This article appears as chapter 10 of What is Cognitive Science?, edited by Ernest Lepore and Zenon Pylyshyn, Blackwell, 1999.

Linguistic and Cognitive Explanation in Optimality Theory

Bruce Tesar
Jane Grimshaw
Alan Prince

Generative linguistics aims to provide an analysis of the grammar-forming capacity that individuals bring to the task of learning their native language (Chomsky 1965, 1981, 1991, 1995). Pursuing this goal amounts to developing a linguistic theory that achieves maximum universality and generality in its premises, while at the same time offering explicit, limited means for representing possible interlinguistic variation. The term "Universal Grammar" is used to refer to the system of principles defining what the grammar of a human language can be.

Optimality Theory (Prince and Smolensky 1993) asserts that Universal Grammar provides a set of general, universal constraints which evaluate possible structural descriptions of linguistic objects. These constraints are assumed to be strongly universal, in the sense that they are present in every grammar; they must be simple and general if they are to have any hope of universality. The structural description that is *grammatical* for a linguistic object in a given language is the one, among all possible structures assignable to that object, which is *optimal*, in the sense that it best satisfies the universal constraints, given the defining characteristics of that object. The theory builds from a notion of "best satisfaction" — optimality rather than perfection — because constraints are often in conflict over the well-formedness of a given candidate analysis, so that satisfying one constraint entails violating others to varying degrees. Indeed, an optimal structural description will typically violate some (or many) of the constraints, because no possible description satisfies all of them.

Differences between languages then emerge as the different ways allowed by the theory for resolving the conflicts that are inherent in the universal constraint set. The theory is therefore one of constraint *interaction*: the effects of any constraint are determined both by its intrinsic demands and by its relation to the other constraints; fixing the relations between the universal constraints defines the grammar of a particular language. Interactionism is the heart and soul of the theory: it places a heavy empirical burden on the positing of constraints, since all interactions must yield possible grammars; it leads to a pattern of explanation in which many universally observed properties of language follow not from hypothesized special principles designed to encode them directly, but from nothing more than the interaction of more general constraints which are not specifically concerned with those particular properties; it opens the way to tremendous simplification of the constraints themselves, since they need not contain codicils and complications that emerge from interaction; and, at the methodological level, it enforces the research ethic that posited constraints *must not* contain such codicils and complications, thereby guiding the search for a general understanding of the nature of the constraints active in human language.

This essay presents an overview of selected work making use of Optimality Theory, with the goal of illuminating these general ideas. Section 1 presents and illustrates the central principles of Optimality Theory. Section 2 gives an example of syntax within the theory. Section 3 examines possible implications of Optimality Theory for studies of language processing, discussing work on the computability of Optimality Theoretic grammars, as well as some conceptual similarities between Optimality Theory and work in connectionism and dynamical systems. Section 4 discusses work on language learnability and acquisition within Optimality Theory.

A useful resource is the Rutgers Optimality Archive (ROA). This is an electronic repository of papers on Optimality Theory, and is accessible on the Internet. The World Wide Web URL is http://ruccs.rutgers.edu/roa.html.

1 The Principles of the Theory

A *grammar* is a formal specification of the structure of a language; cognitively construed, it characterizes a speaker's internalized unconscious grasp of the language. In generative grammar, the structure of a linguistic object such as a sentence is given as a set of representations that explicate meaning, sound, and syntactic form. The grammar defines how such representations can go together to form a legitimate linguistic entity. Grammars are typically organized so as to map from one representation to another, from an 'input' to an 'output': from the lexical representation of a word (input) to its pronunciation (output), from an argument structure (input) to a syntactic structure embodying it (output), and so on. Different linguistic theories within generative grammar put forth different assumptions about what the relevant representations are, and how the mapping between them is accomplished.

A grammar, then, specifies a *function* which assigns to each type of linguistic input an output structural description (or, possibly, a set of such). The grammar itself does not provide an *algorithm* for effectively computing this function. The distinction between function and algorithm is worth emphasizing, because of the ambiguous role of computational procedures in some linguistic theories. For example, early generative phonology defines grammars in terms of serial derivation — a sequence of procedural steps, each a 'phonological rule'; similarly, many theories of syntax have followed this model. Is the derivation only a means for defining the grammatical function, or is it additionally a theory of how language processing is actually conducted in the mind of a language user? Different authors have taken different positions; as have the same authors at different times. Optimality Theory forces the issue by giving an explicitly non-procedural definition of the grammar. How the functions defined by Optimality Theoretic grammars might be explicitly computed is discussed in Section 4.

1.1 Constraints and Their Violation

To see how grammars are organized to represent linguistic generalizations, let us consider the case of syllabification. In every language, when words are pronounced, they divide into syllables, which group consonants and vowels together. Syllable structure typically has important further consequences for pronunciation, determining numerous features of articulation, including tone and stress placement. In this domain of linguistic patterning, there are two basic facts to explain. First, syllabification is predictable, given the grammar of the language and knowledge of the sounds of which a word is composed: in English, for example, "Memphis" is pronounced *mem.phis*, not **me.mphis* or **memph.is*. (Periods indicate syllable-divisions; the notation *X indicates that the form X is ill-formed.) Second, languages place various limits on what syllables are admissible. English is fairly free in this regard, though there are languages that are freer (for example, Mtsensk is a legitimate single-syllable word in Russian, though not in English, even though all the individual sounds are shared by the two languages; and "Mom" is a monosyllable in English, but not Japanese though Japanese has all the requisite sounds). Thus, the syllabic grammar of a given language must not only predict how the various sounds of a word are grouped into syllables, it must also tell us that some groupings are simply not allowed.

Since syllable structure is predictable in each language, we can assume that the input represents only consonants and vowels ('segments') and not syllables. The mapping from input to output supplies the syllables, predicting their composition. Suppose the grammar faces an input like /*apot*/. There are many possible outputs that it could give rise to: some which simply parse the input differently, others that more aggressively insert or delete material, perhaps to facilitate the creation of syllables. Table 1 lists a few alternatives:

Analyses of /apot/		*Remarks*
a	*.a.pot.*	*bisyllabic:* first syllable open
b	*.ap.ot.*	*bisyllabic:* first syllable closed
c	*.a.po.*	*bisyllabic:* with deletion of *t*
d	*.pot.*	*monosyllabic:* via deletion of *a*
e	*.po.*	*monosyllabic:* via deletion of both *a* and *t*
f	*.ʔa.pot.*	*bisyllabic:* with insertion of *ʔ*
g	*.a.po.ti*	*trisyllabic:* with insertion of *i*
h	*.ʔa.po.ti.*	*trisyllabic:* with insertion of both *ʔ* and *i*
...	...	*...and many, many others...*

Table 1 A few candidate analyses of the input /*apot*/

It is the job of a language's grammar to determine which candidate output analysis is the actual one, for a given language. Analysis (a) would be chosen in a language like English; analysis (f) in a language like German or Arabic; Japanese would choose an analysis something like (h), though with a different inserted vowel. Although option (b) simply exploits locally permissible structures, it is not found: a fact that will be explained below.

The same considerations arise in the theory of syntactic structure. Consider the analysis of a question like 'What will John eat?'. The input must specify the presence of the words, with their semantic and syntactic properties: the verb *eat* is two-place predicate; *John* is a noun, *will* an auxiliary, *what* a pronoun and an operator. (It is, of course, the categories that are important, not the particular words, for languages share categories rather than words.) For concreteness and simplicity, we will present the structures as word-strings here; below, we expand on the relevant structural analyses. Even limiting ourselves to those candidate outputs that observe the basic patterns of English phrase structure, quite a number of alternatives present themselves, as shown in table 2.

Candidate Analyses		*Remarks*
a	John will eat what	Question operator in canonical object position
b	*will* John eat what	Auxiliary placed in fronted position
c	*what* John will eat	Q-operator fronted
d	*what will* John eat	Aux and Q-operator fronted, Q first
e	*what do* John will eat	Q-operator fronted, supporting aux inserted
...	...	*...many, many others...*

Table 2 A few syntactic candidates

All of these reflect real structural possibilities, whose components at least are directly observed in many natural languages. English chooses (d); many languages choose (c), and indeed English uses this when the question is syntactically subordinated ('I know *what John will eat*'). Option (a) is also common; structural options (b) and (e), though composed of independently observable sub-structures, are universally impossible: a fact to be explained.

We assume that every language has the ability to choose from the entire collection of possible candidate analyses. Universal Grammar — the set of principles defining the way grammars can be constructed — must be organized to support this choice by making all the alternatives available to particular grammars. We therefore have our first principle of grammatical organization:

Principle 1: *Universal Grammar provides a function, Gen, which maps each input to its set of candidate structural descriptions.*

A linguistic input α thus universally gives rise to the same set of candidate outputs, Gen(α), in every language. The domain of the function Gen defines the set of possible inputs; its range is the set of possible outputs. In table 1 we have listed a fragment of Gen(*apot*); in table 2, some of the members of Gen(*V(x,y); x=DP, y=wh, aux=will*), to anticipate the notation of section 3. Specification of the function Gen involves detailing the set of formal primitives linguistic structure and their basic, ineluctable modes of combination: linguistic representations are constructed by virtue of these, both as inputs and outputs. Principle 1 asserts that the description assigned by the grammar of any particular language must be drawn from the set of candidates provided by Gen.

How then is the choice among the candidates to be accomplished? Optimality Theory claims that this is to done by a set of constraints on structural configurations, and on the fidelity of input to output: these constraints are also posited to universal.

Principle 2: *Universal Grammar provides Con, a set of universal constraints assessing the well-formedness of structural descriptions.*

The constraints of Con assess the well-formedness of all candidate descriptions for a given input. Such evaluation works *in parallel*: that is, the constraints evaluate each candidate independently; so that the assessment of one candidate does not depend upon the assessment of another candidate.

Table 3 shows a set of constraints relevant to the assignment of syllable structure.

Name	*Content*
ONSET	A syllable must begin with a consonant: must have an *onset*.
NOCODA	A syllable must not end on consonant: must lack a *coda*.
NODEL	Input segments must be appear in the output: must not be *deleted*.
NOINSV	The output must not contain an *inserted* vowel: one not present in the input.
NOINSC	The output must not contain an *inserted* consonant: one not present in the input.

Table 3 Some basic universal constraints relevant to syllable theory

There is a fundamental formal division among constraints: ONSET and NOCODA examine only the character of the output: they are *structural constraints*; the others (below the double line) compare input with output, and in each case require that the output preserve some feature of the input: these are *faithfulness constraints*.

Given an input and a candidate description of it, we can determine the extent to which each of the constraints is violated. The following table records this information for the candidates from Gen(*apot*) cited above.

	/*apot*/	ONSET	NOCODA	NODEL	NOINSV	NOINSC
a	*.a.pot.*	*	*			
b	*.ap.ot.*	**	*			
c	*.a.po.*	*		*		
d	*.pot.*		*	*		
e	*.po.*			**		
f	*.ʔa.pot.*		*			*
g	*.a.po.ti*	*			*	
h	*.ʔa.po.ti.*				*	*

Table 4 Evaluation of the Candidate Set

Each violation is notated with an asterisk; lack of violation is not indicated. Multiple violations are recorded as multiple asterisks.

Observe that the fully faithful parse (a) *.a.pot.* does quite poorly on the structural constraints ONSET and NOCODA, violating both of them. Complete success on the structural constraints can be achieved, as in (e) *.po.* and in (h) *.ʔa.po.ti.*, but the cost is failure, even multiple failure, on one or both of the faithfulness constraints. A range of intermediate successes and failures can be found in the rest of the table.

We can already see why candidate (b) *.ap.ot.* is essentially doomed. Although it is locally composed of licit syllables, it fares the same as or worse than candidate (a) on *every* constraint. It is hard to imagine an evaluation system that would evaluate it as the winner over (a) in these circumstances.

Relations between the other competitors are not so straightforward: there is outright conflict on the question of which member of various pairs of candidates should be judged more successful. Is (a) *.a.pot.* better than (h) *.ʔa.po.ti.* ? Candidate (a) is structurally inferior to (h), but (h) is less faithful to the input. Other similar forms of conflict are rife throughout the table. Is it better to lack an onset (*ONSET) or to have an inserted consonant (*NOINSC)? Is it better to insert a vowel (*NOINSV) or to delete a consonant (*NODEL)? To insert a vowel or to have a coda? The answers to such questions will determine what the grammar generates.

Exactly the same situation arises in syntax. Table 5 shows how the candidates of table 2 fare with respect to a set of syntactic constraints. The meaning of the syntactic constraints will be discussed in section 3 below; what is significant here is the clear formal parallel with constraint evaluation in phonology.

	V=(x,y); x=John, y=what, aux=will	OPSPEC	OBHD	NOINS/LEX	NOMVT
a	John will eat what	*			
b	*will* John eat what	*			*
c	*what* John will eat		*		*
d	*what will* John eat				**
e	*what do* John will eat			*	*

Table 5 Evaluation of syntactic candidates

We can immediately see why any candidate formed like (b) will never surface in any language: it fares the same as or worse than (a) on every constraint. Relations elsewhere are more complicated. Is it better to have an operator out of position, as in (a), or is it better to have a defective ('headless') syntactic constituent, as in (c)? Is it better to insert the dummy element *do*, violating NOINS as in (e), or to displace the auxiliary *will* from its canonical position, violating NOMVT as in (d)? A grammar must be able to decide these issues.

1.2 Optimality and Harmonic Ordering

The constraints of Con provide the basis for selecting the correct output from among the candidate set of alternative analyses. Yet if the constraints are to be simple and universal, they must be violable and there is sure to be conflict among them, as we have seen. We must therefore seek to define what it means to "best satisfy" the constraint set, presupposing that there will be much constraint violation in grammatical forms.

To fulfill the goal of articulating a Universal Grammar that allows only the licit grammars of human languages, we seek a scheme of conflict resolution that is as restrictive as possible, allowing just the variation between languages that is empirically observed. Merely summing up violations across the entire constraint set cannot work: as the above examples show, it cannot select a single winning candidate, given the kind of constraints proposed there; worse, it cannot allow the expression of interlinguistic variation while maintaining the universality of the constraint set Con: there could be only one grammar. At the opposite pole of descriptive richness, allowing free numerical weighting opens up a Pandora's box of logical possibilities, with little hope for significant restrictions on the grammars that are generable.

Optimality Theory therefore posits a single, symbolically-based mechanism of conflict resolution: prioritization. In case of conflict between two constraints, one is specified as taking absolute priority over the other: the one constraint "strictly dominates" the other. In case of conflict, the interests of the dominant constraint must be served, at the cost of as much violation of the subordinate constraint as is necessary.

Principle 3: *A grammar is a total ranking of the constraints of Con into a strict domination hierarchy.*

A form which best satisfies the strict domination hierarchy is the *output* for a given *input*; it is said to be *optimal*. What does this mean for the choice among competing candidates?

Suppose ONSET strictly dominates NOINSC, which we notate as ONSET ≫ NOINSC. Then the requirement that syllables begin with consonants takes unqualified precedence over the prohibition against inserted consonants. Therefore, any number of consonants can be inserted to ensure that the output's syllables all have onsets.

Consider an input /*aopu*/. Let us assume that other dominant constraints limit the insertable consonant to the glottal stop [ʔ] and limit syllables to containing just one vowel, forcing a trisyllabic parse. Then we have the following competition shown in table 6 between the faithful parse and various alternatives with insertion.

/*aopu*/	ONSET	NOINSC
a .*a.o.pu.*	* *	
b .*ʔa.o.pu.*	*	*
c .*a.ʔo.pu.*	*	*
☞ d .*ʔa.ʔo.pu.*		* *

Table 6 ONSET ≫ NOINSC

Notation: a table of this sort, with the constraints arrayed in domination order, is called a *tableau*: the optimal form is marked with the sign ☞.

Candidate (d) does best on the dominant constraint ONSET; anything that does worse is eliminated. In order to achieve success on ONSET, the winner tolerates more violations of NOINSC than any other candidate in this set.

To complete the selection of (d), a wider candidate set must be examined. In particular, success on ONSET can also be achieved by deletion, leading to conflict among faithfulness constraints, as shown in table 7.

/*aopu*/	ONSET	NODEL	NOINSC
☞ d .*ʔa.ʔo.pu.*			* *
e .*pu.*		* *	

Table 7 NODEL ≫ NOINSC

These arguments show that if the C-insertion solution to the onset problem is to be taken in a grammar, it must conform to two distinct ranking conditions:

ONSET ≫ NOINSC
NODEL ≫ NOINSC

Any total ranking of the constraint set that meets these conditions will suffice.

Permuting the crucial ranking relations leads directly to the selection of a different winner. If we invert the ranking of the two faithfulness constraints, so that NOINSC ≫ NODEL, the deletion-derived candidate (e) becomes optimal. If we subordinate ONSET to *both* faithfulness constraints, then neither deletion nor insertion is allowed to modify the input, and we must live with *.a.o.pu.* as the optimal output, despite its two onsetless syllables.

The calculation of optimality rests, at bottom, on competitions between pairs of candidates. Imagine two competitors α and β. Any given constraint C may either distinguish them in terms of degree of violation of C, or evaluate them as equivalent, if they both violate C exactly the same number of times. Faced with a constraint hierarchy H, we say that form α is *better* than, or 'more harmonic than' form β, if α fares better than β *on the highest-ranked constraint that distinguishes them*. Table 8 illustrates a typical competition. Assume that the order of constraints across the top of the tableau reflects a strict domination ranking.

/*apot*/	ONSET	NOCODA	NODEL	NOINSC
☞ f *.ʔa.pot.*		*		*
d *.pot.*		*	*!	

Table 8 A typical competition

Candidate (f) is more harmonic than candidate (d), because it fares better on NODEL, the first constraint distinguishing them: the crucial violation is marked with an exclamation point. Although (f) violates NOCODA, this is of no consequence to the comparison, because (d) is just as bad on that constraint. Observe that (f) involves insertion and is therefore *worse* than (d) on NOINSC; but this does not affect the calculation either, since the competition is concluded at the level of NODEL in the ranking, above the ranking level where NOINSC has its effect.

An *optimal form* ω stands at the top of the competitive heap: it never loses any of these pairwise comparisons. If there is only one optimal form, then it is better than all other candidates: it wins against every other candidate.

This discussion has proceeded from the following observation:

> A grammar's constraint hierarchy induces a harmonic ordering of all the candidate descriptions for an input.

The harmonic ordering, essentially a lexicographic order on the candidate set, determines the relative "harmony" of every pair of candidates. An optimal form is a maximal element in this ordering.

1.3 Constraint Ranking and Linguistic Typology

Principle 4: *Cross-linguistic variation is explained by variation in the ranking of the universal constraints.*

Analysis of the optimal descriptions arising from all possible rankings of the constraints provided by Universal Grammar gives the typology of possible human languages. This can be illustrated by considering a different constraint ranking for the constraints of Basic CV Syllable Theory. Table 9 gives the tableau for input /*apot*/ using the ranking in (1).

(1) ONSET ≫ NOINSV ≫ NODEL ≫ NOINSC ≫ NOCODA

/*apot*/	ONSET	NOINSV	NODEL	NOINSC	NOCODA
a *.a.pot.*	*				*
b *.ap.ot.*	* *				* *
c *.a.po.*	*		*		
d *.pot.*			*		*
e *.po.*			* *		
☞ f *.ʔa.pot.*				*	*
g *.a.po.ti*	*	*			
h *.ʔa.po.ti.*		*		*	

Table 9 Constraint tableau for /*apot*/ with ranking (1)

In this language, codas are possible: they are permitted when necessary to avoid violation of the faithfulness constraints, NOINSV, NODEL, NOINSC, as with input /*apot*/. Note that codas are not required, however. Consider the input /*po*/: the candidate description *.po.* does not have a coda, but is the optimal description. In fact, candidate *.po.* violates none of the constraints, and therefore is the optimal candidate for /*po*/ under any ranking. Therefore, this subsystem predicts this input-output relation in all languages, modulo other constraints.

Analysis of all possible rankings of the Basic CV Syllable Theory constraints reveals that the resulting typology of basic CV syllable structures instantiates and indeed goes beyond Jakobson's fundamental cross-linguistic generalizations on patterns of syllabification (Jakobson 1962, Clements and Keyser 1983; see Prince and Smolensky 1993: ch. 6 for full discussion). In this typology, a language may require onsets, or may merely allow them; independently, a language may forbid codas, or allow them. No grammar produced by ranking these constraints can *ban* onsets or *require* codas; this is guaranteed by the optimality of the mapping /*po*/↦ *.po*. under all rankings. This result provides us with an example of how significant generalizations about linguistic structure, like Jakobson's, can emerge from the range of permissible interactions in a constraint set. The theory allows every ranking — a natural null hypothesis — yet it is far from true that every *outcome* is thereby allowed; and the patterns of permitted outcomes predict the structure of linguistic variation.

To support a claim that a particular grammar does not permit some particular type of structure, it must be demonstrated that for any possible input, the optimal description assigned by the grammar does not include the structure in question. This is due to a principle known as richness of the base.

Principle 5: *Richness of the Base: the set of possible inputs is the same for all languages.*

The inventory of a grammar is defined as the types of structures that appear in the set of descriptions arising from all possible inputs. The lexicon of a language is a sample from the inventory of possible inputs.

Principle 6: *The universal constraints include faithfulness constraints, which are violated by discrepancies between the surface form of a description and the input.*

Faithfulness is an essential part of Optimality Theory. For example, NODEL is violated by 'underparsing', when elements from the input fail to appear in the output; and the NOINS constraints are violated by 'overparsing', when material that is not precedented in the input makes it appearance in the output. The presence of faithfulness constraints means that disparities between the input and the output form (both being essential components in a structural description) will only be tolerated in order to satisfy other, higher-ranked, constraints. This is why the same grammar assigns different grammatical descriptions to different inputs.

1.4 Linguistic Markedness

The concept of linguistic markedness, or inherent complexity of structures, has played a significant role in linguistic thought in the twentieth century, though more often as an unformalized perspective or side-issue within generative grammar. Optimality Theory rests directly on a theory of linguistic markedness: 'marked' or linguistically-complex structures are literally marked as such by the constraints they violate. This can be illustrated by the effects of low-ranked constraints. Consider the language described above. The constraints NOINSC and NOCODA are the lowest-ranked constraints. This does not mean, however, that they play no role in the language. For the input /*apot*/, the optimal candidate, .*ʔa.pot.*, violates only NOINSC and NOCODA, and has one coda. Another candidate for the same input, .*ʔap.ʔot.*, has two codas. It also only incurs violation of NOINSC and NOCODA, but to a greater extent (an additional violation of each), rendering it suboptimal. Codas are still avoided when possible. The presence of NOCODA in Con entails the presence of NOCODA in the rankings for all languages. Even in languages that permit codas, they are permitted *only when necessary*. Optimality Theory provides a precise definition of "when necessary": structures marked by a constraint are permitted only when all candidates avoiding that structure contain other structures marked by higher-ranked constraints.

This also illustrates a key conceptual difference between constraint rankings in Optimality Theory and parameter settings in the Principles and Parameters framework (Chomsky 1981). One could imagine a simple parameter set giving rise to the basic inventories of the Jakobson typology. One parameter, OnsetP, would have two settings, an unmarked setting requiring syllables to have onsets, and a marked setting allowing syllables with and without onsets. Another parameter, CodaP, would have an unmarked setting forbidding codas, and a marked setting allowing syllables with and without codas. What the parameter CodaP fails to capture is any sense that codas are marked structures even within languages that permitting coda. Once it is set to the marked setting, CodaP is mute concerning where and when codas will actually appear. By contrast, an Optimality Theoretic constraint still has the potential to play a crucial role in analyses when low-ranked. Placing both ONSET and NOCODA at the bottom of the ranking will not stop /*poto*/ from being assigned the description .*po.to.* (as opposed to, say, .*pot.o.*). The same constraints explaining the cross-linguistic markedness of structures also explain the markedness of structures within languages permitting those marked structures.

The basic notion of marked structure is directly built into the theory: a marked structure is one receiving a violation mark by a constraint in Con. The distribution across environments of marked structures in a language is not directly built into the theory, but follows as a consequence of the constraint ranking for that language. Implications about the cross-linguistic distribution of marked and unmarked structures (of the form "if a language allows structures marked along a certain dimension, it allows structures less marked along that dimension") again are not built into the theory, but follow from the properties of the entire set of possible constraint rankings.

1.5 Work in Phonology

Work in phonology in Optimality Theory is too extensive to quote without raising the specter of tendentiousness. Interested readers should consult the Rutgers Optimality Archive (http:/ruccs.rutgers.edu/roa.html), which contains many works voluntarily posted and freely available to the research community, as well as an extensive bibliography (current only to June, 1996, however). Here we cite some works whose authorship overlaps with that of this paper. Prince and Smolensky 1993 is the foundational work, laying out the theory and exploring it in the context of empirical issues in phonology, including those just discussed. McCarthy and Prince 1993b develops an approach to prosodic morphology based on Optimality Theory and introduces ideas about correspondence of representations which are developed into a general formal reconstrual of faithfulness in McCarthy and Prince 1995. McCarthy and Prince pursue a variety of issues in prosodic morphology in such papers (McCarthy and Prince 1993b, 1994, 1995), including accounts of reduplicative phenomena where the base of reduplication actually changes to be more like the reduplicant, defying standard derivational copying accounts. Work on generalized alignment (McCarthy and Prince 1993a) discusses an important type of constraint, the alignment constraint, which plays a role in defining structural relations throughout phonology, morphology, and syntax.

2 A Syntax Example: English Interrogative Inversion and Do Support

The essential properties of English interrogatives and *do* support are derived from the interactions of constraints on structure and faithfulness. The two properties we will focus on here are these: that "inversion" of an auxiliary verb and the subject occurs in interrogatives but not in declaratives (see 2a,b), and that in the absence of another auxiliary verb *do* appears, except when it is the subject that is being questioned. (unless the *do* is stressed, a case we set aside) (see 3a,b).

(2) a. **He will** read this book. ***Will he** read this book
    b. Which book **will he** read? *Which book **he will** read?

(3) a. Which book **did he** read? *Which book **he read**?
    b. **Which boy read** the book? ***Which boy did** read the book?

The account presented here is a simplified version of the analysis in Grimshaw (1997b), which we have modified slightly to accentuate the fundamental similarities between the phonological theory of syllable structure and the syntactic theory of phrase structure. (Related accounts of interrogatives and similar phenomena can be found in Ackema and Neeleman (1998), Legendre, Wilson, Smolensky, Homer, and Raymond (1995), Legendre, Smolensky, and Wilson (1998), Müller (1997), Pesetsky (1998), and Sells, Rickford, and Wasow (1994).

An input consists of a lexical verb (V) along with its argument structure, the verb's arguments (such as *the boy* and *which book*), and the tense and meaningful auxiliary verbs which encode future tense and aspect. Thus the lexical items in the input are like segments in the phonological input. The candidate analyses are the various possible ways of organizing these lexical items into a phrase structure tree. As the phonological constraints determine the organization of segments into syllables, so the syntactic constraints force certain structural representations of the inputs.

Constraints not discussed here give us some background assumptions: we consider only candidates which are consistent with the basic theory of phrase structure as instantiated in English syntax, and only candidates in which the subject of the verb is in a specifier position, preceding the head auxiliary or main verb when one is present. (See Samek-Lodovici 1996, Grimshaw and Samek-Lodovici 1995, 1998, for an analysis of subjects within OT assumptions). Every "projection" or sub-piece of structure has this basic form, where any of the specifier (always a phrase), head (always a single word) and complement may be missing in a given case:

XP

specifier head complement

The constraints are given in table 10. These constraints are of two types, exactly as for the phonological constraints discussed above. One kind of constraint (above the double line in table 10) bans certain types of marked structures: projections without heads, operators outside specifier positions, chains, and chains with lexical heads.[1] The second kind of constraint enforces faithfulness to the input, in particular NoIns/Lex prohibits the insertion of lexical items not in the input.

Thus we see the same fundamental division into structural and faithfulness constraints that appears in the phonological constraints in table 3. The ranking of the constraints for English is given in(4). A *wh* word is a syntactic operator, and thus required by OpSpec to be in a specifier position, where it can take scope. This pressure to move *wh* words into specifier position is the primary cause of activity in this analysis, and the phenomena of inversion and *do* support are explained by the interaction between OpSpec and the other constraints.

Name	*Content*
ObHd	A projection must have a head
OpSpec	An operator must be in a specifier position
NoMvt	The output must not contain a chain
NoMvt/Lex	The output must not contain a chain with a lexical head
NoIns/Lex	The output must not contain an *inserted* lexical item

Table 10 Basic universal syntactic constraints relevant for inversion theory

(4) The Constraint Hierarchy for English (*simplified: other rankings are motivated within the full system*)

OpSpec, NoMvt/Lex, ObHd $\gg$ NoIns/Lex $\gg$ NoMvt

[1] A chain is a set of positions in a clause, such that one element, the head, effectively fills them all. It is the representational characterization corresponding to the notion of "movement".

When there is no operator in the input, as in a declarative, the optimal candidate violates none of these constraints. Table 11 shows a few of the candidates for a declarative sentence having a main verb, an auxiliary verb, and subject and object DPs, such as (2a) above. The auxiliary *will* is included in the input, marking future tense. (NOMVT/LEX is not relevant to the evaluation of these candidates and is omitted from the tableau.)

The key point here is that the optimal candidate, 11a, has minimal structure. The representation which includes just two projections can satisfy all constraints: OPSPEC is satisfied vacuously, no chains are formed to violate either of the NOMVT constraints. Every projection has a head, and no material is inserted. It is clear that all other structural arrangements of the material can only add violations, with no benefit, since the optimal candidate is already as good as it could be.

V = (x,y); x = DP, y = DP, aux = *will*	OPSPEC	OBHD	NOINS/LEX	NOMVT
☞ a $[_{IP}$ DP *will* $[_{VP}$ V DP]]				
b $[_{CP}$ $[_{IP}$ DP *will* $[_{VP}$ V DP]]]		*!		
c $[_{CP}$ *will*$_i$ $[_{IP}$ DP t_i $[_{VP}$ V DP]]]				*!
d $[_{CP}$ *do* $[_{IP}$ DP *will* $[_{VP}$ V DP]]]			*!	

Table 11 English Declaratives

The relevant modifications of the analysis in 11a involve adding an extra projection, or indeed multiple extra projections, although this is not illustrated. Regardless of how the additional projection is constructed, one of the structural or faithfulness constraints is necessarily violated. 11b fatally violates the structural OBHD. There are two strategies available for avoiding the OBHD violation; namely forming a chain to fill the empty head as in 11c, and inserting *do* as in 11d. The chain option violates NOMVT, and the insertion option violates NOINS/LEX. We can conclude that functional projections will only occur in optimal candidates in response to constraints.

This provides the answer to part of the question posed above: why there is no inversion in declaratives. In fact, with just these constraints, the theory predicts that no language could have inversion in a declarative, since the inversion candidate is harmonically bound by the optimal one. However, additional constraints are at work in linguistic systems, with the result that verb-subject order is cross-linguistically possible.

The true richness of the system becomes apparent when we consider a more challenging input. When the input contains a *wh* word (such as *which*, a syntactic operator), OPSPEC becomes relevant, exerting pressure to move *wh* words into specifier position. If a *wh* word fills an argument role normally realized in complement position, then the *wh* word may "move" to a specifier position only by forming a chain with a co-indexed trace in the argument role position. Such a chain violates NOMVT: thus OPSPEC and NOMVT conflict. The fact that the *wh* phrase appears in specifier position in the optimal candidate, 12d, establishes the ranking OPSPEC $\gg$ NOMVT. The candidates 12a,b, which leave the *wh* phrase in the object position, are thus eliminated.

Now, however, it is clear that the situation is quite different from that faced in declaratives. In an interrogative the optimal candidate *must* contain an extra projection, in order to provide the specifier position to house the operator. As noted in the context of the declarative, an extra projection poses a grammatical challenge: what to do with the head position? OBHD conflicts with both NOMVT and NOINS/LEX. Forming a chain can redeem a violation of OBHD but at the cost of a NOMVT violation. NOINS/LEX bars filling a head position with inserted material and hence conflicts with OBHD,. Thus the three choices for the head position; leaving it empty, filling it with inserted material and having it be part of a chain, each violate a constraint. The grammar must decide which of the constraints will be violated in this configuration.

Since the optimal candidate, 12d, has the auxiliary verb heading a chain in violation of NOMVT but satisfying OBHD, we conclude that OBHD ≫ NOMVT. Why doesn't the language choose to avoid violating OBHD with the other strategy, namely insertion? This is because NOINS/LEX dominates NOMVT, so the ban against chains has lower priority than the ban against inserting material that is not in the input.

V = (x,y); x = DP, y = wh, aux = *will*		OPSPEC	OBHD	NOINS/LEX	NOMVT
a	$[_{IP}$ DP *will* $[_{VP}$ V *wh*]]	*!			
b	$[_{CP}$ *will*$_i$ $[_{IP}$ DP t_i $[_{VP}$ V *wh*]]]	*!			*
c	$[_{CP}$ *wh*$_j$ $[_{IP}$ DP *will* $[_{VP}$ V t_j]]]		*!		*
☞ d	$[_{CP}$ *wh*$_j$ *will*$_i$ $[_{IP}$ DP t_i $[_{VP}$ V t_j]]]				* *
e	$[_{CP}$ *wh*$_j$ *do* $[_{IP}$ DP *will* $[_{VP}$ V t_j]]]			*!	*

Table 12 English interrogatives with an auxiliary

We have just seen that English prefers the marked structure involving a chain to the even more marked structure involving epenthesis of a verb. This does not mean, however, that the insertion of a verb is never possible in the language. In fact it is the solution of choice for a different input. Table 13 shows candidates for the same input, except that the tense is past instead of future, hence there is no auxiliary verb in the input. All the candidates shown satisfy OPSPEC. The problem to be solved here is this: the "extra" projection is required, and OBHD mandates that its head be filled. The options for filling the head position for this input, however, are not quite the same as before. In particular, there is no auxiliary verb in the input which can be moved into the empty position. The only verb available is the main verb, but a main verb is lexical and if this verb inverts the constraint NOMVT/LEX is violated. This constraint makes chains with lexical heads (a main verb in this case) more marked than chains headed by non-lexical material (such as an auxiliary verb). Any movement violates NOMVT; lexical movement additionally violates NOMVT/LEX.

It is in this circumstance that insertion is the best strategy and *do* surfaces, as (3a) above. Candidate 13c has *do* inserted as an auxiliary, which forms a chain filling both the head of CP and the head of IP. This is the optimal candidate: it violates NOINS/LEX once and NOMVT twice, but satisfies the other three (higher-ranked) constraints. The inversion of *do* is required; without the inversion, the insertion of *do* is simply a gratuitous violation of NOINS/LEX, as shown by 13d. It is the fact that NOMVT/LEX dominates NOINS/LEX that determines the outcome here: the opposite ranking gives a language with no "*do*-support" and in which a main verb can raise to fill an empty head. French exemplifies such a system.

V = (x,y); x = DP, y = *wh*, tense = past		OPSPEC	NOMVT/LEX	OBHD	NOINS/LEX	NOMVT
a	$[_{IP}\ wh_j\ [_{VP}\ V\ t_j]]$			*!		*
b	$[_{CP}\ wh_j\ V_i\ [_{VP}\ t_i\ t_j]]$		*!			* *
☞ c	$[_{CP}\ wh_j\ do_i\ [_{IP}\ DP\ t_i\ [_{VP}\ V\ t_j]]]$				*	* *
d	$[_{CP}\ wh_j\ [_{IP}\ DP\ do\ [_{VP}\ V\ t_j]]]$			*!	*	*

Table 13 English interrogatives with *do*

The verb *do* is admitted into interrogatives of the kind just discussed because it solves a problem: filling an empty head position. We have already seen that when there is a better solution for the problem, namely when an auxiliary verb from the input is available, *do* is impossible. It is also impossible when the problem doesn't arise; in configurations with no empty head. This is the explanation for the case illustrated above in (3b) where *do* does not occur in an interrogative, despite the lack of any other auxiliary. This happens where the *wh* operator is the subject, shown in table 14. Because the subject is for independent reasons in a specifier position, OPSPEC is satisfied with no movement. Thus, as in the declaratives, there is no motivation for inversion or *do* support, as demonstrated by the suboptimality of 14b and 14c.

V = (x,y); x = *wh*, y = DP, tense = past		OPSPEC	NOMVT/LEX	OBHD	NOINS/LEX	NOMVT
☞ a	$[_{VP}\ wh\ V\ DP]]$					
b	$[_{IP}\ wh_j\ [_{VP}\ t_i\ V\ DP]]$			*!		*
c	$[_{IP}\ wh_j\ do\ [_{VP}\ V\ t_j]]]$				*!	*

Table 14 English interrogatives with a *wh* subject

Thus, with a combination of constraints which ban marked structures and constraints which enforce faithfulness to the input, we can derive the fundamental properties of the English interrogative inversion system. The constraints evaluate all the alternative organizations of the input into a syntactic structure, and select the optimal one under a given ranking.

This analysis illustrates how the Optimality Theory framework naturally captures the intuitive notion of economy that has received much attention recently (e.g. Chomsky 1991, 1995). Chains are not prohibited absolutely, but are tolerated only to the extent necessary; the most economical option is selected, relative to other constraints. The universal constraint NOMVT is active in all languages, prohibiting unmotivated chains. This is quite apparent even in English, where NOMVT is the lowest-ranked constraint of those discussed. Resources like functional projections, chains, and insertion are used only when necessary, and only as much as necessary. Optimality Theory gives a precise characterization of "only when necessary"; violation is permitted only when all alternatives either violate a higher ranked constraint or incur greater violation of the same constraint.

The many recent studies carried out under Optimality-theoretic assumptions cover a much wider range of empirical issues than can be even touched on here. Interesting examples include aspects of pronominal systems (Grimshaw 1997a; Bresnan, to appear), discourse related constraints (Choi 1996; Samek-Lodovici 1998), and null pronouns (Speas 1997). Further work on the syntactic constraint system promises to elucidate further the similarities between phonological and syntactic representations under OT.

3 Language Processing

Central to language processing is the assigning of structural descriptions to linguistic material via algorithms. Within the Optimality Theory framework, the most obvious such mapping is the assigning of a structural description to a linguistic input. This mapping is naturally understood as a mapping from an underlying form to a description including the surface phonetic form, and thus as corresponding to language production. Language comprehension corresponds to a mapping from a surface form to a complete description which includes the underlying form. The exact relationship between these defining input/output mappings and the performance processes of comprehension and production remains an open issue.

The computability of these mappings was a matter of concern to some people early in the development of the theory. Because a grammar is described in terms of the simultaneous generation, evaluation, and comparison of an infinite number of candidates, a literal, mechanical interpretation of this description would suggest that to compute the optimal description of an input, an algorithm would need to generate and evaluate an infinite number of candidate descriptions. Fortunately, work on the computation of optimal descriptions has shown that it is not necessary to generate and evaluate an infinite number of candidates in order to compute the optimal description. Provably correct algorithms have been developed which efficiently compute the optimal description for several classes of grammars

Section 4.1 presents an illustration of an algorithm for efficiently computing optimal descriptions for a grammar employing an infinite candidate set. Section 4.2 discusses some philosophical affinities between Optimality Theory and contemporary work on cognitive modeling in connectionism and dynamical systems theory.

3.1 Computing Optimal Structural Descriptions

The infinite size of the candidate set results from the possibility of candidates containing material not found in the corresponding input; epenthesis (insertion) in the CV syllable theory is an example. Once the possibility of such insertions is granted (a rather uncontroversial position), there isn't any principled basis for placing any absolute limit on the number of insertions a candidate description may contain. Fortunately, no such limit is necessary, because the faithfulness constraints, specifically NO-DEL, are violated by such deviations from the input, and thus occur in grammatical descriptions only to the extent necessary to satisfy other high-ranked constraints. Thus the independently motivated faithfulness constraints provide the means to ensure that grammatical descriptions have only a finite number of insertions.

The faithfulness constraints themselves say nothing about how to compute optimal descriptions. However, an approach to computing optimal descriptions has been developed by Tesar (1995a, 1995b, 1996). This approach is based upon the computational technique of dynamic programming. This section provides a brief sketch of the approach.

The key to the dynamic programming approach is to build up to the optimal description piece by piece, rather than generating and considering lots of complete candidates. The algorithm uses a table to record the pieces, called partial descriptions, as they are built, and uses the pieces already recorded to build bigger pieces, until a small number of full descriptions are built, one of which is guaranteed to be optimal. Table 15 shows the table built by the algorithm when computing the optimal description for input /ap/. The ranking in use is that shown in (5).

(5) ONSET ≫ NOCODA ≫ NOINSV ≫ NODEL ≫ NOINSC

The table has four rows. The rows labeled for each of the possible syllabic positions (**Onset**, **Nucleus**, **Coda**) contain partial descriptions with the final (rightmost) position corresponding to the row label: the **Onset** row only contains partial descriptions ending with an onset position, and so forth. The **None** row contains partial descriptions with no syllabic positions; this limits the row to partial descriptions containing only deleted input segments. There is a column for each input segment (in order), plus a **Begin** column containing partial descriptions with none of the input segments. The interpretation of the cell in row **Nucleus** and column $\mathbf{i}_1$, [**Nucleus**, $\mathbf{i}_1$], is that it contains the most harmonic partial description containing the input segments up to $\mathbf{i}_1$ and having a nucleus as the rightmost position.

	Begin	$\mathbf{i}_1$ = a	$\mathbf{i}_2$ = p
None		NODEL *	NODEL **
Onset	.*ʔ* NOINSC *	.*ʔ*a.*ʔ* NOINSC **	.*ʔ*a.p NOINSC *
Nucleus	.*ʔi*. NOINSC * NOINSV *	.*ʔ*a. NOINSC *	.*ʔ*a. NOINSC * NODEL *
Coda	.*ʔiʔ*. NOINSC ** NOINSV * NOCODA *	.*ʔ*a*ʔ*. NOINSC ** NOCODA *	.*ʔ*ap. NOINSC * NOCODA *

Table 15 Dynamic programming table for the input /ap/ in L_1; each cell contains a partial description, along with the description's constraint violations. Epenthesized segments are shown in italics.

The algorithm fills the columns in succession, left to right. A cell is filled by considering ways of adding structure to the partial descriptions already contained in the table. Out of all the partial descriptions considered for a cell, the one that is most harmonic with respect to the constraint ranking actually fills the cell. The **Begin** column is filled first. It is relatively easy to fill; because this column is prior to the consideration of the first input segment, each cell is filled with the structure containing the minimum number of insertions necessary to reach the syllable position indicated by the row.

Consider the ways to fill cell [**Nucleus**, $\mathbf{i}_1$], assuming that the **Begin** column has already been filled. The structure in [**Nucleus**, **Begin**], .*ʔi*., could be used with $\mathbf{i}_1$ deleted, incurring a NODEL violation, with the result having violations {NOINSC NOINSV NODEL}. The structure in [**Onset**, **Begin**] could be used with $\mathbf{i}_1$ parsed into a nucleus position, giving .*ʔ*a., which has violations {NOINSC}. The structure in [**Onset**, $\mathbf{i}_1$] could have an inserted nucleus position added, giving .ʔa.ʔí., which has violations {NOINSC NOINSC NOINSV}. When the limit set of possibilities for filling the cell are compared, the most harmonic, .ʔa., is placed in the cell. When processing of the column $\mathbf{i}_1$ is complete, the cell entries are used as building blocks for filling the cells of the next column. This is how the algorithm keeps the large number of combinations forming different candidates under control: as each input segment is considered, the different ways of adding that segment are winnowed down to four (one for each row), and then only those partial descriptions are used when considering how to add the next input segment.

Actions filling the cells of a column must be coordinated. This is because insertion actions can add onto an entry in one cell to fill another cell of the same column, and these dependencies are cyclical. The solution is to first consider only candidates that delete or parse the new input segment, because those actions only add onto partial descriptions in the previous column. Then partial descriptions with one insertion at the right edge are considered. If any of them replace the previous entry in a cell, then partial descriptions with two insertions at right edge are considered. Soon the accumulated faithfulness violations (each insertion is a separate violation of either NOINSC or NOINSV) will prevent any further changes in the cell contents; at that point the column may be declared finished, and the algorithm proceeds to the next column. In this way, the algorithm uses the constraints to tell it how many insertions need to be considered.

Once all of the cells have been filled, the optimal description may be selected from the final column. In the example in table 15, the optimal description is .ʔa., in cell [**Nucleus**, $\mathbf{i}_2$].

3.2 Relation to Connectionism and Dynamical Systems

Although it is perhaps not immediately apparent, Optimality Theory shares some principles with connectionism and dynamical systems theory. In fact, Optimality Theory traces its origin to an effort by Prince and Smolensky to combine generative grammar and optimization ideas operative in certain forms of connectionism (Prince and Smolensky 1991, 1997). As emphasized in Section 1, Optimality Theory's strong commitments are about grammatical functions, not algorithms. The connection, then, is between Optimality Theory and connectionist theory, the mathematical characterizations of connectionist network behavior.

Perhaps the single most influential concept in connectionist theory is the understanding of networks in terms of optimization (Hopfield 1982). The weighted connections of the network define a function, often labeled energy or harmony (Smolensky 1986), over the space of possible activation states. Different inputs determine different spaces of possible activation states. For a given input, the network searches for the available activation state which optimizes the harmony function defined over those activation states. Issues in connectionist algorithms concern when connectionist algorithms do or do not converge on the network configuration optimizing the harmony function. Issues in connectionist theory concern the nature of the function which maps the network inputs to their globally optimal configurations as determined by the weighted connections of the network (independent of any particular algorithm).

The connections of the network determine the harmony function, and can be understood as constraints on the activations of the units they connect, and thus as constraints evaluating network activation states. However, the common situation is that the connections (constraints) conflict: there exists no configuration satisfying all of the connections (constraints) (McClelland and Rumelhart 1981). Thus, connectionist networks can be understood in terms of optimization over violable, or “soft”, constraints. But that is how Optimality Theory is defined: optimization of violable constraints over a space of candidate representations.

The most significant difference between Optimality Theory and connectionist theory is the nature of the harmony function. Connectionist theory uses numerical optimization: the constraints are assigned numeric weights, so the relative strength of different constraints is determined by the relative magnitudes of their respective numeric weights. Optimality Theory uses strict domination optimization (Principles 3 and

4). The possible relationships between connectionist numeric optimization and Optimality theoretic strict domination optimization is a wide open topic, the subject of much future research.

3.3 Other Work

Ellison (1994) has developed a way of representing the candidate set for an input as a finite state automaton, along with an algorithm for identifying the optimal candidate by searching for the least-weight (highest harmony) path through the automaton. The least-weight path algorithm, based upon dynamic programming, is not itself a finite-state algorithm. Work of a similar spirit has been done by Eisner (1997). Frank and Satta (1998) have investigated actual finite-state algorithms for computing optimal forms, demonstrating that faithfulness as normally conceived is strictly beyond the computational capacity of finite-state automata, but can be approximated. Hammond (1997) has computationally modeled the comprehension of syllable structure.

Several computational tools related to Optimality Theory have been made available on the Internet. Tools for simulating OT generation include those created by Andrews (1994), Walther (1996), Hammond (1997). Raymond and Hogan 1993 is a tool for aiding in the study of factorial typologies, and the software suite of Hayes (1998) facilitates both factorial typology and the study of particular systems. The Rutgers Optimality Archive "Utilities" page provides links to these programs.

4 Language Learnability and Acquisition

Cross-linguistic variation is explained in Optimality Theory through the rankings of the universal constraints. Therefore, an important part of language learning in Optimality Theory is learning the correct ranking of the universal constraints for a given language, from positive data. One challenging aspect of this problem in Optimality Theory is imposed by the use of violable constraints. Given a grammatical description, a learner might observe that it violates some of the universal constraints. But if grammatical descriptions are allowed to violate constraints, how can anything be learned from those observations? There is also a combinatorial concern[2]. The number of distinct total rankings is a factorial function of the number of constraints: 10 constraints have 10!= 3,628,800 rankings, and 20 constraints have 20!= 2,432,902,008,176,640,000 rankings. If the amount of data required to learn the correct ranking scales as the number of possible rankings, then a grammar with many constraints could require a prohibitively large amount of data to be learned successfully.

Fortunately, the problem of finding a ranking consistent with a set of grammatical descriptions turns out to be quite tractable.. In fact, the optimizing structure of Optimality Theory can be a significant asset with respect to language learnability. Section 4.1 describes an approach to language learning which makes use of the formal structure of Optimality Theory. Section 4.2 describes recent work using Optimality Theory grammars to account for child language acquisition data.

4.1 Learnability

The learning of constraint rankings is illustrated here using a simplified optimality theoretic analysis of

[2]This combinatorial concern is similar to that faced by the Principles and Parameters framework (Chomsky 1981). A system with 10 binary parameters has 2^{10}= 1024 distinct parameter settings, a system with 20 binary parameters has 2^{20} = 1,048,576. While these exponential functions grow fast enough to make brute-force enumeration impractical, the factorial functions of possible rankings grow faster yet.

metrical stress (more general systems, along with corresponding learning results, can be found in Tesar 1997, 1998b). In this system, languages assign stress to a word by first grouping two of the word's syllables into a foot, and then assigning main word stress to one of the syllables in the foot. The foot is assigned at one edge of the word, and languages vary as to which edge (left or right) the foot is assigned. Another form of variation is the form of the foot. A trochaic foot assigns stress to the first of the two syllables of the foot, while an iambic foot assigns stress to the second syllable of the foot. A word is here delimited by square brackets, with parentheses indicating the foot grouping. Each numeral represents the stress level of a syllable, with 1 denoting primary stress, 2 denoting secondary stress, and 0 denoting an unstressed syllable. The word [(1 0) 0 0 0] has five syllables, with the first two grouped into a trochaic foot; the result is that main stress falls on the first syllable.

The optimality theoretic analysis presented here uses four constraints, shown in table 16. The constraint ALLFTL is violated once for each syllable intervening between a foot and the left edge of the word.

Name	*Content*
ALLFTL	A foot must be at the left edge of the word.
ALLFTR	A foot must be at the right edge of the word.
TROCH	A foot must stress its first syllable.
IAMB	A foot must stress its last syllable.

Table 16 Constraints for metrical stress theory

The challenge of learning constraint rankings is illustrated in table 17. The winner is the grammatical description. The goal of learning is to find a constraint ranking that makes the winner more harmonic than all competitors, such as the loser in table 17. For the winner to be more harmonic than the loser, at least one of the constraints violated by the loser must dominate all of the constraints violated by the winner. The precise information contained in this loser/winner pair is given in (6).

(6) (ALLFTR **or** IAMB) ≫ (ALLFTL **and** TROCH)

The tricky part is the disjunction (the logical **or**) of the constraints violated by the loser: we know one of them must dominate the constraints violated by the winner, but we don't know which (if not both). In systems with a larger number of constraints, it may be possible for a single loser/winner pair to have quite a few constraints in the disjunction, and attempting to maintain and reconcile such information across many examples could be difficult.

		ALLFTL	ALLFTR	IAMB	TROCH
winner	[0 (0 1)]	*			*
loser	[(1 0) 0]		*	*	

Table 17 The disjunction problem

A solution to this problem is the Error-Driven Constraint Demotion algorithm of Tesar and

Smolensky (1995, 1998). At any given time this algorithm has a hypothesis ranking. The algorithm identifies the highest-ranked constraint violated more by the loser. Every constraint which (a) is violated more by the winner, and (b) is not currently dominated in the hierarchy by the highest-ranked loser constraint, is demoted to immediately below the highest-ranked loser constraint. This ensures that the resulting ranking will hold the winner more harmonic than the loser. This is illustrated in table 18 for the loser/winner pair in table 17, assuming that the starting hypothesis ranking is the ranking in (7).

(7) ALLFTL $\gg$ ALLFTR $\gg$ IAMB $\gg$ TROCH

The highest-ranked constraint violated by the loser is ALLFTR. One constraint violated by the winner, TROCH, is already dominated by ALLFTR, and so is left alone. The other constraint violated by the winner, ALLFTL, is not so dominated. Thus, constraint demotion demotes ALLFTL to the stratum immediately below ALLFTR (effectively creating a tie between ALLFTL and IAMB). With respect to the resulting constraint hierarchy, the winner is more harmonic than the loser.

		Before	After			
		(ALLFTL)	ALLFTR	ALLFTL	IAMB	TROCH
winner	[0 (0 1)]	(*)		*		*
loser	[(1 0) 0]		*		*	

Table 18 Constraint demotion: ALLFTL is demoted to below ALLFTR, into a tie with IAMB

The hypothesis rankings used by the algorithm are of a particular form, called a stratified hierarchy. A stratified hierarchy consists of ranked strata, where each stratum contains one or more of the constraints. The hierarchy resulting from the constraint demotion in table 18 is an example: ALLFTL and IAMB are in the same stratum, neither dominating the other. This freedom to have hypotheses that aren't totally ranked is important to the success of the algorithm. However, the algorithm will always converge to a hierarchy consistent with at least one total ranking.

The algorithm is error-driven in the way that it selects loser/winner pairs to be used for demotion (Tesar 1995b, 1998a). When presented with a grammatical description, the algorithm computes the description for the same input that is optimal with respect to the current hypothesis hierarchy. If it is not the same as the grammatical description, then an error has occurred, and the grammatical description is made the winner, and the description currently optimal is made the loser. If the currently optimal description matches the given grammatical one, no error has occurred, so no learning takes place.

The error-driven structure of the algorithm makes it convenient to measure data complexity in terms of the number of errors (mismatches that result in constraint demotions) that can occur prior to convergence on a hierarchy generating the correct language. There is a mathematically provable bound on the worst case number of errors that can occur prior to convergence: N(N-1) errors, where N is the number of constraints. In practice, this worst case is a large overestimate, and the algorithm reaches the correct ranking far more quickly. This demonstrates that the amount of data required to learn a ranking does not scale anything like the number of total rankings.

No. of Constraints	*No. of Total Rankings*	*Maximum no. of Errors*
5	5! = 120	5(5-1) = 20
10	10! = 3,628,800	10(10-1) = 90
20	20! = 2,432,902,008,176,640,000	20(20-1) = 380

Table 19 Data complexity of constraint demotion

Constraint demotion is guaranteed to find a correct ranking, given the correct full structural descriptions of grammatical utterances. But that is only part of the learning story. One of the challenging aspects of language learning is that the learner does not have direct access to full structural descriptions. The learner only has direct access to the audible, overt information in the phonetic stream, referred to here as an overt form. The problem is made challenging by the fact that overt forms are often ambiguous: they are consistent with more than one distinct full structural description. Figuring out the correct descriptions of the overt forms is part of the learner's task. An example is the overt form [0 1 0], that is, a three-syllable word with stress on the middle syllable. This overt form is ambiguous between the interpretations of a left-aligned iambic foot, [(0 1) 0], and a right-aligned trochaic foot, [0 (1 0)]. The “hidden structure” not present in the overt form is the foot structure.

Recent work on learnability has investigated a learning strategy for overcoming ambiguity by capitalizing on Optimality Theory's optimizing structure (Tesar and Smolensky 1996; Tesar 1997, 1998b). The strategy, inspired by work in statistical learning theory (Baum 1972; Dempster, Laird, and Rubin 1977), is to give the learner a starting hypothesized constraint ranking, and then compute the structural description for an overt form that is most consistent with the hypothesized constraint ranking. Such a computation is dubbed *robust interpretive parsing*; it is in essence the same mapping as that suggested earlier for language comprehension. The advantage provided by Optimality Theory is that the interpretive mapping is defined even for overt forms which are not consistent with a learner’s hypothesis grammar (constraint ranking), hence the label robust. Robust interpretive parsing, when given an overt form and the learner’s current ranking, selects, from among those structural descriptions with an overt portion matching the overt form, the description most harmonic with respect to the learner’s current ranking, even if the ranking selects as optimal a different structural description (one with a different overt portion). When presented with an overt form inconsistent with its current grammar, the learner makes its best attempt to interpret the utterance.

The iterative strategy is for the learner to use this best guess interpretation for learning, on the (quite possibly mistaken) assumption that it is correct. The interpretation is used for learning by treating it as a winner and applying constraint demotion as described above. The intuition behind the strategy is that even if the interpretation arrived at by the learner is incorrect, it will still contain useful information because it has been constrained to match the overt form (the information the learner is attempting to learn from). This approach results in an iterative procedure: use a hypothesized constraint ranking to construct hypothesized full structural descriptions of available overt forms, then use the hypothesized structural descriptions to construct a new hypothesized constraint ranking (via constraint demotion). This may then be repeated, back and forth, using the new hypothesized ranking to determine new hypothesized structural descriptions of the overt forms. Learning is successful when the process converges on the correct ranking and the correct structural descriptions. This strategy has been investigated with Optimality Theoretic systems for metrical stress with quite promising results (Tesar 1997, 1998b).

As an illustration, suppose that the learner is attempting to learn a language which places the main stress foot at the right of the word, and uses a trochaic foot. However, the learner here starts with the constraint ranking shown in (8), generating a very different language.

(8) ALLFTL ≫ ALLFTR ≫ IAMB ≫ TROCH

Suppose the learner is confronted with the overt form [0 0 0 1 0], a five-syllable word with main stress on the fourth syllable. The learner will apply interpretive parsing to this form, using its current ranking, the result being as shown in the winner row of table 20. The learner will also use its ranking to determine the optimal description of a five-syllable word in its current grammar, which is an iambic foot on the left side of the word, and shown in the loser row of table 20.

		ALLFTL	ALLFTR	IAMB	TROCH
winner	[0 0 (0 1) 0]	* *	*		*
loser	[(0 1) 0 0 0]		* * *		*

Table 20 Before the first demotion.

The winner matches the overt form; it has main stress on the fourth syllable. Notice, however, that the winner contains an incorrect analysis; it assigns an iambic foot grouping the third and fourth syllables, while the language being learned in fact requires a trochaic foot grouping the last two syllables. The learner, however, has no way of knowing this in advance, and proceeds (temporarily) on the assumption that its analysis of the overt form is correct. The learner proceeds by applying constraint demotion, which demotes ALLFTL to below ALLFTR, and into a tie with IAMB.

(9) ALLFTR ≫ {IAMB ALLFTL} ≫ TROCH

The learner has now modified its ranking, but it is not done yet. Now that it has modified its ranking, it re-interprets the overt form using the new constraint ranking. Table 21 shows the new interpretation in the winner row; the loser row shows the optimal stress assignment for a five syllable word under the new ranking.

		ALLFTR	ALLFTL	IAMB	TROCH
winner	[0 0 0 (1 0)]		* * *	*	
loser	[0 0 0 (0 1)]		* * *		*

Table 21 Before the second demotion

The interpretation of the overt form has now changed, and for the better: the learner now has the correct interpretation of the overt form. The learner now applies constraint demotion again, demoting IAMB to below TROCH; the result is shown in table 22.

		ALLFTR	ALLFTL	TROCH	IAMB
winner	[0 0 0 (1 0)]		* * *		*
loser	[0 0 0 (1 0)]		* * *		*

Table 22 After the second demotion, the learner has reached the correct ranking

What is important is that even though the learner initially misanalyzed the overt form, due to an incorrect ranking, the learner was able to make progress using its own analysis. The overt information was enough to get the learner moving in the right direction, ultimately arriving at both the correct analysis and the correct constraint ranking.

4.2 Acquisition

Several recent papers on child language acquisition have accounted for observed patterns in child language development in terms of changes in constraint rankings (Levelt 1994, to appear; Demuth 1995; Gnanadesikan 1995). There is a theme that has emerged in much of this work, one that has been explicitly articulated by Smolensky (1996): children start with the faithfulness constraints ranked below structural constraints. This results in early child language production exhibiting only the most unmarked forms. The claim here is that children have underlying forms closely matching the surface forms they hear from adults, but that the low ranking of faithfulness results in outputs which modify the structure to contain less marked forms than appear in the underlying form (or appear in the surface forms they hear). Over time, the children demote some structural constraints to below some faithfulness constraints, permitting marked structures to appear in optimal descriptions and thus in their language production.

4.3 Other Work

Turkel (1994) has investigated the learning of Optimality Theory rankings using genetic algorithms. Broihier (1995) has investigated issues in the learning of OT grammars which make use of non-vacuous ties among constraints.

5 Summary

Optimality Theory explains linguistic phenomena in terms of optimization over violable constraints. In so doing, it formalizes notions of linguistic markedness and economy and makes them central to the explanations. Linguistic markedness is captured by the use of constraints which can be violated, but are nevertheless present and active in all languages. Economy is reflected (and generalized) in the use of optimization to choose among several ways of describing an input: the grammar always selects the best way possible. Optimality Theory shares several underlying principles with certain forms of connectionism, and those shared principles are apparent in work on language processing within Optimality Theory. By defining grammaticality in terms of optimization over violable constraints, and by differentiating the strength of constraints via strict domination, Optimality Theory makes possible new directions in language learning, with some significant results already obtained both in formal language learnability and in empirical language acquisition studies. Current and future work within Optimality Theory promises to provide further interesting results and directions for research on fundamental issues of linguistic theory.

6 Acknowledgments

All of the analyses presented in this paper are simplified from the originals for reasons of space and clarity. The authors alone are responsible for any and all errors and misrepresentations.

References

Ackema, Peter and Ad Neeleman. 1998. WHOT? In Barbosa, Pilar, Danny Fox, Paul Hagstrom, Martha McGinnis and David Pesetsky eds. *Is the Best Good Enough?* The MIT Press and MIT Working Papers in Linguistics, Cambridge, MA. 15-33.

Andrews, Avery. 1994. OT for Windows 1.1 (prolog/C++ program). ROA-91.

Barbosa, Pilar, Danny Fox, Paul Hagstrom, Martha McGinnis and David Pesetsky eds. 1998. *Is the Best Good Enough?* The MIT Press and MIT Working Papers in Linguistics, Cambridge, MA.

Baum, L. 1972. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequalities* 3, 1-8.

Beckman, Jill, Laura Walsh Dickey, and Suzanne Urbanczyk, editors. 1995. *Papers on Optimality Theory*. GLSA, University of Massachusetts, Amherst, MA. University of Massachusetts Occasional Papers 18.

Bresnan, Joan. To appear. The Emergence of the Unmarked Pronoun: Chichewa Pronominals in Optimality Theory. In *Proceedings of the Berkeley Linguistics Society* 23.

Broihier, Kevin. 1995. Optimality Theoretic rankings with tied constraints: Slavic relatives, resumptive pronouns and learnability. Ms., Department of Linguistics, MIT, ROA-46.

Choi, Hye-Won. 1996. *Optimizing Structure in Context: Scrambling and Information Structure*. Ph.D. thesis, Stanford University.

Chomsky, Noam. 1965. *Aspects of the Theory of Syntax*. MIT Press, Cambridge, MA.

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

Chomsky, Noam. 1991. Some notes on economy of derivation and representation. In Robert Freidin, eds., *Principles and Parameters in Comparative Grammar*. MIT Press, Cambridge, MA. 417-453.

Chomsky, Noam. 1995. *A Minimalist Program for Linguistic Theory*. MIT Press, Cambridge, MA.

Clements, G. and Samuel Keyser. 1983. *CV Phonology*. MIT Press, Cambridge, MA.

Dempster, A., Laird, N., and Rubin, D. 1977. Maximum likelihood from incomplete data via the *EM* algorithm. *Journal of the Royal Statistical Society B* 39, 1-38.

Demuth, Katherine. 1995. Markedness and the development of prosodic structure. In *Proceedings of NELS 25*. GLSA, University of Massachusetts, Amherst.

Eisner, Jason. 1997. Efficient generation in primitive Optimality Theory. Ms., ROA-206.

Ellison, T. Mark. 1994. Phonological derivation in Optimality Theory. In *Proceedings of the Fifteenth International Conference on Computational Linguistics*, 1007-1013. ROA-75.

Frank, Robert, and Satta, Giorgio. 1998. Optimality Theory and the generative complexity of constraint violability. *Computational Linguistics* 24(2), 307-315.

Gnanadesikan, Amalia. 1995. Markedness and faithfulness constraints in child phonology. Ms., Department of Linguistics, University of Massachusetts, Amherst.

Grimshaw, Jane, 1997a. The best clitic: Constraint interaction in morphosyntax. In Lilliane Haegeman, ed., *Elements of Grammar*: *Handbook of Generative Syntax.* Kluwer. 169-196.

Grimshaw, Jane.1997b. Projection, Heads and Optimality. *Linguistic Inquiry* 28, 373-422.

Grimshaw, Jane. To appear. Constraints on constraints in optimality theoretic syntax. In Mark Baltin and Chris Collins, eds., *The Handbook of Contemporary Syntactic Theory*. Blackwell.

Grimshaw, Jane and Vieri Samek-Lodovici. 1995. Optimal Subjects. In Jill Beckman, Laura Walsh Dickey, and Suzanne Urbancyzk, eds., *Papers in Optimality Theory*. GLSA, University of Massachusetts, Amherst, MA, pages 589-605.

Grimshaw, Jane and Vieri Samek-Lodovici. 1998. Optimal subjects and subject universals. In Barbosa, Pilar, Danny Fox, Paul Hagstrom, Martha McGinnis and David Pesetsky, eds., *Is the Best Good Enough?* The MIT Press and MIT Working Papers in Linguistics. 193-219.

Hammond, Michael. 1997. Parsing syllables: modeling OT computationally (prolog program). Ms., Department of Linguistics, University of Arizona. (ROA-222)

Hayes, Bruce P. 1998. Optimality Theory Software. UCLA: Department of Linguistics website. http://www.humnet.ucla.edu/humnet/linguistics/people/hayes/otsoft/otsoft.htm

Hopfield, J. J. 1982. Neural networks and physical systems with emergent collective computational abilities. In *Proceedings of the National Academy of Sciences 18*, 2554-2558.

Jakobson, Roman. 1962. *Selected Writings 1: Phonological Studies*. Mouton, The Hague.

Legendre, Geraldine, Colin Wilson, Paul Smolensky, Kristin Homer, and William Raymond. 1995. Optimality and *wh*-extraction. In Jill Beckman, Laura Walsh Dickey, and Suzanne Urbancyzk, eds., *Papers in Optimality Theory*. GLSA, University of Massachusetts, Amherst, MA. 607-636.

Legendre, Géraldine, Paul Smolensky and Colin Wilson. 1998. When is less more? Faithfulness and minimal links in *wh*-chains. In Barbosa, Pilar, Danny Fox, Paul Hagstrom, Martha McGinnis and David Pesetsky, eds., *Is the Best Good Enough?* The MIT Press and MIT Working Papers in Linguistics. 249-289.

Levelt, Clara. 1994. *On the Acquisition of Place*. Holland Institute of Generative Linguistics Dissertations in Linguistics 8. Holland Academic Graphics, den Haag.

Levelt, Clara. To appear. The segmental structure of early words: Articulatory frames or phonological constraints? In Eve Clark, ed., *Proceedings of the 27th Child Language Research Forum*, Stanford University. CSLI Publications, Stanford, CA.

McCarthy, John and Alan Prince. 1993a. Generalized alignment. In G. Booij and J. van Marle, eds., *Yearbook of Morphology*. Kluwer, Dordrecht. 79-154.

McCarthy, John and Alan Prince. 1993b. Prosodic morphology I: Constraint interaction and satisfaction. Technical report, TR-3, Rutgers University Center for Cognitive Science. To appear in the Linguistic Inquiry Monograph Series, MIT Press.

McCarthy, John and Alan Prince. 1994. The emergence of the unmarked: Optimality in prosodic morphology. In Mercè Gonzàlez, ed., *Proceedings of the North-East Linguistics Society 24*. GLSA, University of Massachusetts, Amherst, MA. 333-379. ROA-13.

McCarthy, John and Alan Prince. 1995. Faithfulness and Reduplicative Identity. In Jill Beckman, Laura Walsh Dickey, and Suzanne Urbancyzk, eds., *Papers in Optimality Theory*. GLSA, University of Massachusetts, Amherst, MA. 249-384.

McClelland, James, and David Rumelhart. 1981. An interactive activation model of context effects in letter perception: Part 1. An account of basic findings. *Psychological Review* 88, 375-407.

Müller, Gereon. 1997. Partial wh-movement and Optimality Theory. *The Linguistic Review* 14, 249-306.

Pesetsky, David. 1998. Some optimality principles of sentence pronunciation. In Barbosa, Pilar, Danny Fox, Paul Hagstrom, Martha McGinnis and David Pesetsky, eds., *Is the Best Good Enough?* The MIT Press and MIT Working Papers in Linguistics. 337-383.

Prince, Alan and Paul Smolensky. 1991. Notes on connectionism and Harmony Theory in linguistics. Technical report CU-CS-533-91, Department of Computer Science, University of Colorado, Boulder.

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, Boulder. To appear in the Linguistic Inquiry Monograph Series, MIT Press.

Prince, Alan and Paul Smolensky. 1997. Optimality: From neural networks to universal grammar. *Science* 275, 1604-1610.

Raymond, William, and Hogan, Apollo. 1993. The optimality interpreter, and a user's guide to the optimality interpreter (C program). ROA-130.

Samek-Lodovici, Vieri. 1996. Constraints on subjects: an optimality theoretic analysis. Ph.D. thesis, Rutgers University, May.

Samek-Lodovici, Vieri. 1998. Opposite constraints: Left and right focus-alignment in Kanakuru. *Lingua* 104.

Sells, Peter, John Rickford, and Thomas Wasow. 1994. An Optimality-Theoretic approach to variation in negative inversion in AAVE. Ms., Stanford University, ROA-53.

Smolensky, Paul. 1986. Information processing in dynamical systems: Foundations of harmony theory. In David Rumelhart and James McClelland, eds., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations*. MIT Press/Bradford Books, Cambridge, MA. 194-281.

Smolensky, Paul. 1996. On the comprehension/production dilemma in child language. *Linguistic Inquiry* 27, 720-731. ROA-118.

Speas, Margaret. 1997. Optimality Theory and Syntax: Null Pronouns and Control. In Diana Archangeli and D. Terrence Langendoen, eds., *Optimality Theory: An Overview*. Blackwell. 171-199.

Tesar, Bruce. 1995a. Computing optimal descriptions for Optimality Theory: Basic syllabification. Technical Report CU-CS-763-95, Department of Computer Science, University of Colorado, Boulder. February. ROA-52.

Tesar, Bruce. 1995b. Computational Optimality Theory. Ph.D. thesis, Department of Computer Science, University of Colorado, Boulder. ROA-90.

Tesar, Bruce. 1996. Computing optimal descriptions for Optimality Theory grammars with context-free position structures. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, 101-107.

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*. Somerville, MA, Cascadilla Press. 615-626. ROA-177.

Tesar, Bruce. 1998a. Error-driven learning in Optimality Theory via the efficient computation of optimal forms. In Barbosa, Pilar, Danny Fox, Paul Hagstrom, Martha McGinnis and David Pesetsky, eds., *Is the Best Good Enough?* The MIT Press and MIT Working Papers in Linguistics. 421-435.

Tesar, Bruce. 1998b. An iterative strategy for language learning. *Lingua* 104, 131-145.

Tesar, Bruce and Paul Smolensky.1995. The learnability of Optimality Theory. In *Proceedings of the Thirteenth West Coast Conference on Formal Linguistics*, CSLI Publications, Stanford, CA. 122-137.

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* 29(2), 229-268.

Turkel, William. 1994. The acquisition of Optimality Theoretic systems. Ms., Department of Linguistics, University of British Columbia, ROA-11.

Walther, Markus. 1996. OT SIMPLE - a construction kit approach to Optimality Theory implementation. ROA-152.