1
Interp-X2	Interp-Y1	Interp-Z2
Interp-X2	Interp-Y2	Interp-Z1
Interp-X2	Interp-Y2	Interp-Z2

Table 5: The eight possible combinations of interpretations for three overt forms with two interpretations each.

each overt form. While this might be the most obvious way to consider all interpretations of overt forms, it is not obviously tractable. The number of possible combinations will be the product of the number of interpretations for each overt form. For example, suppose a language has three overt forms, Overt-X, Overt-Y, and Overt-Z. Each of these overt forms has two interpretations: the interpretations of Overt-X are Interp-X1 and Interp-X2, the interpretations of Overt-Y are Interp-Y1 and Interp-Y2, and so forth for Overt-Z. There are $2 \times 2 \times 2 = 8$ possible combinations of the interpretations, shown in table 5.

If many of the overt forms for a language have significant ambiguity, then that product could be quite large. If a language contains a large number of overt forms with any ambiguity at all, then the number of combinations of interpretations will grow exponentially with the number of overt forms. In fact, as will be illustrated in section 5, the combinatorial explosion of the number of combinations of interpretations can be much greater than that of the number of total rankings of the constraints.

Fortunately, there is a better way. It requires, however, a different formulation of constraint demotion than has been used previously. The new formulation, called Multi-Recursive Constraint Demotion, is presented in this section, along with a particular strategy for applying it to the problem of learning constraint rankings from overt forms.

3.1 Multi-Recursive Constraint Demotion

Recall EDCD, the procedure for selecting losers, described in section 2.4. Learning takes place whenever an error occurs, where an error is a mismatch between the correct interpretation (full structural description) of an overt form, and the structural description of the overt form that is optimal according to the learner's current constraint hierarchy. The learner then uses the currently optimal (but incorrect) structural description as the loser, and the correct interpretation as the winner, forming a mark-data pair. The learner then modifies their constraint hierarchy based upon that mark-data pair.

The new formulation concerns how the constraint hierarchy is modified in response to a mark-data pair. The new formulation, *Multi-Recursive Constraint Demotion* (MRCD), requires the learner to keep not only a hypothesized constraint hierarchy, but also a list of the mark-data pairs used to arrive at that hierarchy. Thus, when a new mark-data pair is selected for learning, the learner adds the new pair to their existing list of mark-data pairs. Then, instead of directly acting on their current constraint hierarchy using only the new mark-data pair, the learner applies RCD (as described in section 2.3) to the entire list of mark-data pairs (all of the previously used ones, along with the new one), deriving a new constraint hierarchy from that list. Error-driven learning can proceed as before, with each learning step adding another mark-data pair to the list. The formal results for error-driven learning that limit the number of demotions prior to

		A-F-L	A-F-R	M-L	M-R	Troch	Iamb
a	[0 (1 0)]	*		*			*
b	[(0 1) 0]		*		*	*	
c	[(0 1) 0 0]		* *		* *	*	
d	[0 0 (1 0)]	* *		* *			*

Table 6: The interpretations [0 (1 0)] and [(0 1) 0 0] cannot simultaneously be optimal under any ranking.

convergence upon a correct hierarchy carry over to the new formulation; the number of instances of demotion is equivalent to the number of mark-data pairs added to the list. Once a correct ranking is reached, there will be no more errors on overt forms, so no further mark-data pairs will be added to the list.

By keeping the list of mark-data pairs, the learner keeps more information about what evidence they have seen so far, more information than can be contained in just the current hypothesized constraint hierarchy itself. The value of that additional information is explained next.

3.2 Detecting Inconsistencies

The value of keeping lists of mark-data pairs along with their constraint hierarchies follows from a particular property of RCD: it quickly and easily detects when the list of mark-data pairs is inconsistent, that is, when there is no possible ranking of the constraints making the winner of each pair more harmonic than its corresponding loser.

3.2.1 Sets of Interpretations Which Are Collectively Inconsistent

A set of interpretations can consist of interpretations each of which is possibly optimal, but that cannot all be simultaneously optimal. In the metrical stress system, the descriptions [0 (1 0)] and [(0 1) 0 0] each are possibly optimal, but they cannot both be optimal in the same grammar. Table 6 shows why. In order for [0 (1 0)] (candidate *a* in table 6) to be optimal, the top-ranked constraint must be one that it violates no more than does the shown competitor *b*. Thus, the possible top-ranked constraints consistent with *a* are All-Feet-Right, Main-Right, and Trochaic. However, the other would-be winner, [(0 1) 0 0] (candidate *c* in the table), has greater violation of each of these constraints that one of its competitors, *d*. Thus, any constraint ranking making *a* optimal causes *c* to lose to *d*. There is an insurmountable inconsistency in trying to make both *a* and *c* optimal in the same grammar.

To see how RCD can detect this, consider the list of mark-data pairs shown in (7), created by pairing *b* with *a* (the first row), and *d* with *c* (the second row), with *a* and *c* as the winners.

Loser Marks	Winner Marks
All-Feet-Right Main-Right Trochaic	All-Feet-Left Main-Left Iambic
All-Feet-Left Main-Left Iambic	All-Feet-Right Main-Right Trochaic

(7)

RCD is then applied to this list. None of the constraints can be placed into the hierarchy; all of the unranked constraints still appear in the Winner Marks column. This condition signals an inconsistency: for all the unranked constraints, each is required to be dominated by at least one other of the unranked constraints, an impossibility.

		A-F-L	A-F-R	M-L	M-R	Troch	Iamb
a	[0 (1 0) 0 0]	*	* *	*	* *		*
b	[(0 1) 0 0 0]		* * *		* * *	*	
c	[(1 0) 0 0 0]		* * *		* * *		*
d	[0 0 0 (1 0)]	* * *		* * *			*

Table 7: The interpretation [0 (1 0) 0 0] cannot be optimal under any ranking, because it will always lose to some competitor.

3.2.2 Interpretations Which Cannot Possibly Be Optimal

Another way a set of interpretations can be inconsistent is for the set to contain an interpretation that by itself cannot be optimal under any ranking of the constraints. Consider an overt form of five syllables with peninitial stress, [0 1 0 0 0]. One possible interpretation is [0 (1 0) 0 0]. However, this cannot be a grammatical interpretation, because there is no ranking of the constraints that makes it optimal. This description along with some key competitors are shown in Table 7. The impossibility of candidate *a* can be seen by comparing its violations to the other candidates shown. The only constraint on which *a* has minimal violation is Trochaic; on all other constraints, there is at least one candidate with fewer violations than *a*. Even if Trochaic is top-ranked, candidates *c* and *d* also have no violations of that constraint, equaling *a* for satisfaction. Thus, the job of distinguishing among those three candidates is passed to the next constraint in the ranking. If it is All-Feet-Left or Main-Left, then *c* beats *a*; if it is All-Feet-Right or Main-Right, then *d* beats *a*[2]. Changing the ranking changes the optimal candidate, but no ranking can make *a* optimal.

RCD will detect inconsistencies arising from the use, as a winner, of a structural description that cannot be optimal. Applying error-driven learning repeatedly to such a description will result in an inconsistent set of mark-data pairs. For example, given candidates *a* through *d* in table 7, the mark-data pairs formed by pairing *a*, as the winner, with *c* and *d* as losers, gives two mark-data pairs, shown in (8), which are already inconsistent. The inconsistency in the list will then be detected by RCD.

Loser Marks	Winner Marks
All-Feet-Right Main-Right	All-Feet-Left Main-Left
All-Feet-Left Main-Left	All-Feet-Right Main-Right

(8)

3.3 A Strategy for Applying MRCD

As explained at the beginning of this section, applying constraint demotion to each possible combination of interpretations for all of the overt forms of a language runs into combinatorial difficulty, because of the great number of such combinations. A better way, made possible by MRCD, is to consider the overt forms one at a time, processing one overt form fully before continuing to the next, and eliminating as many interpretations of an overt form as possible before proceeding to the next overt form.

Processing an overt form V means to consider, in turn, each possible interpretation I(V) of V as a winner. Considering a particular interpretation I(V) entails searching, via the application of MRCD, for a constraint hierarchy H which (a) holds the interpretation I(V) optimal; and (b) is consistent with the mark-data pairs previously obtained from other overt forms. If an interpretation I(V) is consistent with the existing set of

[2] The ranking of Iambic will not play a role in the decision, provided that it is dominated by Trochaic.

mark-data pairs, then additional mark-data pairs will be added to the list by MRCD until the list generates a constraint hierarchy holding I(V) optimal. If the interpretation I(V) is not consistent with the existing list of mark-data pairs, MRCD will detect the inconsistency, and the interpretation I(V) will be discarded and not further considered. The mark-data pair lists are kept for those interpretations I(V) of V which result in consistent mark-data pair lists (lists generating hierarchies holding the interpretation I(V) optimal); the other lists are discarded, along with their corresponding interpretations of V. Each mark-data pair list that is retained constitutes, along with its generated constraint hierarchy, a *grammar hypothesis* held by the learner.

The learner then turns to other overt forms, one at a time. For each grammar hypothesis (a mark-data pair list and its corresponding constraint hierarchy) held by the learner, whenever the currently optimal description of the underlying form doesn't match the overt form (indicating an error), the learner generates all possible interpretations of the overt form, and processes them. MRCD's ability to quickly detect inconsistencies allows the learner to eliminate inconsistent combinations of interpretations with prior grammar hypotheses. The learner can thus avoid having to evaluate many of the possible combinations of overt forms, while still guaranteeing that a correct constraint hierarchy will be obtained.

If a particular grammar hypothesis holds as optimal one of the interpretations of a new overt form (without addition of further mark-data pairs to the list), then the learner does not consider the other interpretations of the overt form in combination with that grammar hypothesis. This is the error-driven component of the learner. The assumption made by the learner is that one of the interpretations of the overt form is already optimal, that it is the correct interpretation. Notice that if this assumption is incorrect, no harm is done, because no additional mark-data pairs have been added: no error was detected, so no learning took place. That the currently optimal interpretation is incorrect cannot be inferred on the basis of this overt form; that fact must be the result of the requirements of other overt forms (most likely overt forms not yet seen by the learner). The significant advantage of this error-learning approach is that, once the learner has obtained the correct ranking, it does not continue to process all possible interpretations of each overt form it sees. The learner only engages in learning when it is forced to do so by an error; otherwise, the learner simply proceeds with normal processing on the assumption that its current grammar is correct.

Suppose a learner has a grammar hypothesis in the form of a list of mark-data pairs, and observes a new overt form which supports several interpretations. Suppose further that none of the interpretations is optimal under the learner's current grammar hypothesis, causing an error to be detected. In considering all possible interpretations, what the learner does is create a separate hypothesis for each interpretation by combining that interpretation with the learner's existing grammar hypothesis. Each new hypothesis may be pursued by searching for additional mark-data pairs which make the hypothesis' new interpretation optimal. If the pursuit of a particular interpretation results in an inconsistency (detected by RCD), then that interpretation and its attempted new hypothesis can be discarded; the interpretation being pursued was not reconcilable with the existing list of mark-data pairs.

This procedure can leave the learner with more than one tenable grammar hypothesis after the processing of an overt form. It may be that the learner does not yet have enough information to rule out all but one interpretation of that overt form. In this case, the learner keeps all of the tenable hypotheses. On the next overt form, the learner separately checks for an error between that overt form and each existing grammar hypothesis. Processing occurs for each grammar hypothesis detecting an error with the overt form.

Suppose that, after processing some number of overt forms, the learner has two viable hypotheses, represented by mark-data pair lists L_1 and L_2, and that the next overt form, V, has three interpretations, I_1, I_2, and I_3. Suppose further that L_1 holds as optimal, for the underlying form of V, the interpretation I_2; it doesn't actually matter for now which interpretation is optimal, so long as one of them is, thus avoiding an error. In that event, L_1 will be retained without further processing on V. Suppose that L_2 holds as optimal, for the underlying form of V, a description with an overt portion not matching V. That is an error,

and the learner will independently pursue (via the application of MRCD) three new possibilities: $L_2 + I_1$, $L_2 + I_2$, and $L_2 + I_3$. Of these three, those (if any) which lead to consistent grammar hypotheses will also be retained, along with L_1.

What keeps the number of hypotheses from exploding as more overt forms are observed is that the vast majority of combinations of existing grammar hypotheses with interpretations will be inconsistent, and thus are eliminated before the learner considers further overt forms.

3.4 The Learning Algorithm

This procedure starts with a single, empty list, and processes overt forms one at a time. Once enough overt forms have been observed, the learner will possess a list of mark-data pairs which gives rise to a constraint ranking generating the target language.

3.4.1 The Main Procedure

```
Initially, the learner's set of hypotheses {L} consists of a single empty list.
For each overt form V, with its underlying form U
    For each hypothesis L ∈ {L}
        remove L from the learner's list
        apply RCD to L to get the corresponding hierarchy H
        compute the optimal description D for U, using H
        If the overt portion of D does not match V
            find the set of interpretations {I} for V
            For each interpretation I ∈ {I}
                apply MRCD to I and L, getting new list N-L
                If N-L is an inconsistency code
                    discard N-L
                Else
                    add N-L to {L}
                End-If
            End-For
        Else
            place L back into {L}
        End-If
    End-For
End-For
```

	Parse	M-R	A-F-R	Troch	M-L	Iamb	A-F-L
[(1 0)]						*	
[0 (1 0)]	*				*	*	*
[(2 0) (1 0)]			* *		* *	* *	* *
[0 (2 0) (1 0)]	*		* *		* * *	* *	4(*)
[(2 0) (2 0) (1 0)]			6(*)		4(*)	* * *	6(*)
[0 (2 0) (2 0) (1 0)]	*		6(*)		5(*)	* * *	9(*)

Table 8: The Grammatical Descriptions for Warao. Large numbers of violations are abbreviated with a numeral; 6(*) indicates six violations.

3.4.2 The MRCD Procedure

Given a mark-data pair list L, an interpretation I:
apply RCD to list L to get constraint hierarchy H
compute the optimal description D assigned by H to U, the underlying form of I
While (D $\neq$ I) and (L not inconsistent)
cancel the common constraint violation marks to D and I
create a new mark-data pair with Marks(D) as the loser and Marks(I) the winner
add the new mark-data pair to L
apply RCD to L, getting either a new hierarchy H or an inconsistency code
compute the optimal description D assigned by H to U
End-While
If L is inconsistent (an inconsistency code was given by RCD)
return an inconsistency code
Else
return the new L
End-If

4 An Illustration: Stress in Warao

Consider the language Warao (Osborn, 1966) (Hayes, 1980) (Hayes, 1995). It has a trochaic, iterative right-to-left stress system, with penultimate main stress. The optimal structural descriptions for words of between two and seven syllables are given in table 8, along with the constraint violations for the optimality theoretic analysis presumed here (McCarthy and Prince, 1993).

In this section, an example run of the learning algorithm on overt forms from Warao is played out. The learner will learn a correct constraint hierarchy on the basis of three of the overt forms of the language: [0 1 0], [2 0 1 0], and [0 2 0 1 0] (in that order).

4.1 The First Overt Form

Suppose the first overt form presented to the learner is [0 1 0]. The correct interpretation of this form in Warao is [0 (1 0)], a right-aligned trochaic foot. However, another interpretation is also available as a candidate: [(0 1) 0]. The learner, starting with no initial information about the constraint ranking for Warao,

	Parse	M-R	A-F-R	Trochaic	M-L	Iambic	A-F-L
[0 (1 0)]	*				*	*	*
[(0 1) 0]	*	*	*	*			
[(1 0) 0]	*	*	*			*	
[0 (0 1)]	*			*	*		*

Table 9: Important Candidates for the three-syllable input.

will pursue each interpretation in turn. The constraint violations for the relevant candidates are given in table 9.

4.1.1 First Interpretation

First, consider the interpretation [0 (1 0)] (the order of consideration is arbitrary, and cannot affect the ultimate outcome). This description violates All-Feet-Left, Main-Left, Parse, and Iambic. An alternative description of three syllables, and one of the descriptions returned by production-directed parsing, is [(1 0) 0], violating All-Feet-Right, Main-Right, Parse, and Iambic. This description is as harmonic as the current interpretation, given that no domination relations are yet established. Using the interpretation as the winner and the alternate description as the loser gives the mark-data pair (9).

Loser Marks	Winner Marks
All-Feet-Right Main-Right	All-Feet-Left Main-Left

(9)

Note that the common violation marks for Parse and Iambic are canceled, and so do not appear in the mark-data pair. RCD may now be applied to this mark-data pair, giving the ranking (10).

Stratum 1		Stratum 2
Parse All-Feet-Right Main-Right Trochaic Iambic	$\gg$	All-Feet-Left Main-Left

(10)

Re-applying production-directed parsing to the trisyllabic underlying form with this ranking produces [0 (0 1)] as an optimal description, along with [0 (1 0)]. The former, [0 (0 1)] does not match the current interpretation (it does not even match the overt form), and so, as a loser, will form another informative mark-data pair with the same winner, shown as (11).

Loser Marks	Winner Marks
Trochaic	Iambic

(11)

Notice that even though the same structural description is being used as the winner, the constraints in the winner marks for (11) are different than for (9). This is because the mark cancellation is with respect to different losers in the two pairs. Pair (11) is added to (9) in the list. Applying RCD to the list (of two pairs)

produces constraint hierarchy (12).

Stratum 1		Stratum 2
PARSE ALL-FEET-RIGHT MAIN-RIGHT TROCHAIC	≫	ALL-FEET-LEFT MAIN-LEFT IAMBIC

(12)

This ranking holds the interpretation, [0 (1 0)], as the sole optimal description, as the application of production-directed parsing to the trisyllabic form reveals to the learner.

4.1.2 Second Interpretation

The learner now turns to the other interpretation of the overt form, [(0 1) 0]. This is pursued independently of the first interpretation, so the learner starts fresh, initially assuming no mark-data pairs. One of the descriptions optimal under the unrefined hierarchy is [(1 0) 0]. This differs from the interpretation only in foot form, so mark-data pair (13) is generated.

Loser Marks	Winner Marks
IAMBIC	TROCHAIC

(13)

Applying RCD to this pair generates a constraint hierarchy with all other constraints dominating IAMBIC. Production-directed parsing, applied to the trisyllabic underlying form with this hierarchy, produces candidate [0 (0 1)] as an optimal alternative. The resulting mark-data pair is (14).

Loser Marks	Winner Marks
ALL-FEET-LEFT MAIN-LEFT	ALL-FEET-RIGHT MAIN-RIGHT

(14)

Adding this to (13), and applying RCD, gives constraint hierarchy (15), which holds the desired interpretation as the sole optimal candidate.

Stratum 1		Stratum 2
PARSE ALL-FEET-LEFT MAIN-LEFT IAMBIC	≫	ALL-FEET-RIGHT MAIN-RIGHT TROCHAIC

(15)

This ends the processing of the first overt form. The learner currently has two grammar hypotheses. The first hypothesis, associated with the first interpretation, [0 (1 0)], is represented by mark-data pair list (16), which produces constraint hierarchy (12).

Loser Marks	Winner Marks
ALL-FEET-RIGHT MAIN-RIGHT	ALL-FEET-LEFT MAIN-LEFT
TROCHAIC	IAMBIC

(16)

The second hypothesis, associated, with the second interpretation, [(0 1) 0], is represented by mark-data pair list (17), which produces constraint hierarchy (15).

Loser Marks	Winner Marks
IAMBIC	TROCHAIC
ALL-FEET-LEFT MAIN-LEFT	ALL-FEET-RIGHT MAIN-RIGHT

(17)

	Parse	M-R	A-F-R	Trochaic	M-L	Iambic	A-F-L
[(2 0) (1 0)]			* *		* *	* *	* *
[0 0 (1 0)]	* *				* *	*	* *
[(0 1) 0 0]	* *	* *	* *	*			
[(0 2) (0 1)]			* *	* *	* *		* *

Table 10: Important Candidates for the four-syllable input.

4.2 The Second Overt Form

The second overt form encountered in this example is [2 0 1 0]. This overt form is unambiguous, supporting only the interpretation [(2 0) (1 0)], due to the grammar-imposed requirement that feet be bisyllabic. The learner will attempt to reconcile this interpretation with each of the two grammar hypotheses resulting from the previous overt form. The constraint violations for the correct interpretation and some significant competing descriptions for a four-syllable input are given in table 10.

4.2.1 First Hypothesis

First, learner attempts to reconcile their first grammar hypothesis (16) with the new interpretation, [(2 0) (1 0)]. Production-directed parsing is applied to a four-syllable word using the constraint hierarchy (12), resulting in the description [0 0 (1 0)]. This description is then used as a loser to form, along with winner [(2 0) (1 0)], mark-data pair (18).

Loser Marks	Winner Marks
Parse	All-Feet-Right Iambic

(18)

Adding this mark-data pair produces the list (19).

Loser Marks	Winner Marks
All-Feet-Right Main-Right	All-Feet-Left Main-Left
Trochaic	Iambic
Parse	All-Feet-Right Iambic

(19)

The application of RCD produces the constraint hierarchy (20).

Stratum 1		Stratum 2
Parse Main-Right Trochaic	$\gg$	All-Feet-Left Main-Left Iambic All-Feet-Right

(20)

This hierarchy makes the desired interpretation, [(2 0) (1 0)], the sole optimal description, so the reconciliation is successful, and the learner retains this (now refined) hypothesis.

4.2.2 Second Hypothesis

The learner then turns to the second hypothesis, (17), attempting to reconcile it with the same single interpretation, [(2 0) (1 0)]. Applying production-directed parsing to a four-syllable word using constraint hierarchy

(15) produces the structural description [(0 1) 0 0]. This description, together with the interpretation, forms mark-data pair (21).

Loser Marks	Winner Marks
Parse Main-Right Trochaic	Main-Left All-Feet-Left Iambic

(21)

Adding this mark-data pair produces the list (22).

Loser Marks	Winner Marks
Iambic	Trochaic
All-Feet-Left Main-Left	All-Feet-Right Main-Right
Parse Main-Right Trochaic	All-Feet-Left Main-Left Iambic

(22)

Applying RCD to list (22) results in constraint hierarchy (23).

Stratum 1		Stratum 2		Stratum 3
Parse	$\gg$	All-Feet-Left Main-Left Iambic	$\gg$	Main-Right All-Feet-Right Trochaic

(23)

The learner now re-applies production-directed parsing using (23). This results in description [(0 1) (0 2)], which forms mark-data pair (24).

Loser Marks	Winner Marks
Main-Right Trochaic	Main-Left Iambic

(24)

The full list of mark-data pairs is now (25).

Loser Marks	Winner Marks
Iambic	Trochaic
All-Feet-Left Main-Left	All-Feet-Right Main-Right
Parse Main-Right Trochaic	All-Feet-Left Main-Left Iambic
Main-Right Trochaic	Main-Left Iambic

(25)

Applying RCD to (25) results in constraint hierarchy (26).

Strat 1		Strat 2		Strat 3		Strat 4		Strat 5
Parse	$\gg$	A-F-L	$\gg$	M-R A-F-R	$\gg$	M-L Iambic	$\gg$	Troch

(26)

The persistent learner now applies production-directed parsing using (26). The resulting description, [(0 2) (0 1)] still doesn't match the desired interpretation, and forms mark-data pair (27).

Loser Marks	Winner Marks
Trochaic	Iambic

(27)

	Parse	M-R	A-F-R	Trochaic	M-L	Iambic	A-F-L
[0 (2 0) (1 0)]	*		* *		* * *	* *	* * * *
[(0 2) 0 (1 0)]	*		* * *	*	* * *	*	* * *
[(0 2) (0 1) 0]	*	*	* * * *	* *	* *		* *
[(2 0) 0 (1 0)]	*		* * *		* * *	* *	* * *

Table 11: Important Candidates for the five-syllable input.

The full list of mark-data pairs is now (28).

Loser Marks	Winner Marks
Iambic	Trochaic
All-Feet-Left Main-Left	All-Feet-Right Main-Right
Parse Main-Right Trochaic	All-Feet-Left Main-Left Iambic
Main-Right Trochaic	Main-Left Iambic
Trochaic	Iambic

(28)

The learner then applies RCD to (28), but does not get a constraint hierarchy. Instead, RCD informs the learner that the list is inconsistent. To see why, just compare the first and last pairs of the list. One requires Iambic to dominate Trochaic, while the other requires Trochaic to dominate Iambic. This tells the learner that this grammar hypothesis cannot be right; it cannot be reconciled with the interpretation [(2 0) (1 0)]. Therefore, the learner discards the hypothesis.

The learner has now finished processing the second overt form, and is now left with only one grammar hypothesis, the one in (19).

4.3 The Third Overt Form

The third form presented to the learner is [0 2 0 1 0]. This actually sustains three interpretations: [(0 2) (0 1) 0], [(0 2) 0 (1 0)], and the correct interpretation, [0 (2 0) (1 0)]. The constraint violation marks for the important structural descriptions are given in table 11. The learner is using hypothesis 19 with constraint hierarchy 20. The currently optimal descriptions of a five syllable underlying form are [(2 0) 0 (1 0)] and [0 (2 0) (1 0)].

4.3.1 First and Second Interpretations

The first two interpretations tried by the learner, [(0 2) (0 1) 0] and [(0 2) 0 (1 0)], both immediately lead to inconsistencies detected by RCD. This should not be surprising; each has at least one iambic foot, which is not achievable given that the only grammar hypothesis considered by the learner at the beginning of this step explicitly requires Trochaic to dominate Iambic.

4.3.2 The Correct Interpretation

The learner then considers the interpretation [0 (2 0) (1 0)]. The description [(2 0) 0 (1 0)] is selected as the loser, as it is a currently optimal structural description and not identical to the current interpretation

(which will be the winner). The resulting mark-data pair is (29).

Loser Marks	Winner Marks
All-Feet-Right	All-Feet-Left

(29)

The complete list of mark-data pairs is (30).

Loser Marks	Winner Marks
All-Feet-Right Main-Right	All-Feet-Left Main-Left
Trochaic	Iambic
Parse	All-Feet-Right Iambic
All-Feet-Right	All-Feet-Left

(30)

Applying RCD to the list produces constraint hierarchy (31).

Stratum 1		Stratum 2		Stratum 3
Parse Main-Right Trochaic	$\gg$	All-Feet-Right Main-Left Iambic	$\gg$	All-Feet-Left

(31)

This constraint hierarchy is adequate to generate the entire Warao stress pattern. Because this is the learner's sole hypothesis at this point, no further learning will occur. For every subsequent overt form, the single optimal structural description will match the overt form, with no disparity to trigger learning.

5 Simulation Results

By design, the strategy of trying all possible interpretations is guaranteed to find a correct constraint ranking for a language, provided that the data (the overt forms) presented are all consistent with some language realizable by the system, each overt form has only a finite number of possible interpretations, and each underlying form is apparent from its corresponding overt form. The interesting question concerns the amount of work required to obtain a correct ranking. The effectiveness of the strategy was tested empirically on several cases of optimality theoretic systems for metrical stress.

Because the primary interest is in how well the strategy contends with the size of the number of combinations of interpretations of the overt forms, the measure of effort presented here is the number of times, during the course of learning, that a loser-winner pair is added to the list for a grammar hypothesis (triggering the application of RCD to that list). This includes all the additions made to lists that are ultimately discarded; an important part of the measure of work is the amount of work required to test out and eliminate the inconsistent combinations. The number of applications of RCD prior to reaching a correct constraint hierarchy will include one for every mark-data pair in the list giving rise to the correct hierarchy, along with one for each occurrence of a mark-data pair in a list determined to be inconsistent (and thus discarded).

For the illustration in section 4, the total number of applications of RCD is 11. Listed by section: section 4.1.1 has 2 RCD applications, section 4.1.2 has 2 applications, section 4.2.1 has 1 application, section 4.2.2 has 3 applications, section 4.3.1 has 2 applications, and section 4.3.2 has 1 application. Four of those applications are accounted for by the four mark-data pairs in list (30), which give rise to the learned constraint hierarchy (31). The other 7 applications were involved in eliminating inconsistent (and therefore incorrect) combinations.

Name	Description
PARSE	a syllable must be footed
MAIN-RIGHT	align the head-foot with the word, on the right edge
MAIN-LEFT	align the head-foot with the word, on the left edge
ALL-FEET-RIGHT	align each foot with the word, on the right edge
ALL-FEET-LEFT	align each foot with the word, on the left edge
IAMBIC	align the head syllable with its foot, on the right edge
TROCHAIC	align the head syllable with its foot, on the left edge
WORD-FOOT-LEFT	align the word with a foot, on the left edge
WORD-FOOT-RIGHT	align the word with a foot, on the right edge
NON-FINAL	the right-most syllable should not be footed

Table 12: The constraints for the bisyllabic metrical system.

5.1 Case 1: The Bisyllabic System

The first optimality theoretic system investigated is the system described in section 2.1, with three additional constraints (giving a total of ten). The constraints are displayed in table 12. The constraints WORD-FOOT-LEFT and WORD-FOOT-RIGHT also come from (McCarthy and Prince, 1993). The constraint NON-FINAL captures effects traditionally analyzed as extra-metricality (Liberman and Prince, 1977) (Hayes, 1980) (Prince and Smolensky, 1993). The GEN function is the same as in the simple system, permitting only structural descriptions with strictly bisyllabic feet. Despite the great number of possible total rankings of the constraints, there are a total of 56 possible languages in this system. The learning procedure was applied, in turn, to the overt forms of each of the 56 languages, and the number of applications of RCD was measured for each. The results are shown in (32).

Number of Languages	Number of RCD Applications		
	Median	Minimum	Maximum
56	8	1	23

(32)

As the results indicate, the procedure converges on a correct hierarchy extremely rapidly; in the majority of cases, the total number of steps is less than the number of constraints, which is 10.

The extreme speed of learning is the result of several factors. Perhaps the most significant is that the degree of ambiguity of the overt forms is quite limited, due to the restriction of GEN to candidates with strictly bisyllabic feet. Overt forms like [1 0 2 0 0] are completely unambiguous in this system: the only possible structural description matching the overt form is [(1 0) (2 0) 0]. The overt form [0 1 0 2 0 2 0] has 4 interpretations. It is worth considering just how much ambiguity is in the sets of forms being learned. Each language consists of six overt forms of from 2 to 7 light syllables. The total number of combinations of interpretations for a language may be computed by multiplying together the number of interpretations for each overt form in the language. Thus, for each language, the total number of combinations is a product of six numbers. The number of combinations for the 56 languages is summarized in (33).

Number of Languages	Number of Overt Forms	Number of Interpretation Combinations		
		Median	Minimum	Maximum
56	6	8	1	192

(33)

If this restriction of GEN to only bisyllabic feet were fully supported linguistically, then these results alone might make a case for the claim that metrical structure is learned quite easily. However, this restriction

Name	Description
PARSE	a syllable must be footed
MAIN-RIGHT	align the head-foot with the word, on the right edge
MAIN-LEFT	align the head-foot with the word, on the left edge
ALL-FEET-RIGHT	align each foot with the word, on the right edge
ALL-FEET-LEFT	align each foot with the word, on the left edge
IAMBIC	align the head syllable with its foot, on the right edge
FOOT-NON-FINAL	the head syllable must not be rightmost in its foot
WORD-FOOT-LEFT	align the word with a foot, on the left edge
WORD-FOOT-RIGHT	align the word with a foot, on the right edge
NON-FINAL	the right-most syllable must not be footed
FOOT-BINARITY	a foot must have two moras or two syllables
WSP	a heavy syllable must be footed

Table 13: The constraints for the full syllabic/moraic metrical system.

is actually quite a strong simplification; there is abundant evidence for monosyllabic feet. What might be the effect of removing this restriction on GEN?

5.2 Case 2: The Full Syllabic/Moraic System with Only Light Syllables

The second optimality theoretic system permits a much larger range of candidate structural descriptions: feet may now consist of either one or two syllables. The underlying forms are also significantly enhanced, with each syllable now labeled as either light or heavy. This permits constraints which are sensitive to both syllabic and moraic quantity. The system has a total of 12 constraints, listed in table 13. Most of the constraints of the earlier system appear here, enhanced by some others. The constraint TROCHAIC is replaced by the constraint FOOT-NON-FINAL, which is identical in definition to Kager's MAX-FT (Kager, 1994). This constraint enables the system to capture the typological absence of quantity-insensitive iambic systems (Hayes, 1995). Quantity sensitivity effects result from both the relative unmarkedness of monosyllabic feet with a heavy syllable, and the presence of the weight-to-stress constraint WSP (Prince, 1990).

One set of learning simulations run with the full system used exactly the same overt forms as used with the purely bisyllabic system. Specifically, all the forms had only light syllables; forms with heavy syllables were not presented to the learner. While the correct interpretations involve bisyllabic feet, the learner now has to figure that out as part of learning. Structural descriptions with monosyllabic feet, including monomoraic feet, are possible now, both as candidates and as substructures of optimal descriptions for some constraint rankings. The overt form [1 0 2 0 0] with all light syllables, which had only one interpretation in the previous system, now has five distinct interpretations that are valid candidates, as shown in (34).

$$\begin{array}{l} [(1)\ 0\ (2)\ 0\ 0] \\ [(1\ 0)\ (2)\ 0\ 0] \\ [(1)\ 0\ (2\ 0)\ 0] \\ [(1\ 0)\ (2\ 0)\ 0] \\ [(1)\ (0\ 2)\ 0\ 0] \end{array} \tag{34}$$

The overt form [0 1 0 2 0 2 0] has 21 distinct interpretations.

To see how the ambiguity has increased in this system, compare the number of combinations of interpretations for the same languages of overt forms to the number of combinations for the strictly bisyllabic system (which was given in (33) above). Each language is the same as in the bisyllabic system, consisting of six overt forms of from 2 to 7 light syllables. The total number of combinations of interpretations for a language is again the product of the number of interpretations for each overt form in the language. The number of combinations for the 56 languages under the new metrical system are summarized in (35).

Number of Languages	Number of Overt Forms	Number of Interpretation Combinations		
		Median	Minimum	Maximum
56	6	16,900	64	65,520

(35)

Half of the languages have overt forms supporting a total of 16,900 or more different combinations of interpretations; the maximum number of combinations in the strictly bisyllabic system was 192. Exhaustively enumerating and checking all combinations of interpretations could require checking as many as 65,520 combinations. Further, the evaluation of each combination via error-driven constraint demotion requires several rounds of demotion, so the number of applications of RCD would be several times that number. Notice that this is still nowhere near the number of distinct total rankings of the constraints, which is $12! = 479,001,600$. But the procedure investigated here can do much better than enumerate all possible combinations.

The results of the simulations for the new metrical system, run on only the light syllable forms, are shown in (36).

Number of Languages	Number of RCD Applications		
	Median	Minimum	Maximum
56	42	2	93

(36)

As expected, the number of steps is typically larger than for the strictly bisyllabic system. But, the number of steps is still quite low, less than the square of the number of constraints ($12^2 = 144$). Further, the increase in the number of steps is nothing like the increase in the number of possible combinations of interpretations. Moving to the metrical system with a more realistic GEN greatly increased the number of possible combinations of interpretations, but the ability of MRCD to preserve information across forms and quickly detect inconsistencies permits it to converge without having to explicitly consider the vast majority of combinations of interpretations.

5.3 Case 3: The Full Syllabic/Moraic System with Full Inputs

While those results show the ability to overcome greater ambiguity, they do not test the full potential of the OT system. That requires testing it on forms with heavy as well as light syllables. For a given language, learning the constraint ranking means, in part, determining if the language is quantity-sensitive, and, if so, the degree of sensitivity.

The next set of results was obtained from languages consisting of a total of 62 forms: words of length 2 to 5 with all possible combinations of light and heavy syllables (60 words in all), along with words of 6 and 7 light syllables. Each language was generated by at least one total ranking of the 12 constraints of the system.

The great increase in the number of distinct overt forms in the languages (from 6 to 62) makes the exact number of possible combinations of interpretations of the overt forms not really worth computing. Given that every overt form has at least two interpretations, and the vast majority of overt forms have more than two possible interpretations, the number of possible combinations of interpretations will typically be much, much greater than $2^{62} = 4,611,686,018,427,387,904$. This figure dwarfs the number of total rankings of

12 constraints. If the learner cannot completely escape the combinatorial growth of the combinations of interpretations, then this approach is hopeless.

The simulation results, shown in (37), offer plenty of hope.

Number of Languages	Number of RCD Applications		
	Median	Minimum	Maximum
124	50	8	160

(37)

Despite the explosive growth in the number of combinations of interpretations, the learner is able to quickly arrive at a correct constraint hierarchy, doing so in 50 applications of RCD on average.

124 languages were tested in this case. For each of the 56 languages previously tested, there were two in this test set. The two languages each had the same stress pattern on the light syllable forms as in the corresponding light-only language, and differed from each other in the degree of quantity sensitivity displayed by the forms with heavy syllables. Thus, 112 of the 124 languages tested matched the previous languages on the overt forms, while the other 12 languages had stress patterns that deviated even for forms with only light syllables.

6 Discussion

Multi-Recursive Constraint Demotion, the maintenance and use of a list of mark-data pairs to represent a grammatical hypothesis, allows a learner to retain information that is not recoverable from a constraint hierarchy alone. The learner can efficiently determine if a list of mark-data pairs is consistent by applying Recursive Constraint Demotion. When the list is consistent, the very same computation produces a constraint hierarchy holding all the harmonic relations contained in the list. By adding new pairs to the list as learning progresses, the learner can refine their working constraint hierarchy without losing information obtained on earlier forms. The learner can also determine quite rapidly if the interpretation currently being pursued is inconsistent with other information already obtained by the learner.

MRCD can in principle be used in conjunction with a variety of strategies for hypothesizing interpretations of overt forms. One particular strategy for hypothesizing interpretations of overt forms is presented in this paper: the generation and consideration of all possible interpretations of an overt form. This strategy plays it safe by considering all interpretations of an overt form when learning is necessary, while still avoiding the explosive growth in the number of combinations of interpretations across different overt forms.

The performance of this combination (MRCD combined with the consideration of all possible interpretations) is due to several factors. The use of constraint demotion avoids the combinatorial growth of possible constraint rankings, detangling complex constraint interactions in a reasonable time. The error-driven nature of the algorithm contributes in two ways simultaneously. First, it makes it possible to identify informative competitors for the formation of mark-data pairs, generating only as many pairs as necessary. Second, it allows the learner to avoid spending extra time processing overt forms when unnecessary; if a current grammar hypothesis already holds as optimal a description matching the observed overt form, the learner doesn't waste time examining other possible interpretations of the overt form. The use of MRCD, keeping the list mark-data pairs in addition to the hypothesized ranking, preserves enough information that the learner is able to determine when a given interpretation is inconsistent with data seen earlier, allowing many possible interpretations to be eliminated immediately upon consideration. The quick elimination of many interpretations allows the learner to avoid the exponential growth of the number of combinations of interpretations across all of the overt forms of the language.

Another factor contributing to the performance of the learning algorithm is the order in which the overt forms are processed. The learner processes them in order of increasing size, shortest to longest. The shorter forms typically have fewer possible interpretations. By processing shorter forms first, the learner can learn quite a bit about the language before moving to longer forms. This allows the learner to use what they have already learned to quickly eliminate many interpretations of longer forms. Further, if the shorter forms contain enough information to determine the entire language, then the learner need never consider multiple interpretations of the longer forms; the error-driven nature of the algorithm allows it to avoid lengthy processing of forms on which the learner is already producing the correct stressing. This suggests a rather plausible general learning strategy: in the early stages of learning, focus processing effort on forms that have relatively limited ambiguity (for example, short forms).

The complexity of learning for OT systems can be divided into a couple of components. The number of constraints is one source of complexity, and its effect is observed in the number of mark-data pairs needed to form a list which completely determines a language in the system. The more constraints, the longer the list may need to be. Here, formal analysis provides a guaranteed upper bound: the required length of the list cannot exceed a figure on the order of the square of the number of constraints: to be precise, $\frac{(N-1)N}{2}$, where N is the number of constraints.

The other source of complexity is the degree of ambiguity in the overt forms of the system, along with the degree of interdependence of the different forms. The effect of the degree of ambiguity is seen in the number of applications of RCD to lists made during the course of learning. If the overt forms of the system have a great deal of ambiguity, then the learner will need to make lots of RCD applications to eliminate considered interpretations whenever it needs to learn an overt form. The interdependence of the overt forms has an inverse relationship to learning complexity: the greater the interdependence, the greater the number of possible interpretations that can be eliminated immediately upon consideration, reducing the number of grammar hypotheses maintained by the learner from one overt form to the next.

Other work on learning in Optimality Theory (Tesar, to appear) (Tesar, 1997) has investigated an iterative strategy for selecting interpretations of overt forms. In that work, the strategy is to select, out of all possible interpretations of an overt form, the interpretation which is most harmonic given the learner's current constraint hierarchy. The selected interpretation then provides the basis for modification of the constraint hierarchy, towards the goal of making the selected interpretation optimal. Every time the constraint hierarchy is changed, the learner recomputes the optimal interpretation, which may have changed as a result of the change in ranking. That strategy requires less computational effort on any one overt form than the one discussed in this paper, because it pursues only a single interpretation of an overt form at any time, rather than considering all of them. While that strategy shows significant promise, it is not guaranteed to succeed; it is possible for the currently optimal description to be incorrect. It uses less computational effort, at the cost of occasionally failing to converge to a correct constraint hierarchy. Taken together, that single-interpretation strategy and the all-interpretations strategy informally define a space of strategies for selecting interpretations of optimal forms.

It is not clear, however, if the savings in computational effort for the iterative strategy is significant. On the full metrical stress system used in section 5.3, the overall computational requirements of MRCD is rather modest. The actual computational effort required by this approach is of course dependent upon the details of the particular optimality theoretic system being used. If the trend suggested by the simulation results holds generally, then the computational effort required by the MRCD approach may be a small price to pay in exchange for guaranteed convergence to a correct constraint hierarchy.

References

Chomsky, Noam. 1981. *Lectures on Government and Binding.* Foris Publications, Dordrecht.

Daelemans, Walter, Steven Gillis, and Gert Durieux. 1994. The acquisition of stress: A data-oriented approach. *Computational Linguistics*, 20(3):421–451.

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

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

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

Hale, Mark and Charles Reiss. 1996. Formal and empirical arguments concerning phonological acquisition. Unpublished Ms., Concordia University.

Hayes, Bruce. 1980. *A Metrical Theory of Stress Rules.* Ph.D. thesis, Massachusetts Institute of Technology, Cambridge. [Revised version published by Garland Press, New York 1985].

Hayes, Bruce. 1995. *Metrical Stress Theory: Principles and Case Studies.* The University of Chicago Press, Chicago.

Kager, René. 1994. Ternary rhythm in alignment theory. Ms., Research Institute for Language and Speech, Utrecht University.

Liberman, Mark and Alan Prince. 1977. On stress and linguistic rhythm. *Linguistic Inquiry*, 8:249–336.

McCarthy, John and Alan Prince. 1993. Generalized alignment. In Geert Booij and Jaap Van Marle, editors, *Yearbook of Morphology*, pages 79–154, Dordrecht. Kluwer.

Osborn, Henry. 1966. Warao I: Phonology and morphophonemics. *International Journal of American Linguistics*, 32:108–123.

Prince, Alan. 1990. Quantitative consequences of rhythmic organization. In K. Deaton, M. Noske, and M. Ziolkowski, editors, *CLS26-II: Papers from the Parasession on the Syllable in Phonetics and Phonology.* pages 355–398.

Prince, Alan and Paul Smolensky. 1993. Optimality Theory: Constraint interaction in generative grammar. Technical report, TR-2, Rutgers University Cognitive Science Center, and CU-CS-696-93, Department of Computer Science, University of Colorado at Boulder. To appear in the Linguistic Inquiry Monograph Series, MIT Press.

Smolensky, Paul. 1996. On the comprehension/production dilemma in child language. *Linguistic Inquiry*, 27. (ROA-118).

Tesar, Bruce. 1996. Error-driven learning in Optimality Theory via the efficient computation of optimal forms. In *Proceedings of the Workshop on Optimality Theory in Syntax: Is the Best Good Enough?*, Cambridge, MA. MIT Press.

Tesar, Bruce. 1997. An iterative strategy for learning metrical stress in Optimality Theory. In Elizabeth Hughes, Mary Hughes, and Annabel Greenhill, editors, *The Proceedings of the 21st Annual Boston University Conference on Language Development*, pages 615–626, Somerville, MA. Cascadilla Press.

Tesar, Bruce. to appear. An iterative strategy for language learning. *Lingua*.

Tesar, Bruce and Paul Smolensky. 1995. The learnability of Optimality Theory. In *Proceedings of the Thirteenth West Coast Conference on Formal Linguistics*, pages 122–137.

Tesar, Bruce and Paul Smolensky. to appear. The learnability of Optimality Theory: An algorithm and some basic complexity results. *Linguistic Inquiry*. (ROA-155).