1 Language Learning

1.1 What This Book Is About

This book argues that the linguistic framework of *Optimality Theory* (OT) (Prince and Smolensky 1993) makes possible a particularly strong union of the interests of language learnability and linguistic theory. In support of this claim, a particular approach to language learning, *Robust Interpretive Parsing / Constraint Demotion* (RIP/CD), is presented and evaluated. This learning proposal is tightly bound to the central principles of OT, and the success of the learning proposal is evidence in favor of the main claim.

The language learning issue of primary concern in this book is the ambiguity of the overt information that constitutes the actual data received by a learner, and the resulting interdependence of the core grammar and the structural analysis of overt linguistic forms: which grammar a learner chooses depends on how they interpret the forms they hear, and which analysis they choose for a form depends on what grammar they are using. The RIP/CD proposal claims that this interdependence can be finessed by successive iteration: the learner can use a first guess at a grammar to estimate the structural analysis of the data, use the estimated analyses to improve the grammar, use the improved grammar to improve the analyses, and so forth. The learning procedure learns both the correct interpretations of the data and the correct grammar simultaneously. The viability of this "back-and-forth" strategy is heavily dependent on the use of OT to characterize the knowledge of language that the learner comes to possess.

The RIP/CD learning proposal is evaluated by a series of computer experiments, applying the proposal to overt data from a number of languages generated by an OT system for metrical stress. This system exhibits a nontrivial degree of ambiguity in the overt forms: most overt forms have several viable structural interpretations, with different interpretations favored by different grammars of the system. The performance is evaluated both on accuracy—whether or not the correct grammar was in fact learned—and computational efficiency—the amount of effort exerted during the process of learning the correct grammar.

The empirical results just mentioned are supported by stronger formal results concerning major parts of the proposal. It is not necessary to conduct any simulations to attempt to measure the amount of information required by the learner to determine the correct grammar, because

of a strong upper bound on the amount of data required. This result, which applies to all language systems defined within OT, is proved correct in chapter 7. This result is an important part of the proposal made here, for it demonstrates that the adoption of OT guarantees a strong solution to one of the major issues in language learning.

Chapter 1 is devoted to laying out the larger context of this work, including the nature of relationships between learnability and universal grammar, and the background work on general learning theory that has informed and inspired the specific language learning proposal made here. Readers who would prefer to skip the background on the first reading are advised to jump to section 1.4, which presents a top-level outline of the proposals made in this book, along with pointers to the location of each topic within the book.

1.2 Learnability and Universal Grammar

It has become commonplace in generative linguistics circles to see the logical problem of language acquisition as a driving force in shaping grammatical theory (Chomsky 1981). The basic logic is essentially as follows. Learning a grammar is difficult because there are so many conceivable grammars and the available data is so impoverished. Thus a crucial job of a theory of universal grammar is to *restrict* the space of possible grammars the learner must consider, so that impoverished data may suffice to determine a correct grammar. This notion of restrictiveness is often reduced to the criterion that a satisfactory grammatical theory will delimit a finite set of possible grammars—distinguished from one another by the values of a finite number of *parameters*, for example. The fewer the possible grammars, the more learnable the theory.

Or so it would seem. In fact, however, limiting the set of possible grammars to a finite number serves only to improve the worst-case performance of the *least informed* learning method of all: exhaustive search, in which every possible hypothesis is examined. For, with finitely many possible grammars, search for a correct one is guaranteed to terminate eventually: at worst, once all possible grammars have been examined. With infinitely many possible grammars, such search may continue forever.

But comfort from the finiteness of the space of possible grammars is tenuous indeed. For a grammatical theory with an infinite number of pos-

sible grammars might be well structured, permitting *informed* search that converges quickly to the correct grammar—even though uninformed, exhaustive search is infeasible. And it is of little value that exhaustive search is guaranteed to terminate eventually when the space of possible grammars is finite, if the number of grammars is astronomical. In fact, a well-structured theory admitting an infinity of grammars could well be feasibly learnable,[1] while a poorly structured theory admitting a finite, but very large, number of possible grammars might not.

And indeed, a *principles-and-parameters* (P&P) *universal grammar* (UG) with n parameters admits at least 2^n grammars; more, if the parameters are not binary. Such exponential growth in the number of parameters quickly leads to spaces much too large to search exhaustively. An OT UG with N constraints admits $N!$ grammars, which grows still faster.

Thus to achieve meaningful assurance of learnability from our grammatical theory, we must seek evidence that the theory provides the space of possible grammars with the *kind of structure* that learning can effectively exploit.

Consider P&P theory in this regard. Two types of learnability research are useful as contrasts to the results we offer in this book. The first is *cue learning*, exemplified by work such as Dresher and Kaye 1990. These authors adopt a particular parameterized space of grammars, and analyze in great detail the relationships between the parameter settings and the forms overtly available to the learner. They propose a *specific* learning algorithm to make use of the structure provided by a *specific* P&P theory. Their analysis is entirely limited to their particular parametric system for metrical stress; a cue learning approach to a parametric grammar for some other component of linguistic theory, or even to an alternative parametric analysis of metrical stress, would essentially require starting over from scratch.

Another approach to learnability within P&P, quite different from cue learning, is represented in the work of Gibson and Wexler (1994) and Niyogi and Berwick (1996). The *triggering learning algorithm* (and its variations) is designed to learn grammars from data overtly available to the learner. Like those developed in our work, these algorithms apply to any instance of a very general class of systems: in their case, the class of P&P systems. Further, Niyogi and Berwick (1996) provide formal analysis of the algorithms. However, this work differs from ours in a direction representing the opposite extreme from cue learning: these learning

algorithms are *minimally informed* by the grammatical theory. For triggering learning algorithms treat the grammar only as a black box evaluating learning data as either grammatically analyzable or not; the algorithms either randomly flip grammar parameters in order to render an input analyzable (Gibson and Wexler's *Triggering Learning Algorithm*), or randomly flip parameters without regard to immediate resulting analyzability (which, Niyogi and Berwick argue, can actually outperform the Triggering Learning Algorithm). These learning algorithms are equally appropriate as procedures for learning parameterized grammars and as procedures for, say, training a neural network[2] (with binary weights) to classify radar images of submarines: if flipping a parameter (connection in the network) gives better classification of a submarine, flip it. These are simply generic search algorithms that employ no properties of the grammatical theory per se.

Further, the learnability results relating to triggering learning algorithms assume the existence of overt data that directly reveal individual parameter values. Such an assumption limits how impoverished the learning data can be and has unclear relevance to realistic grammars (see Frank and Kapur 1996); we discuss this further in section 6.1. Finally, regardless of the availability of such "triggering" forms, these algorithms offer little justification for confidence in their tractability. In fact, the only result regarding the time required for learning is that the probability of learning the correct grammar increases toward 1 as the number of learning instances approaches infinity[3]—leaving open the possibility of doing even worse than exhaustive search.

In sum, these two approaches to learnability analysis within P&P either (1) use grammatical structure in the learning algorithm, but the structure of a *particular* parametric system, or (2) develop general algorithms applicable to any P&P system, but algorithms *so* general they apply just as well to any nongrammatical parameterized system. This dichotomy of approaches is likely a consequence of the nature of P&P. A particular P&P system, like one for stress, has sufficient structure to inform a learning procedure (option 1). But as a general theory of how grammars may differ (as opposed to how stress systems may differ), P&P provides little structure for a learner to exploit beyond the existence of a finite space for searching. In particular, P&P theory per se provides no characteristically *grammatical* structure for a language learner to exploit.

But the situation in OT is quite different. This theory is reviewed in chapter 2, but the immediately relevant claims of OT are these:

(1.1) OT in a nutshell

- What is it that all languages have in common? *A set of constraints on well-formedness.*
- How may languages differ? *Only in which constraints have priority in case of conflict.*
- Language-particular relative constraint priorities are characterized by a *ranking* of the universal well-formedness constraints into a *dominance hierarchy*, with each constraint having absolute priority over all lower-ranked constraints.
- The grammar of a particular language—its constraint hierarchy—is an evaluator of structural descriptions, assigning a (nonnumerical) *Harmony* value that assesses the degree to which the constraints are met, taking into account the language-particular priorities. This provides the *harmonic ordering of forms*, ordering structural descriptions from maximal to minimal Harmony.
- The grammatical forms of the language are the *optimal* ones: the well-formed structural description of an input is the one with maximal Harmony.

Note that the constraints mentioned in (1.1) are the same in all languages: they contain no parameters. Unlike P&P, this is a theory of crosslinguistic variation with sufficient structure to enable grammatically informed learning algorithms independent of substantive grammatical assumptions.

(1.2) Main claim of this book: OT is a theory of UG that provides sufficient structure at the level of the grammatical framework itself to allow general but grammatically informed learning algorithms to be formally defined. Further, the efficiency of the algorithms can be argued to follow in large part from the formal structure of the grammatical framework.

The algorithms we develop are procedures for learning the priority ranking of constraints that, by (1.1), is all that distinguishes the grammar of a particular language. These are unquestionably grammar learning algorithms, not generic search algorithms.[4] Yet the structure that makes

these algorithms possible is not the structure of a theory of stress, nor a theory of phonology: it is the structure defining any OT grammar, that given in (1.1).

Of course, if a grammatically *un*informed learning algorithm, such as the Triggering Learning Algorithm, is desired, it can be obtained as easily in OT as in P&P; in fact, Pulleyblank and Turkel (1995, 1998) have already formulated and studied the *Constraint-Ranking Triggering Learning Algorithm*. Indeed, we can apply any of a number of generic search algorithms to the space of OT grammars—for example, Pulleyblank and Turkel (1995, 1998) have also applied the genetic algorithm to learning OT grammars. But unlike P&P, with OT we have an alternative to grammatically uninformed learning: learning algorithms specially constructed to exploit the structure provided by OT's theory of crosslinguistic variation.

1.3 Decomposing the Learning Problem

1.3.1 Grammar Learning and Robust Interpretive Parsing

To begin our analysis of grammar learning, we must distinguish the following three types of linguistic structure:

(1.3) The players in order of their appearance

- *Overt part of grammatical forms*: directly accessible to the learner
- *Full structural descriptions*: combine overt and nonovert ("hidden") structure
- *The grammar*: determines which structural descriptions are well formed

These three elements are all intimately connected, yet we propose to distinguish two subproblems, as schematically shown in figure 1.1.

(1.4) Decomposition of the problem

- *Robust interpretive parsing*: mapping the overt part of a form into a full structural description, complete with all hidden structure—given a grammar
- *Learning the grammar*—given a (robust) parser

(An interpretive parser is "robust" if it can parse an overt structure with a grammar, even when that structure is not grammatical according to the grammar. The importance of robustness will be discussed shortly.)

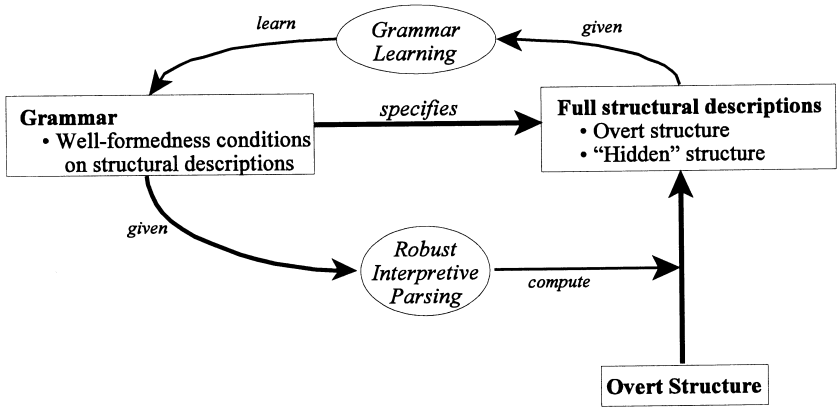

Figure 1.1
Problem decomposition

A competence theory of grammatical structure is most useful to an ultimate performance theory of language processing and acquisition when it provides sufficient structure so that procedures for both parsing and grammar learning can strongly exploit grammatical principles. Showing that this is indeed the case for OT is a major goal of our work.

We propose that the problems of parsing and grammar learning be decoupled to some degree. Such separation does at first seem problematic, however. One of the central difficulties of language learning, of course, is that grammars refer crucially to nonovert, hidden structure. Let us take the acquisition of stress as an expository example. The problem, then, is that the grammatical principles concern (say) metrical feet, yet these are hidden in the data presented to the learner: only the location of some stressed syllables is provided overtly. The learner cannot learn the metrical grammar until she knows where the feet lie, but she cannot know where the feet lie until she knows the grammar. We argue in section 1.3.2 that, despite this conundrum, partial decoupling of the parsing and learning problems is possible, and further, that such decoupling can enable powerful learning algorithms.

1.3.2 Iterative Model–Based Solutions to the Problem of Learning Hidden Structure

The learner cannot deduce the hidden structure in learning data until she has learned the grammar, but she cannot learn the grammar until

she has the hidden structure. This feature of the language learning problem is challenging indeed—but not at all special to language, as it turns out. Even in such mundane contexts as a computer learning to recognize handwritten digits, the same problem arises. Given an example of a 5, the computer needs to adapt its model for what makes a good 5. But in many cases, the system is not told which digit a given training example exemplifies: it is often impractical for all the digits in a huge training corpus to be hand labeled as to what category they belong to, so the computer is forced to learn which digits *are* 5s at the same time as learning what makes a *good* 5. The computer-learner cannot improve its model of what makes a well-formed 5 until it knows when it is seeing a 5, but it cannot know when it has seen a 5 until it knows what makes a well-formed 5.

This problem has been extensively studied in the learning theory literature (often under the label *unsupervised learning*; e.g., Hinton 1989). Much of the work has addressed automatic speech recognition, mostly under the name *Hidden Markov Models* (Baum and Petrie 1966; Bahl, Jelinek, and Mercer 1983; Brown et al. 1990). These speech systems are simultaneously learning (1) when the acoustic data they are "hearing" is an example of, say, a (hidden) phone [f], and (2) what makes for a good [f].

This problem has been successfully addressed, in theory and practice. The necessary formalization is approximately as follows. A parameterized system (e.g., a neural network) is assumed that, given the values of hidden variables, produces the probabilities that overt variables will have various values: this is the *model* of the relation between hidden and overt variables. (As we will see shortly, this model corresponds to the grammar in our problem.) Given a hidden [f] in a sequence of phones, such a model would specify the probabilities of different acoustic values in the portion of the acoustic stream corresponding to the hidden [f]. The learning system needs to learn the correct model parameters so that hidden [f]s will be associated with the correct acoustic values, at the same time it is learning to classify all acoustic tokens of [f]s as being of type [f]. The general problem is usually formalized along the lines indicated in (1.5).

(1.5) Problem of Learning Hidden Structure

Given: A set of overt learning data (e.g., acoustic data) and a parameterized model that relates overt information to hidden structure (e.g., phones)

Find: A set of model parameters such that the hidden structure assigned to the data by the model makes the overt data most probable (this model "best explains" the data)

There is a class of successful algorithms for solving this problem, the most important of which is the *Expectation Maximization* (EM) algorithm (Dempster, Laird, and Rubin 1977; for recent tutorial introductions, see Nádas and Mercer 1996, Smolensky 1996c). The basic idea common to this class of algorithms, which we will call *iterative model–based learning algorithms*, is characterized in highly general terms in (1.6).

(1.6) Iterative model–based solution to the Problem of Learning Hidden Structure

Adopt some initial model of the relation between hidden and overt structure; this can be a random set of parameter values, or a more informed initial guess.
Step 1: Given this initial model, and given some overt learning data, find the hidden structure that makes the observed data most probable according to the model.[5] Hypothesizing this hidden structure provides the best explanation of the overt data, assuming the current (initially poor) model. This first step of the algorithm is performed on all the available data.
Step 2: Now that we have deduced some hidden structure (initially incorrect), we use it to improve our model, in the second step of the algorithm. Since all the overt (acoustic) data have been connected to corresponding hidden (phonemic) structure, we can now improve the model, changing its parameters so that the imputed hidden structure optimally predicts the actual overt structure observed. (For example, the model for hidden phone [f] is changed so that it now predicts as closely as possible the actual acoustic values in the data that have been identified as instances of [f].)
Now that the model has been changed, it will assign different (generally more correct) hidden structure to the original overt data. So the algorithm goes back through the data and executes step 1 over again, reassigning hidden structure.
This new assignment of hidden structure permits step 2 to be repeated, leading to a new (generally improved) model. And so the algorithm executes steps 1 and 2 repeatedly.

This is summarized in row (a) of table 1.1.[6]

In various formalizations, iterative model–based algorithms have been shown to converge to a model that is in some sense optimal. In practice, convergence often occurs rather quickly, even with a relatively poor initial model. The key to constructing a successful iterative algorithm is combining correct solutions to the two subproblems addressed in steps 1 and 2. Crucially, *correct* here means finding the correct solution to one subproblem, assuming that the other subproblem has been correctly solved. This is summarized in (1.7).

(1.7) Correctness criteria for solutions of iterative model–based subproblems

For step 1: Given the correct *model* of overt/hidden relations, correctly compute the hidden structure that is *most probable* when paired with the overt data.
For step 2: Given the correct hidden structure, correctly compute the *model* that makes the given pairing of overt and hidden structure *most probable*.

The iterative model–based approach to learning can be connected directly with OT with the mediation of a piece of neural network theory called *Harmony Theory* (Smolensky 1983, 1986). In Harmony Theory, the well-formedness of a representation in a neural network is numerically measured by its Harmony value, and the probability of a representation is governed by its Harmony: the greater the Harmony, the higher the

Table 1.1
Iterative model-based learning algorithms

Framework		Iterative Solution Steps	
		Step 1 **Find the hidden structure…**	**Step 2** **Find…**
(a)	Iterative Model	… that is most probable when paired with the overt data, given the current model.	… the model that makes step 1's pairing most probable.
(b)	Harmonic Grammar	… that is most harmonic (numeric) when paired with the overt data, given the current grammar.	… the grammar that makes step 1's pairing of overt and hidden structure most harmonic (numeric).
(c)	OT **RIP/CD**	… of the most harmonic (OT) structural description consistent with the overt data, given the current grammar. **Robust Interpretive Parsing**	… a grammar that makes step 1's structural description optimal. **Constraint Demotion**

probability.[7] A representation has a hidden part and an overt part, and the Harmony function provides the model that relates these two parts: given some overt structure, associating it with different hidden structures leads to different Harmony values (and hence different probabilities). In step 1 of the iterative learning algorithm (1.6), given some overt learning data we find the hidden structure that makes the overt data most probable. This means finding the hidden structure that maximizes Harmony, when associated with the given overt structure. In step 2, we use this hidden structure to change the model—that is, change the Harmony function so that the just-derived hidden/overt associations have the greatest possible Harmony.

In Harmonic Grammar (Legendre, Miyata, and Smolensky 1990a, 1990b), an application of Harmony Theory to linguistics, the overt and hidden structures are part of linguistic structural descriptions, and the model that governs the relation between overt and hidden structure is a grammar. In this context, the iterative model algorithm in table 1.1(a) becomes the Harmonic Grammar algorithm of table 1.1(b), and the correctness criteria are like those in (1.7), but with *grammar* in place of *model*, and *harmonic* in place of *probable*. In this case, *harmonic* refers to the numeric conception of Harmony used in Harmony Theory.

In OT, the Harmony of structural descriptions is computed from the grammar nonnumerically, and there is (as yet) no probabilistic interpretation of Harmony. But the learning procedure of table 1.1(b) is still perfectly well defined; it is summarized in table 1.1(c) and labeled RIP/CD, for *Robust Interpretive Parsing / Constraint Demotion*. Robust interpretive parsing (further discussed in section 1.3.3) is the procedure that will be used to perform the hidden structure assignment of step 1. Constraint Demotion (presented in chapter 3) is the procedure that will be used to perform the grammar learning of step 2.

Given some overt learning data, RIP/CD first computes the hidden structure that has maximal Harmony when combined with the overt structure. Given learning data consisting of a sequence of syllables with stresses, for example, we find the foot structure that, in combination with the given stress pattern, has maximal Harmony. Which foot structure this is depends jointly on the overt stresses and on the currently assumed grammar—the current ranking of metrical constraints. So the algorithm proceeds as follows. Start with an initial grammar (the selection of an

initial grammar is further discussed in chapter 5). In step 1 (the RIP step), use this grammar to assign (initially incorrect) hidden structure to the overt learning data by maximizing Harmony. In step 2 (the CD step), use this hidden structure to learn a new grammar, one in which each combined hidden/overt structure of the currently analyzed data has higher Harmony than all its competitors. With this improved grammar, return to step 1 and repeat.

The prospects of success for this algorithm are supported by the fact that the analogous "correctness" criteria are met. When translated from the probabilistic framework into the OT framework, the correctness criteria given in (1.7) become those stated in (1.8).

(1.8) Correctness criteria for solutions to the subproblems under OT

For step 1, robust interpretive parsing: Given the correct *grammar* of overt/hidden relations, correctly compute the hidden structure that is *most harmonic* when paired with the overt data.
For step 2, grammar learning: Given the correct hidden structure, correctly compute the *grammar* that makes the given pairing of overt and hidden structure *optimal*.

Procedures for performing robust interpretive parsing are discussed in section 1.3.3 (general parsing with OT grammars is discussed at greater length in chapter 8). The Constraint Demotion algorithm for grammar learning is presented and discussed at length in chapter 3. The correctness of the Constraint Demotion algorithm (i.e., that it satisfies the second criterion specified in 1.8) is a theorem; the full proofs are given in chapter 7. The performance of the overall RIP/CD algorithm is explored in chapter 4, where results are presented for simulations applying this algorithm to the learning of metrical stress. The results presented in this book are from the latest and most extensive simulations of this algorithm. For the results of earlier studies, see Tesar 1997, 1998b.

1.3.3 Remarks on Parsing

Step 1 of our problem decomposition, given in table 1.1(c), makes it essential that we have a parser that can use a grammar to assign hidden structure to overt forms that are not grammatical according to that very grammar: this is what we mean by *robustness*. Our problem decomposi-

tion, we can now see, imposes a seemingly paradoxical requirement. An overt form will be informative (allow the learner to improve the grammar) if the current grammar (incorrectly) declares it to be ungrammatical. Step 1 of the RIP/CD algorithm requires that we use our current (incorrect) grammar to parse this input (assign it hidden structure), even though the grammar declares it ill formed. For many formal grammars, such an ungrammatical form is, by definition, unparsable, yet step 1 requires the grammar to parse it just the same.

OT grammars can easily cope with this demand. An OT grammar provides a harmonic ordering of all full structural descriptions, as described in (1.1). This harmonic ordering can be used in a variety of ways. The customary use is as follows: Given an input I, $Gen(I)$ is the set of all structural descriptions of I; we find the maximal-Harmony member of this set, and it is the output assigned to I. This use of the grammar corresponds to the "language generation" problem of computational linguistics, or the "language production" problem of psycholinguistics. We will call this *production-directed parsing* to contrast it with the interpretive parsing used in RIP/CD.

But, as proposed in Smolensky 1996a and developed in Tesar 1999, harmonic ordering can be used for the "language interpretation" or "language comprehension" problem as well. In this problem, we are given an overt "phonetic" form φ. The set $Int(\varphi)$ is the class of all structural descriptions with overt part equal to φ. Let us call the maximal-Harmony member of this set the *interpretive parse* assigned to φ by the grammar. Crucially for present purposes, this interpretation process makes sense even when the grammar declares φ ungrammatical (i.e., even when there is no input I for which the optimal member of $Gen(I)$ has overt form φ). An algorithm that can compute this mapping from φ to its interpretative parse is thus a robust interpretive parser capable of performing step 1 of the RIP/CD algorithm.

The most significant and general result, then, is the observation that the structure of OT grammars makes it possible to coherently define robust interpretive parsing. This definition works for any OT system. Further, the function computed by robust interpretive parsing, when given the correct grammar for a language, is the problem of "language comprehension" under OT. Thus, the assumption that robust interpretive parsing can be effectively computed is really little more than the assumption that language comprehension can be effectively computed,

an assumption most work on language learnability uncontroversially relies on.

While it is not conceptually possible to provide parsing algorithms that will work for every conceivable OT system, parsing algorithms have been developed for particular linguistically interesting classes of OT systems. For production-directed parsing, algorithms of several kinds have been developed. The production-directed parsing algorithm used in the simulations of this book comes from a class of OT parsing algorithms based on dynamic programming (Tesar 1995, 1996). Under general formal assumptions on *Gen* and *Con*, these algorithms are proved correct and efficient. The algorithms used in this book's simulations have a time complexity that is linear in the length of the input—for example, for syllabification, the amount of computation required grows linearly with the number of segments in the input. Other production-directed parsing algorithms for various classes of OT systems have also been developed (Ellison 1994, Eisner 1997, Frank and Satta 1998, Karttunen 1998).

Robust interpretive parsing algorithms have also been developed for specific classes of OT systems. The robust interpretive parsing algorithm used in the simulations of this book (Tesar 1999) is quite similar to its production-directed parsing counterpart and shares the linear computational complexity. The algorithm is a member of a class of interpretive parsing algorithms that apply to cases where the underlying form is contained within the overt form, so that hidden structure consists entirely of structural (not lexical) information.

1.3.4 Remark on Grammar Learning from Full Structural Descriptions

Given the decomposition of the learning problem developed in this section, the subproblem of grammar learning is the problem of finding a correct grammar given learning data consisting of grammatical full structural descriptions. This is the central problem solved by the Constraint Demotion algorithms developed later.

On first glance, this problem may seem trivial, since knowing the full structural descriptions provides considerable information about the grammar that is not evident in the overt data. What this first glance fails to perceive is that in OT, the grammatical principles (constraints) interact in a rich, complex way. There is nothing like a transparent mapping

from the hidden structure to the grammar: the explanatory power of OT lies precisely in the diversity of structural consequences of a constraint embedded within a hierarchy. Knowing the location of the metrical feet in a word, for example, leaves one far short of knowing the metrical grammar. An OT grammar is a collection of violable constraints, and any given foot structure will typically involve the violation of many different constraints: many language-particular OT grammars will be consistent with the given foot structure. Linguists who have actually faced the problem of deducing OT grammars from a complete set of full structural descriptions can attest to the nontriviality of solving this problem, especially in the general case. Indeed, the algorithms presented here (in chapter 3) have significant practical value for linguists working in OT. Given hypothesized structural descriptions of language data and a hypothesized set of constraints, these algorithms can quickly and easily provide a class of constraint rankings that account for the data, or directly determine that no such rankings exist.

1.4 Outline of the Book

The following is a guide to the main proposals of this book and where they may be found.

The central claim of the book, stated in (1.2), is that OT provides sufficient structure at the level of the grammatical framework itself to allow general but grammatically informed learning algorithms to be formally defined. Specifically, an algorithm is proposed in which the interdependence of grammars and structural descriptions is overcome by using successive approximation, iterating between "robust interpretive parsing" to assign structure to overt data, and grammar learning from the assigned structure. This proposal, named RIP/CD for robust interpretive parsing / Constraint Demotion, was introduced in section 1.3.

RIP/CD relies heavily on the structure of OT. An overview of OT, including illustrations with OT analyses of syllable structure and clausal subject distribution, is presented in chapter 2.

RIP/CD employs a decomposition of learning into two central subproblems. The first subproblem is that of assigning a structural description to an overt linguistic form given a grammar that may not be correct. This is the computation named robust interpretive parsing. Section 1.3.3 showed how this problem may be formally characterized within OT as

optimization over a space of candidates all of which match the overt linguistic form. Concrete algorithms for computing robust interpretive parsing are discussed in section 8.5 of chapter 8, devoted to parsing algorithms for classes of OT grammars.

The second subproblem of RIP/CD is the learning of a constraint ranking from a set of full structural descriptions. This problem is solved by a family of algorithms based on the principle of Constraint Demotion. This principle states that constraints violated by grammatical structural descriptions must be demoted (in the ranking) below constraints violated by competing structural descriptions. Constraint Demotion is presented in chapter 3, where it is illustrated and discussed.

Constraint Demotion has two important formal properties. First, it is guaranteed to learn a correct ranking from an adequate data set. Second, there is a strict bound on the amount of data needed to form an adequate data set: Constraint Demotion will never need more than $N(N - 1)$ informative examples to correctly determine the grammar (where N is the number of constraints). Formal proofs of these results are given in chapter 7.

Constraint demotion reranks constraints based on the relative constraint violation patterns of structural descriptions of (1) grammatical forms, and (2) some competing forms. It thus depends on an ability to efficiently compute (1) structural descriptions of overt learning data, and (2) informative competing structural descriptions. Computation of the first is achieved by robust interpretive parsing, as discussed earlier. Computation of the second, informative competitors, is achieved by production-directed parsing, the very same computational procedure at work in language production. The use of production-directed parsing in learning is discussed in section 3.3; algorithms for performing production-directed parsing are presented in an chapter 8.

Given concrete proposals for solving the two subproblems, it is possible to evaluate RIP/CD, the strategy of iterating between structure assignment and ranking adjustment. Such an evaluation is conducted here through a series of experiments using a computer implementation of RIP/CD applied to an OT system for metrical stress. Many of the overt forms in the languages of this system have a nontrivial degree of ambiguity—the same overt form is consistent with several different possible structural descriptions—so this is a meaningful test. The experimental results are presented and discussed in chapter 4.

Given a credible approach to learning grammars by unraveling the basic interdependence between structural descriptions and constraint rankings, the possibility of multiple grammars consistent with the same data may be raised. In particular, the familiar issue of subset relations among different languages can be raised: Can the learner be constrained so as to always select the smallest language consistent with the positive data presented? This question is briefly discussed in section 5.1, along with a proposed solution: set the initial state of the learner to a ranking in which all markedness constraints dominate all faithfulness constraints.

One key component of the language learning problem that remains is the language-specific inventory of lexical underlying forms, which clearly must also be learned. The problem is made challenging by an interdependence quite similar to that addressed by RIP/CD: the actual form of the lexical entries is dependent on the constraint ranking, and vice versa. Section 5.2 discusses the prospects for extending the same iterative strategy embodied by RIP/CD to include the simultaneous learning of rankings and lexical underlying forms.

Chapter 6 revisits the larger issue of the relationship between learnability and linguistic theory, the issue first discussed in section 1.2. This chapter discusses the observation that the approach to language learning proposed in this book is not at all neutral with respect to linguistic theory: it is highly specific to OT. Further, this approach to learning actually thrives when substantive universal principles interact strongly in the determination of linguistic patterns, a property hardly universal among language learning proposals. The consequence is that the demands of linguistic explanation and the requirements of language learnability converge and are mutually supportive. We take this convergence as evidence that OT, and RIP/CD, are on the right track.