To appear in *Proceedings of WCCFL XVII*.

Robust Interpretive Parsing in Metrical Stress Theory

BRUCE TESAR
Rutgers University

1. Introduction

Most computational work to date within Optimality Theory (Prince and Smolensky 1993) has focused on generation, the mapping from an underlying form to its full structural description (Eisner 1997, Ellison 1995, Frank and Satta in press, Tesar 1995). While such a computation, labeled *production-directed parsing* by Tesar and Smolensky (1998), is a natural one, it is not the only function of interest related to an OT grammar. Language comprehension involves the interpretation of *overt forms*, forms consisting of the auditory information directly available to a listener. The function is not the inverse of generation, but the mapping from an overt form to its full structural description. The process of computing this latter function is *interpretive parsing*. Tesar and Smolensky (1996, 1998) have proposed that a particular form of interpretation, robust interpretive parsing (discussed below), has an important role to play not only in language comprehension but in language acquisition (see also Smolensky 1996). Hammond (1997) has proposed a procedure specific to syllable parsing which is interpretive in nature, although it does not strictly enforce conformity of the interpretation to the overt form. Hammond's procedure is not "robust" in the sense discussed in this paper.

This paper presents an efficient algorithm for the interpretive parsing of forms for optimality theoretic systems for metrical stress. This algorithm has been used in learning simulations that have been presented elsewhere (Tesar 1997, in press). The system used for illustration in this paper has 8 freely rankable constraints, and can account for core metrical phenomena (the same basic algorithm works for several metrical stress systems accommodating a larger variety of stress phenomena; see section 2.1 below). The underlying forms are strings of syllables, and the full structural descriptions include foot structure and the assignment of stress levels to the syllables. The overt forms include the syllables and their stress levels, but not the foot structure. Overt forms are inherently ambiguous; the same pattern of stress levels is consistent with multiple foot structures. For example, a tri-syllabic overt form with main stress on the middle syllable, [˘˘̀˘], is consistent with three distinct interpretations: a left-aligned iambic foot, [(˘ ˘̀)˘], a right-aligned trochaic foot, [˘ (˘̀ ˘)], and a monosyllabic foot containing only the stressed syllable, [˘ (˘̀) ˘]. Interpretive parsing must recover the correct foot structure from the overt form, using the constraint ranking of the grammar. The algorithm presented here is based upon the dynamic programming approach of Tesar (1995), but includes extensions to deal with the non-localities of metrical structure.

Interpretive parsing poses an interesting issue that does not arise in generation: the issue of dealing with ungrammatical overt forms. By the principle of richness of the base (Prince and Smolensky 1993), an optimality theoretic grammar assigns a description to every possible input. But there can be overt forms which do not correspond to any optimal structural description of a grammar. Most traditional parsing techniques (such as those used in syntactic parsing) are designed to identify and reject ungrammatical overt forms. However, in language acquisition, the learner cannot simply reject and ignore forms which are inconsistent with their current grammar. To the contrary, such data are precisely what a learner needs to attend to; they indicate that the learner needs to modify their grammar, in such a way as to render the forms grammatical.

The point demonstrated here is that ungrammatical (by a learner's current grammar) overt forms are nevertheless interpretable. The interpretive parsing algorithm presented in this paper assigns the best possible interpretation to an overt form, whether it is grammatical or not. Thus, the procedure is called *robust* interpretive parsing. The ability to interpret ungrammatical overt forms depends crucially on the framework of Optimality Theory; the algorithm selects, from among the possible interpretations of the overt form, that interpretation which best satisfies the ranked constraints.

Robust interpretive parsing has great significance for language learning. A child in the process of learning their native language does not yet have the correct grammar, and thus cannot rely on their grammar to judge the grammaticality of the overt forms they hear. Robust interpretive parsing allows a child to use what they have already learned to estimate a best interpretation of overt forms, which they can then use to perform further learning. Significant learning results, reported elsewhere, have already been obtained using this approach to learning. Those learning results are critically dependent upon the robust interpretive parsing procedure presented in this paper.

2. Computing Optimal Descriptions in Optimality Theory

Tesar (1995) developed algorithms that computing optimal descriptions for several classes of Optimality Theoretic grammars. Those algorithms compute production-directed parsing. A primary focus of that work was on algorithms that could deal efficiently with *faithfulness* in OT systems, that is, grammars that permit insertion and deletion of segments.

The types of OT systems discussed here don't involve faithfulness. The underlying form is an ordered list of syllables, and the grammar assigns stress levels to the syllables; no insertion or deletion of syllables is considered. However, the same basic dynamic programming approach works for these systems. This section describes an algorithm for production-directed parsing of metrical stress: given a string of syllables and a ranking of the constraints, this algorithm computes the optimal metrical structure for that string of syllables. The robust interpretive parsing algorithm, presented in section 3, will be based closely on the production-directed parsing algorithm presented here, differing only in the inclusion of one additional restriction.

2.1 The Metrical Stress System

The optimality theoretic analysis of stress described here is simplified for purposes of presentation; the eight constraints presented here are a subset of the constraints used in a larger Optimality Theoretic system of metrical stress grammars, used in the learning simulations investigated by Tesar (1997, in press). That larger Optimality Theoretic system uses ideas from several sources (McCarthy and Prince 1993, Prince and Smolensky 1993, Prince 1990, Hayes 1995, Hayes 1980), and includes analyses of quantity sensitivity and non-finality/extrametricality effects. The parsing approach presented in this paper is equally effective on the larger metrical system.

The metrical stress system for this paper has as possible inputs words consisting of strings of syllables, each labeled for weight (light or heavy). A candidate structural description for an input is a grouping of (some of) the syllables of the input into feet, under the following conditions: (i) a foot contains either one or two syllables; (ii) each foot assigns stress to exactly one of its syllables; (iii) each candidate has exactly one head foot assigning main stress, with any other feet assigning secondary stress. The system has 8 constraints, listed in (1).

(1)	PARSE	a syllable must be footed
	MR	the head-foot must be rightmost in the word
	ML	the head-foot must be leftmost in the word
	AFR	a foot must be aligned with the right word edge
	AFL	a foot must be aligned with the left word edge
	IAMB	a head syllable must be rightmost its foot
	TROCH	a head syllable must be leftmost in its foot
	FOOTBIN	a foot must have two moras or two syllables

2.2 Categories for Candidates

The key idea is to build up candidate structural descriptions in stages, one syllable at a time. This section will illustrate the basic ideas, abstracting away from distinctions of main vs. secondary stress (that distinction will be added in the section 2.4). The algorithm is designed around a data structure, called the dynamic programming table. This table is used to store partial structural descriptions as they are constructed.

The ranking used for this first illustration is given in (2).

(2) FOOTBIN » PARSE » AFR » TROCH » {AFL, IAMB}

Table 1 shows the completed dynamic programming table for the input / ˘ ¯ ˘ ˘ /, using the constraint ranking in (2). Each syllable of the input is the head of a column in the table. Observe that the column headed by the first syllable contains several partial structural descriptions, each containing the first syllable. The next column, headed by the second syllable (which happens to be heavy in this example), contains partial structural descriptions containing the first two syllables. Each partial description in the second column is obtained by taking one of the partial descriptions in the first column, and adding the second syllable to it in one way or another. The final column, headed by the final syllable, has structural descriptions containing all four syllables. The optimal description will be selected from among the candidates in this final column.

	⏑	–	⏑	⏑
NoF	[⏑]	[⏑ –]	[(⏑̀ –) ⏑]	[⏑ (–̀ ⏑) ⏑]
F1NoS	[(⏑)]	[⏑ (–)]	[(⏑̀ –) (⏑)]	[⏑ (–̀ ⏑) (⏑)]
F1S	[(⏑̀)]	[⏑ (–̀)]	[(⏑̀ –) (⏑̀)]	[⏑ (–̀ ⏑) (⏑̀)]
F2		[(⏑̀ –)]	[⏑ (–̀ ⏑)]	[(⏑̀ –) (⏑̀ ⏑)]

Table 1: Production-Directed Parsing for / ⏑ – ⏑ ⏑ / (without main stress).

The labels at the far left of the rows indicate the categories for the partial structural descriptions in the rows. The label **NoF** (no-foot) means that the syllable just added to a partial description (the rightmost syllable) is unfooted. In the column for the first syllable, the partial description in row **NoF** contains the single syllable unfooted. In the subsequent cells in the row labeled **NoF**, the rightmost syllable of each partial description is unfooted. Partial descriptions are grouped into categories based upon the condition of the right-most syllable in the partial description. The four categories shown in the table are the four different ways of parsing a syllable into metrical structure.

The category **F1NoS** (foot-one-no-stress) means that the new syllable (the rightmost syllable) is the first syllable of a foot, and that the new syllable is unstressed. Because each foot must contain precisely one stressed syllable (the head syllable of the foot), a partial description in this row will only be a part of valid structural descriptions in which the next syllable is stressed and added to the same foot. Such a foot would be a bi-syllabic, iambic foot. The category **F1S** (foot-one-stress) means that the new syllable is the first syllable of a foot, and that the new syllable is stressed. Such a partial description can be legally extended in several ways: the next syllable could be added as unstressed to the same foot, creating a bi-syllabic, trochaic foot; or, the next syllable could be added outside the foot (either unfooted or beginning a new foot), leaving the current foot as monosyllabic.

The category **F2** (foot-two) means that the new syllable is the second syllable of a foot. Because feet are maximally bi-syllabic, this category need not be partitioned based on whether the new syllable is stressed or not; either the new syllable or the one before it must be stressed. A foot with two unstressed syllables is universally ill-formed, and is considered here to be banned by GEN from possible candidate descriptions. Thus, the algorithm will not generate or consider such structures.

Notice that in the column for the first syllable, there is no entry in the row for category **F2**. This is because at this point only one syllable has been processed, and the **F2** is only for partial descriptions ending on a foot with two syllables. Because insertion/deletion of syllables is not permitted in this system, there is no way to construct a partial description satisfying the requirements of that cell.

2.3 Filling the Table

The descriptions in the cells of the column for the first syllable are the different ways of parsing the first syllable. Cells in subsequent columns are filled (in order) by considering ways of adding the new syllable to the partial descriptions of the previous column. The key to the operation of the algorithm is that constraint violations can be assessed to partial descriptions, so that they may be compared with respect to the constraint hierarchy.

Consider the cell in the column for the second syllable and the row **NoF**. Only partial descriptions with the second syllable unfooted can compete to fill this cell. There are two such candidates, shown along with some of their constraint violations in the tableau in (3). Candidate (3a) extends the partial description from the same row in the first column, [˘]. Candidate (3b) extends the partial description from row **F1S** in the first column, [(˘̀)].

(3) The candidate partial descriptions of the first two syllables competing to fill cell **NoF**.

		FOOTBIN	PARSE	AFR	TROCH
a.	[˘ –]		* *		
b.	[(˘̀) –]	*!	*	*	

Candidate (3b) has a violation of FOOTBIN (the top-ranked constraint), while its competitor does not, so it loses, and candidate (3a) is placed in the cell. By excluding the losing partial description at this point, the algorithm has successfully dismissed all structural descriptions of the entire word which have the first syllable footed by itself and the second syllable unfooted, without having to explicitly generate, evaluate, and compare all such candidates.

Consider two different partial structural descriptions of the first two syllables, candidates (4a) and (4b). After seeing only two syllables, it is premature to determine which of these two is more harmonic. The reason is that the (non-)existence of additional syllables in the word is crucial. If the word has no more syllables, then the foot in (4a) satisfies AFR, while (4b) violates PARSE. If there are more syllables to the word, then the foot in (4a) will ultimately incur a violation of AFR for each additional syllable. If there is exactly one more syllable, it would be added onto (4a) unfooted to become (4c) (to satisfy FOOTBIN), incurring violations of both PARSE and AFR, while the foot in (4b) could be extended as (4d), incurring a violation of PARSE but not of AFR. So, after only two syllables have been processed, both (4a) and (4b) need to be retained, pending further information.

(4) Candidates for the first two syllables, and possible extensions with the third syllable.

		FOOTBIN	PARSE	AFR	TROCH
a.	[(◡̀ –)]				
b.	[◡ (–̀)]		*		
c.	[(◡̀ –) ◡]		*	*	
d.	[◡ (–̀ ◡)]		*		

It is this issue that is successfully addressed by the row categorization of the dynamic programming table. Because **F1S** and **F2** are separate categories, the partial descriptions (4a) and (4b) don't compete with each other in the second column. Each is attempting to fill a *different* cell.

The appropriate time to directly compare two partial descriptions is when they qualify for the same cell. The categories defining the rows are designed so that the constraint violations incurred by the parsing of the remaining syllables will be dependent only on the category (the way in which the last syllable is parsed), and not on structural details that would distinguish different partial descriptions seeking to fill the same cell.

Consider the cell in the column for the third syllable and the row **NoF**. Candidates (5a), (5b), and (5c) compete to fill the cell. The less harmonic members of the set may be discarded, because the constraint violations resulting from the addition of subsequent structure will be the same for all of these candidates. In (5d), (5e), and (5f), an additional foot is added onto (5a), (5b), and (5c), respectively (NOTE: this discussion in the rest of this section is not a proper continuation of the example from table 1). For the constraints shown, the violations to the right of the dotted line are the additional violations incurred when the new foot is added. The new violations are identical for each of (5d), (5e), and (5f). For any ranking, if (5a) is more harmonic than (5b) and (5c), then it is guaranteed that (5d) will be more harmonic than (5e) and (5f). Thus, all that needs to be retained is the most harmonic of (5a), (5b), and (5c). The unfooted third syllable which qualifies the partial descriptions for the category **NoS** acts as a shield: the additional constraint violations incurred by subsequent parsing decisions will not depend on the candidates' structure to the left of the unfooted third syllable.

(5) Candidates for the same cell (a, b and c) will incur identical additional violations when more structure is added (d, e and f). The additional violations are shown after the dotted lines.

		PARSE		AFR		AFL	
a.	[(˘̀ –) ˘]	*		*			
b.	[˘ (–̀) ˘]	* *		*		*	
c.	[(˘̀) – ˘]	* *		* *			
d.	[(˘̀ –) ˘ (˘̀ ˘)]	*		*	* *		* * *
e.	[˘ (–̀) ˘ (˘̀ ˘)]	* *		*	* *	*	* * *
f.	[(˘̀) – ˘ (˘̀ ˘)]	* *		* *	* *		* * *

The additional violations of AFR shown for candidates (5d), (5e), and (5f) are all for the lack of alignment of the foot containing the first two syllables of each partial description. When candidates (5a), (5b), and (5c) are compared (when filling the **NoF** cell in the column for the third syllable), these violations have not yet been assessed, because the additional syllables added to the word have not yet been seen. But this is not a problem. Candidate (5f), which has more violations (four) of AFR than (5d) or (5e), would be derived from candidate (5c), which has more violations (two) of AFR than (5a) or (5b). Thus, when AFR is highly ranked, eliminating (5c) in favor of (5a) when filling the **NoF** cell in the column of the third syllable is correct and desirable; for any candidate of the whole word containing (5c), there will be a more harmonic candidate containing (5a).

2.4 Keeping Track of Main Stress

GEN requires that every prosodic word have exactly one main stress. The actions of the processor must reflect and enforce this condition. Thus, care must be taken not to add a syllable with main stress onto a partial description already containing a syllable with main stress.

This is handled by the algorithm by splitting each of the previous row categories into two variants: one for partial descriptions in which no main stress has yet been assigned, and one for descriptions in which main stress has been assigned. The algorithm can only consider assigning main stress to a new syllable if it is adding the syllable to a partial description from a cell in a no-main-stress category to create a candidate for a cell in a main-stress category. As a result, there are a total of eight categories, as shown in table 2. The first four categories, with the prefix **NoM-**, are for partial descriptions lacking a main stress, while the second four categories, with the prefix **M-**, are for partial descriptions already containing a main stress.

The constraints specific to the head foot bearing main stress, ML and MR, can now be added to the working constraint hierarchy. The complete hierarchy is given in (6). One constraint on the location of main stress, ML, is near the top of the hierarchy.

(6) FOOTBIN » ML » PARSE » AFR » TROCH » {AFL, MR, IAMB}

	◡	–	◡	◡
NoMNoF	[◡]	[◡ –]	[(◡̀ –) ◡]	[◡ (–̀ ◡) ◡]
NoMF1NS	[(◡)]	[◡ (–)]	[(◡̀ –) (◡)]	[◡ (–̀ ◡) (◡)]
NoMF1S	[(◡̀)]	[◡ (–̀)]	[(◡̀ –) (◡̀)]	[◡ (–̀ ◡) (◡̀)]
NoMF2		[(◡̀ –)]	[◡ (–̀ ◡)]	[(◡̀ –) (◡̀ ◡)]
MNoF		[(◡́) –]	[(◡́ –) ◡]	[(◡́ –) ◡ ◡]
MF1NoS		[(◡́) (–)]	[(◡́ –) (◡)]	[(◡́ –) ◡ (◡)]
MF1S	[(◡́)]	[◡ (–́)]	[(◡́ –) (◡̀)]	[(◡́ –) ◡ (◡̀)]
MF2		[(◡́ –)]	[◡ (–́ ◡)]	[(◡́ –) (◡̀ ◡)]

Table 2: Production-Directed Parsing for the Input / ◡ – ◡ ◡ / using all categories, and including main stress.

Once all of the cells in the table have been filled, the algorithm selects the optimal description from among those in the last column. Not all entries in the final column are considered, however; several are not valid full structural descriptions. None of the candidates in the first four rows qualify as full structural descriptions, because they do not contain a main stress, and any well-formed full description must include a main stress. Among the four categories including main stress, the **MF1NoS** category is also ruled out, because the final foot does not have a head (a stressed syllable).

The optimal candidate, then, will be one of the entries in the final column for row categories **MNoF**, **MF1S**, and **MF2**. A tableau showing these three candidates and their violations of the top few constraints is shown in (7).

(7) The final three full candidate descriptions and their high-ranked constraint violations; candidate c is the optimal one.

		FOOTBIN	ML	PARSE	AFR	TROCH
a.	[(˘́ ¯) ˘ ˘]			*! *	* *	
b.	[(˘́ ¯) ˘ (˘̀)]	*!		*	* *	
c.	[(˘́ ¯) (˘̀ ˘)]				* *	

The optimal candidate, (7c), assigning initial main stress and penultimate secondary stress, is thus correctly selected by the algorithm.

3. Robust Interpretive Parsing

The interpretive parsing algorithm is similar in many respects to the production-directed parsing algorithm: the table for storing partial descriptions is structured exactly the same, the same categories are used, and the procedure for filling the cells of the table is similar. The difference comes with the addition of one extra restriction: when a new syllable is added to a partial description, the stress level of the new syllable must match the stress level it bears in the overt form. In other words, the candidate set being optimized over in interpretive parsing is really a subset of the candidate set used in production-directed parsing: interpretive parsing selects from among those candidates (defined by GEN) which match the overt form in the assignment of stress levels to the syllables.

Table 3 shows the full parsing table for interpretive parsing. The syllables at the top of each column now constitute the overt form, and each reflects the stress level assigned to it in the overt form: the third syllable bears main stress, and the other three syllables are unstressed.

	˘	¯	˘́	˘
NoMNoF	[˘]	[˘ ¯]		
NoMF1NoS	[(˘)]	[˘ (¯)]		
NoMF1S				
NoMF2				
MNoF				[˘ (¯ ˘́) ˘]
MF1NoS				[˘ (¯ ˘́) (˘)]
MF1S			[˘ ¯ (˘́)]	
MF2			[˘ (¯ ˘́)]	[˘ ¯ (˘́ ˘)]

Table 3: Interpretive Parsing for the Input [˘ ¯ ˘́ ˘] using all categories.

Observe that many of the cells in the table are empty. This is because many of the cells could only contain partial descriptions that would fail to match the overt form. The cell for category **NoMF1S** for the first syllable cannot be filled, because such a description would necessarily assign a secondary stress to the first syllable, contra the overt form. The same kind of reasoning explains why a majority of the cells in the table are empty.

Once the table has been completed, the algorithm must select the optimal interpretation of the overt form, from the candidates in the final column. The candidate in the row for category **MF1NoS** is excluded, as it is in production-directed parsing, because the final foot has no head. This leaves two candidates, shown with their high-ranked constraint violations in (8). The more harmonic of the two, (8a), is selected as the optimal interpretation.

(8) The two full descriptions for interpretive parsing, and their high-ranked constraint violations; candidate a is the optimal one.

		FOOTBIN	ML	PARSE	AFR	TROCH
a.	[˘ (¯ ˘́) ˘]		*	* *	*	*
b.	[˘ ¯ (˘́ ˘)]		* *!	* *		

Interpretive parsing uses the same data structure (the dynamic programming table) and basic construction operations as production-directed parsing. All that is added is an additional restriction that the stress levels match the overt form. As a consequence, interpretive parsing proceeds even more quickly, and more candidates are eliminated early due to failure to match the stress levels of the overt form (yielding many empty cells in the parsing table). Generation and comprehension can both be accomplished with the same core optimization machinery.

4. The Role of Interpretive Parsing in Language Learning

Robust interpretive parsing plays a central role in an approach to language learning proposed by Tesar and Smolensky (1996), an approach that has more recently been implemented and investigated (Tesar in press). Due to space limitations, only a brief outline of the approach will be given here; details and further discussion can be found in the works just cited.

The learning algorithm takes as input an overt form. Robust interpretive parsing is then applied to the overt form, using the learner's current constraint hierarchy. The learner then extracts the underlying form from the interpretation (recall that, for metrical stress, the underlying form is the syllables without any stress levels), and applies production-directed parsing to the underlying form, using the same constraint hierarchy. The learner compares the results of the two parsing procedures, in essence checking to see if they would pronounce the form in the same way as they just heard. If the descriptions are identical, no modification of the constraint hierarchy results; as far as the learner can tell, their grammar is fine. However, if the two do not match, then the learner will assume that the interpretation of the overt form assigned by robust interpretive parsing is the grammatical structural description, and will attempt to modify the constraint ranking in order to make the interpretation optimal.

The procedure used to modify the constraint ranking is called *constraint demotion*. Given the learner's current constraint ranking, and two descriptions, one that is intended to be optimal, called the *winner*, and a competing structural description that is currently more harmonic than the winner, called the *loser*, constraint demotion modifies the ranking so that, with respect to the new ranking, the winner is more harmonic than the loser (if this in fact possible). It does this by identifying the highest-ranked constraint violated more by the loser, and demoting all constraints violated more by the winner to just below it.

Recall the constraint hierarchy used in the illustrations of the previous sections, given again in (9).

(9) FOOTBIN » ML » PARSE » AFR » TROCH » {AFL, MR, IAMB}

Given the overt form [⏑ – ⏑́ ⏑], the learner applies robust interpretive parsing to that overt form, and production-directed parsing to the underlying form, the results being exactly as shown in the examples of the previous sections. The respective descriptions are shown in (10), with the result of production-directed parsing labeled the winner (10b), and the result of interpretive parsing labeled the loser (10a).

(10) The loser and winner before constraint demotion.

	Overt:	[⏑ – ⏑́ ⏑]	ML	PARSE	AFR	TROCH
a.	Loser	[(⏑́ –) (⏑̀ ⏑)]			* *	
b.	Winner	[⏑ (– ⏑́) ⏑]	*	* *	*	*

Constraint demotion is applied to the pair, identifying the constraint AFR as the highest-ranked constraint violated more by the loser, and demoting constraints ML and PARSE down below AFR into the stratum already occupied by TROCH (the constraint FOOTBIN, not shown in the tableau , is not violated by either candidate, and remains at the top of the hierarchy). The full resulting hierarchy is shown in (11).

(11) FOOTBIN » AFR » {TROCH, ML, PARSE} » {AFL, MR, IAMB}

Now that the constraint hierarchy has been changed, both production-directed parsing and interpretive parsing can be re-applied, using the new hierarchy. The results are shown in (12), with the loser row (12a) showing the outcome of production-directed parsing, and the winner row (12b) showing the outcome of interpretive parsing.

(12) The new loser and winner, after constraint demotion.

	Overt:	[⏑ – ⏑́ ⏑]	AFR	PARSE	ML	TROCH
a.	Loser	[⏑ – (⏑́ ⏑)]		* *	* *	
b.	Winner	[⏑ – (⏑́ ⏑)]		* *	* *	

The loser and the winner are now identical; so far as the learner can tell from this data, the new constraint hierarchy is correct. The overt form was clearly ungrammatical with respect to the previous ranking. However, because of the robustness of the interpretive parsing algorithm, the learner was able to assign a best possible interpretation to the overt form, rather than simply declaring the overt form ungrammatical and returning nothing. Notice that the result of robust interpretive parsing with the previous hierarchy, (10b), was incorrect (it has one foot stranded in the center, not aligned with either word edge), as a result of the incorrect constraint hierarchy. The best interpretation still contained enough information indicate to the learner how to modify the ranking. The application of constraint demotion to the best interpretation (as the winner) allowed the learner to end up at a constraint hierarchy for which the overt form is indeed grammatical.

5. Acknowledgments

The author would like to thank Paul Smolensky and Alan Prince for useful discussions. Beneficial interactions also took place with the participants of the Stanford workshop on Optimality Theory (December 1996), where some of this material was presented. The references with a listed ROA number can be obtained electronically, from the Rutgers Optimality Archive, at http://ruccs.rutgers.edu/roa.html.

References

Eisner, Jason. 1997. Efficient Generation in Primitive Optimality Theory, in *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics* (ROA-206).

Ellison, T. Mark. 1995. Phonological Derivation in Optimality Theory, in *Proceedings of the Fifteenth International Conference on Computational Linguistics*. 1007-1013 (ROA-75).

Frank, Robert and Giorgio Satta. Optimality Theory and the Generative Complexity of Constraint Violability. In press, *Computational Linguistics* (ROA-228).

Hammond, Michael. 1995. Syllable Parsing in English and French. Ms., University of Arizona (ROA-58).

Hammond, Michael. 1997. Parsing in OT. Ms., University of Arizona (ROA-222).

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

McCarthy, John and Alan Prince. 1993. Generalized Alignment, in G. Booij and J. van Marle, eds., *Yearbook of Morphology*, 79-154. Dordrecht: Kluwer (ROA-7).

Prince, Alan. 1990. Quantitative Consequences of Rhythmic Organization, in K. Deaton, M. Noske and M. Ziolkowski, eds., *CLS26-II: Papers from the Parasession on the Syllable in Phonetics and Phonology*, 355-398.

Prince, Alan and Paul Smolensky. 1993. Optimality Theory: Constraint interaction in generative grammar. Technical report, TR-2, Rutgers University Center for Cognitive Science, 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. 1995. Computing Optimal Forms in Optimality Theory: Basic Syllabification. Ms., University of Colorado at Boulder (ROA-52).

Tesar, Bruce. 1995. *Computational Optimality Theory*. Doctoral dissertation, University of Colorado, Boulder (ROA-90).

Tesar, Bruce. 1997. An Iterative Strategy for Learning Metrical Stress in Optimality Theory, in E. Hughes, M. Hughes and A. Greenhill, eds., *The Proceedings of the 21st Annual Boston University Conference on Language Development*, 615-626. Somerville, Mass.: Cascadilla Press (ROA-177).

Tesar, Bruce. An Iterative Strategy for Language Learning. In press, *Lingua*, special issue on conflicting constraints.

Tesar, Bruce and Paul Smolensky. 1996. Learnability in Optimality Theory (long version). Technical Report JHU-CogSci-96-4, Department of Cognitive Science, the Johns Hopkins University (ROA-156).

Tesar, Bruce and Paul Smolensky. 1998. Learnability in Optimality Theory. *Linguistic Inquiry*, Spring.