LANGUAGE ACQUISITION, 2(2), 83–149

The Selection of Syntactic Knowledge

Robin Clark
Department of Linguistics
University of Geneva

Most recent approaches to language learnability and acquisition have assumed that parameter setting is largely a deductive process. This article develops the thesis that parameter setting is correctly viewed as nondeductive. In particular, deductive approaches can be computationally costly and, in the worst case, are equal in cost to a brute enumerative search through the hypothesis space. The approach developed here uses natural selection, as simulated by a genetic algorithm, to simulate parameter setting. A method is developed for evaluating the behavior of parsing devices relative to an environment (the input text), translating between parsing devices and a genome (a hypothesis string), and combining hypotheses via mating and mutation. A learner based on such a system will eventually arrive at the grammar for the least language compatible with its environment. We discuss three basic learnability properties that must characterize the learner's linguistic environment. Finally, we develop some recommendations for modeling real language acquisition.

1. INTRODUCTION

A central problem faced by the formal study of language learnability can be framed in terms of how a learning device with finite computational capacities can determine the relevance of input data to its hypotheses. Put another way, a complete theory of language learnability must specify how the learner uses the input data (and its errors on the input data) to drive hypothesis formation. Learning theory quite generally must deal with the problem of how an organism, when placed in an environment, learns to cope appropriately with its world. The environment will be rich in information and the organism will be bombarded with a constant flow of sensory input. There will be potentially no end to the number of generalizations available to the learner, some of which will be of survival value, and some

Requests for reprints should be sent to Robin Clark, Department of Linguistics, 619 Williams Hall, University of Pennsylvania, Philadelphia, PA 19104–6305.

of which will be mere noise. How can the learner utilize this input in such a way as to make good hypotheses about how the world works? One may legitimately wonder whether it is possible to devise an algorithm that can decide whether a given generalization is of value to the learner or not. This question can be taken as fundamental to research in machine learning (Carbonell, 1990; Michalsky, Carbonell, & Mitchell, 1983, 1986). The ability to sort through the input quickly and efficiently will be crucial to the organism's survival; taking too long to formulate a good hypothesis may well be just as bad as not formulating the hypothesis at all.

A fruitful approach to the stated problem has been to assume that the learner comes equipped to make the right sorts of generalizations at the start. In some sense, the learner knows what to expect of the world as soon as he or she arrives.[1] In this view, learners need not sort through the entire range of possible generalizations about the world, trying to distinguish the useful from the noise. They will make useful generalizations because they are built to do so from the start; they know from the beginning what to look for and how to find it. Considered from another perspective, the nervous system allows for a certain degree of plasticity (Edelman, 1987), although its general structure does not vary arbitrarily across individuals. Learning theory can be taken as abstractly specifying the points of variability and how they interact with the overall organization of the organism to account for how behavior varies as a function of experience. It should be remarked, however, that too much "prewiring" may be just as bad for the organism as too little. If an organism without innate knowledge is too slow to learn, an organism that is too heavily endowed with innate knowledge about its world will be inflexible. A change in the environment, some unexpected event, could be fatal. In a dynamic environment, rigid dogmatism is not a particularly adaptive trait.

A good compromise between speed and flexibility is to allow innate principles to be shaped in particular, definite ways by the environment. The learner expects that his or her environment will generally obey certain principles, but the exact operation of these principles will be influenced by variation along certain fixed dimensions. These domains of variation, *parameters*, are finite in number and scope, but they will allow the learner a fair degree of flexibility without slowing learning beyond acceptable limits. The learner knows the broad outlines of how things work but needs to fill in the details in a few places. Nevertheless, we might still inquire as to how the stream of sensory data causes the learner to set parameters one way and not some other way. What is it that connects input data to parameter setting?

It is tempting to view the principles and parameters (P&P) approach

[1]See the elegant discussion of this point in Mehler and Dupoux (1990).

(Chomsky, 1981b) to the study of grammar as having elegantly solved the logical problem of language acquisition. It certainly represents a major conceptual advance over earlier models of language learning (e.g., Chomsky, 1965). According to this approach, universal grammar consists of a set of grammatical principles whose behavior can be modified by fixing the values for a finite set of grammatical parameters. As each of this finite set of parameters has only a finite set of possible values, the number of possible grammars that the learner must consider is finite. One can determine the size of the learner's hypothesis space simply by multiplying out the number of parameter values. In the worst case, it would appear that the learner could simply enumerate the possible grammars and, according to a small set of criteria, find that grammar that best fits the sentences encountered in their linguistic environment. I hope to demonstrate, in this article, that more must be said about the properties of learners than that they come equipped with a finite set of principles and parameters.

Finite problems are not the same as tractable problems. Consider, for example, the possibility that universal grammar contains 30 binary parameters, an apparently modest number. In this case, the space, all else being equal, consists of 2^{30} or 1,073,741,824 possible grammars. Suppose that a machine were built that could instantiate all the parameters and print out the resulting grammar at the rate of one grammar per second. It would take the machine over 34 years of constant printing to produce all 1,073,741,824 grammars. In the worst case, a learner who could test grammars at the same rate as the machine prints them will be rather old before hitting upon the target grammar.

It is unlikely, of course, that real learners tediously enumerate the set of possible grammars, especially in light of the fact that 34-year-olds have generally long ago learned their mother tongue. A more appealing view is that learners play a game of "20 questions" with their linguistic environment. In other words, learners translate each parameter value into a *yes–no* question of the form "Does my language have the property represented by this parameter value?" and then pose the resulting question to their linguistic environment. If learners encounter a sentence of the appropriate form (a *trigger*) in the input text, the answer is *yes;* otherwise, the answer is *no*. By posing questions in the appropriate order (the subset principle; see Berwick, 1985), the learner can rely on these *yes* answers to converge on the target parameter settings. I argue here that the linguistic environment often answers with a vague "perhaps" rather than with an unambiguous *yes* or *no*. A given sentence that the learner encounters might be a positive answer for conflicting parameter settings; the learner might then have difficulty interpreting the relevance of the responses given to his or her questions by the linguistic environment.

Following much research in the language learnability literature (see

Osherson, Stob, & Weinstein, 1986; Wexler & Culicover, 1980). I make a number of assumptions about properties of input data presented to the learner during language acquisition. In particular, I assume that children are faced with input that consists, by and large, of simple, grammatical sentences (Morgan, 1986; Newport, 1976; Wexler & Culicover, 1980) and that children have no access to systematic error correction nor are they presented with ungrammatical strings marked as such. Thus, although they have some evidence about what lies inside the target language, they have no reliable evidence about what lies outside the target. For the learner, drawing the boundary between grammatical and ungrammatical material is quite difficult. Machine learning systems have often relied on access both to positive and negative data (or, at least, explicit and systematic error correction), as a perusal of the literature quickly shows (Carbonell, 1990; Michalsky et al., 1983, 1986). In this respect, language learnability presents a singularly challenging problem. The simple, ambiguous nature of the input data and the fact that the learner is not supplied with a relatively complete structural analysis imply that the learner is richly structured. Any set of input data will be consistent with an enormous number of generalizations. Without constraints on the learning device it would remain mysterious as to why learners exposed to the same target language in different linguistic environments achieve comparable grammatical generalizations (see Wexler & Culicover, 1980, where this point is made quite forcefully). These assumptions lead to a very restrictive theory of learnability; if we can present a tractable algorithm for the acquisition of syntactic knowledge under these assumptions, then we stand a good chance of arriving at a precise theory of human language acquisition.

A formal theory of language learnability must explain how the learner is structured in such a way as to exploit input data of an equivocal nature. I argue that the learner uses a form of natural selection (Darwin, 1859) to decide between competing analyses. In brief, the learner's hypotheses are represented as a population of parsing devices. This population is evaluated against the input text by a fitness metric. The most fit of the parsing devices "breed" to create a new generation of parsing devices. The succeeding generations of parsing devices will tend to inherit those parameter settings that are most fit relative to the linguistic environment. Differential breeding of fit hypotheses, plus elimination of unfit hypotheses, will drive the learner toward the target. To this end, I formalize the learner as a *genetic algorithm* (GA; Goldberg, 1989; Holland, 1975) that represents hypotheses as strings that can be combined by a small set of genetic operators.

In section 2, I lay out some arguments in favor of a deductive theory of parameter setting. In brief, such models attempt to exploit the inherently deductive structure of a P&P grammar to guide the learner. This is intended to improve upon the lack of efficiency associated with enumerative learning

schemes, and to help the learner to avoid potential traps in the hypothesis space (in particular, superset languages). The first part of section 2 is devoted to sketching such a learner and considering some types of deductive processes that are associated with P&P grammars. I then argue that, precisely because of the deductive interactions in the grammar, parameter setting via deduction is no more efficient than an enumerative search and may potentially be more costly.

In section 3, I discuss the core mechanisms of a genetic algorithm, show how parameter setting may be simulated by such an algorithm, and develop a metric of fitness relative to a population of hypotheses and an input text. Three basic properties of the learner are derived from the fitness metric that are important to studying the learnability properties of parameterized systems of grammar. These three properties ensure that the learner will arrive at those parameter settings that minimize the number of grammatical violations on the input text, minimize the size of the grammar, and minimize the complexity of representations assigned to the input text. Crucially, this learner is nondeductive and thus circumvents the objection raised in section 2.

Section 4 turns to formalizing evaluations of system performance and convergence to a sequence of parameter settings. Three learnability properties that must hold for the input sequence and a parameterized system are developed.

Finally, section 5 considers, first, the problem of studying developmental sequences in a nondeductive approach and, second, the relationship between the present model and biological models of language processing. A case study, acquisition of verb second structures, forms the core part of this section; I argue that (some) developmental sequences follow from independent properties of the input evidence combined with the properties of the fitness metric, proven in section 3, and the learnability properties discussed in section 4.

2. PARAMETERS AND GRAMMATICAL THEORY

In this section, I turn to a general characterization of the P&P approach to grammar and consider some of its consequences for learning theory. I am particularly concerned with the deductive structure of a P&P-style theory and whether or not this deductive structure aids the learner in converging on the target parameter settings. I will argue that, despite the a priori appeal of the deductive approach, the complexity of the task faced by the learner is such that the learner cannot afford to rely on purely deductive methods. Instead, the effects of deductions must be simulated by the learner in a computationally inexpensive manner.

As noted briefly earlier, a P&P approach hypothesizes that universal grammar (UG) is divided into a fixed core of principles and a finite set of parameters that modulate the functioning of the principles. All languages are organized around these core principles, but particular languages differ in systematic ways due to the effects of different parameter settings. Intuitively, one can imagine UG as a computer to which one must add a certain number of special circuit boards (the parameters) to adapt its operation to the purposes at hand. The learner must determine which circuit boards to add to fulfill the task (efficiently parse the target language); the learner does so by selecting circuits and testing the resulting machine against an input text. I endeavor to make the notion of a self-constructing parsing device more precise later.

As a first approximation, let us take a particular example of restricted parametric variation (discussed in Wexler & Manzini, 1987). Virtually all languages have elements that act as anaphors, items that must be *bound by* an element within some syntactic domain.[2] An example of such an element is *himself,* which must corefer with another element in a sentence and must find this element within a restricted syntactic domain (roughly, a tensed clause).

(1) a. Mary thinks that [John$_i$ kicked himself$_i$]
b. *John$_i$ thinks that [Mary kicked himself$_i$]

In (1b), the anaphor *himself* is separated from its antecedent *John* by too large a syntactic domain; *John* occurs outside the governing category (GC) of the anaphor. Because *Mary* is not a potential antecedent for *himself,* the sentence is ruled out as ungrammatical. Languages vary, however, as to whether or not they admit long distance anaphora; that is, an item that is obligatorily bound but may be bound by an antecedent outside of its local domain. Let us assume that the presence of long distance anaphora is a parameter within UG. English, for one, does not allow this type of anaphora, whereas Korean, Japanese, and Icelandic do (Anderson, 1986; Yang, 1983). If such anaphors are allowed, they exhibit systematic regularities. In some languages, the antecedent for a long distance anaphor must be a subject. Japanese is such a case of a language that has subject-oriented long distance anaphora. Thus, whether or not a long distance anaphor must be bound by a subject is a further parameter. In addition, some languages

[2]Pica (1984) hypothesized that *long distance* anaphors are necessarily monomorphemic. If this is correct, then the learner can distinguish between short distance and long distance anaphors without setting a parameter. That is, the learner would simply note whether the item was monomorphemic. The parameters discussed here, as is the case throughout article, are intended as illustrations of how the logic of parameters and parameter setting works generally. Hence, no great weight should be given to particulars of the analysis.

define syntactic domains (relative to the tense/aspect system of the language) within which the long distance anaphor must be bound, thus yielding further parameters.

Crucially, however, languages do not vary arbitrarily with respect to long distance anaphora. The anaphor, long distance or not, must stand in the c-command relation with its antecedent. The set of syntactic domains within which an anaphor must find its antecedent is also a subset of the possible domains within which an anaphor could be embedded. Although UG admits a certain amount of variability, not all the logically possible systems are admitted by the network of principles and parameters. Thus, parameters specify finite vectors along which languages may vary; in a sense, they have the effect of regulating the functioning of the core principles, and they place an upper bound on the degree of variability among natural languages.

As noted, above, one possible approach to the learnability problem would be to imagine that the learner plays a sort of game of "20 questions" with the input data. Imagine that the learner has the following set of parameters:

(2) a. The language allows long distance anaphors. {*yes, no*}
    b. Long distance anaphors must be bound by a subject. {*yes, no*}
    c. The domain in which a long distance anaphor must be bound is defined by indicative tense. {*yes, no*}

The learner assumes that these parameters have certain *default* values; (2a) would have *no* as its default value, whereas (2b) and (2c) would both have *yes* as theirs.[3] The learner would then attend to the stream of input data and attempt to give a "yes" or "no" value to each of the three propositions. If the learner's parsing device works appropriately for the input text, the learner would change nothing about his or her hypothesis. On the other hand, if the learner's parsing device failed to parse, he or she would select a different parameter setting (a new circuit board in our imaginary machine) and so alter the hypothesis about the target grammar.

This scenario supposes that learning is *error driven* in the sense of Wexler and Culicover (1980). Error-driven learning requires that the learner change an hypothesis just in case there is evidence that the hypothesis is incorrect. This will occur when the learner's parsing device cannot associate a

[3]Logically, given that (2a) is answered *no*, it does not matter what values the other two parameters have. The learner can truthfully attribute any properties to nonexistent long distance anaphors. If, however, the learner should be forced to change his or her mind with respect to the setting for (2a), the settings of the other two parameters will matter very much; thus, we can assume here that all parameters have default values, even if these default settings are never reflected in the language.

well-formed representation to some input item. Imagine the case where a learner is exposed to a language with long distance anaphora. Our learning scenario predicts that the learner will begin with the assumption that the target has only short distance anaphors. Thus, there will be some context in which the learner encounters the following:

(3) $[_{GC1}\ NP_1 \ldots [_{GC2}\ NP_2 \ldots$ anaphor]]

where NP_1 is the only plausible antecedent for the anaphor. As NP_1 is not contained in GC_2, the governing category for the anaphor, the learner has direct evidence that his or her assumption about anaphora in the target is incorrect. The learner can then reset the value of (2a) to the value "yes." Similar scenarios for error-driven learning can be established for (2b–c). Thus, it appears that the small system of parameters in (2) is learnable from a positive-only input text using an error-driven learner, provided that the learner can exploit the pragmatic context well enough to establish a set of plausible antecedents for at least some instances of anaphora encountered.

One way to formalize the 20 questions game is to require that each parameter comes with a description of a trigger; that is, an abstract description of a syntactic structure that is decisive for setting the parameter to that particular value.[4] When the learner encounters a new input sentence, he or she would scan the set of parameters to see if the current item matches a trigger associated with some value. If so, the learner would set the parameter to the relevant value. If not, the learner would simply continue on to the next datum. Implicit in this approach is the idea that the learner may only modify his or her grammar gradually. One might require, for example, that the learner can only change one parameter per error. I make this assumption explicit in the following way:

(4) *The Single-Value Constraint:*
Assume that the sequence $(h_0, h_1, \ldots h_n)$ is the successive series of hypotheses proposed by the learner ϕ, where h_0 is the initial hypothesis and h_n is the target parameter setting. Then, h_i differs from h_{i-1} by the value of at most one parameter for $i > 0$.

[4] I ignore here the problem of how such a description would be phrased. For example, the long distance anaphora parameter would be set to *yes* if the learner encountered an input item of the form:

(i) . . . antecedent . . . $[_{GC}$ . . . anaphor . . .] . . .

where *GC* is the governing category for the anaphor. The description language for parameters would have to be such that irrelevent details (e.g., word order in this case) were ignored. As we see shortly, this account does not provide a tractable model of the learnability problem.

The Single-Value Constraint encodes the notion that language learning is gradual by specifying that each successive hypothesis made by the learner differs from the previous hypothesis by the value of at most one parameter. Combined with the error driven learning hypothesis, it permits at most one change in hypothesis per error. The Single-Value Constraint corresponds to the constraint that Berwick (1985) placed on his learning program to the effect that the module that proposes new grammatical rules cannot call itself recursively. This stance entails that if the learner's current hypothesis differs from the target parameter setting by more than one parameter value, then there is no single datum that will allow the learner to converge to the correct hypothesis. The minimal length of the input text required for the learner to lock onto the correct hypothesis is a function of the number of parameters incorrectly set in the learner's current hypothesis, as error driven learning and the Single-Value Constraint restrict the learner to making a minimal change in current hypothesis just in case the learner cannot assign a representation to some datum.[5]

Arguably, the Single-Value Constraint vastly reduces the hypothesis space that the learner must consider at any given step in the acquisition process. In a system where the learner could reset any number of parameters, the hypothesis space would be the entire set of languages allowed by UG less the current hypothesis. In a system that obeys the Single-Value Constraint, the number of hypotheses that the learner must entertain is reduced to the number of parameters. The constraint significantly reduces the burden placed on the learner because a vastly smaller number of potential hypotheses would need to be sifted through at any given step in the procedure. In the case at hand, the learner, when faced with an error involving the binding theory, would consider as a next hypothesis one that differs from the current hypothesis by at most one setting. This reduces the set of possible next hypotheses from 6 (= 3!) to 3. Such savings would become considerable in a large hypothesis space.

Notice that we must already add something beyond UG in order to give a learnability account of the fragment under discussion. In particular, suppose that the learner is exposed to the following sentence in a context where *John* and *him* clearly corefer:

(5) John$_i$ wants [Bill to kick him$_i$]

[5]See also Dresher and Kaye (1990) on deterministic learning. Their learner clearly manifests the Single-Value Constraint. Nyberg (1991) provided a good critique of their model (as well as of the model developed in section 3). The present model abandons the Single-Value Constraint; instead, the incremental nature of convergence is modeled via take over of a parameter setting within a population.

Given that *John* and *him* corefer in the example, the learner is not licensed to conclude that English has long distance anaphora. Although *him* is bound in this particular case, the learner will be exposed to plenty of examples where it is free and, therefore, not an anaphor. In order to avoid this trap, the learner must have some means, most probably statistical, of comparing the usage of items across utterances, if only to not make foolhardy generalizations on the basis of too little evidence. This mechanism would tell the learner that (5) is irrelevant to the question of whether or not to set (2a) to *yes* because *him* distributes as a pronoun, not an anaphor. To be realistic, an error-driven learner must be equipped with such a device if the learner is to be immune from missetting parameters due to "accidents" in the input text. The learner must, at very least, be immune from reasonable amounts of noise in the input text. It would be unfortunate if one changed one's grammar due to speech errors, for example. The learner described in Kazman (1991), for example, uses a notion of confidence in a hypothesis to circumvent both noise and misleading generalizations that are possible when distributions are ignored.

Example (5) is a simple case where the learner must have a means of deciding whether or not a particular bit of evidence is relevant or irrelevant to setting a particular parameter. The given account seems plausible enough; surely the learner will need to be able to discover properties of words, and there must be some means of comparing the various uses of a word in a variety of contexts. Having encountered *him* in contexts where it is free within the sentence, the learner will have spotted it as a pronoun and will not even consider setting a parameter regulating the distribution of anaphora on the basis of such an example.

There are, of course, a number of different algorithms, some of them deductive, that could correctly sort nominals with respect to the binding theory. For example, the learner could rely on a markedness hierarchy to overcome the temptation to hypothesize that *him* is a long distance anaphor in (5).[6] All such accounts must be structured in such a way that, in attributing a property to a particular item, the learner can recognize a counterexample to an hypothesis. Thus, it is quite plausible to maintain that once the learner has hypothesized that *him* is a long distance anaphor, the learner is structured in such a way as to recognize examples where *him* is not bound as evidence against the hypothesis that *him* is a long distance anaphor. Such mechanisms, whether statistical or deductive, have computational costs and make empirical predictions regarding the course of acquisition. Our task is to formalize the learner so that these consequences

[6]I am grateful to an anonymous reviewer for *Language Acquisition* for pointing out several deductive schemes for sorting nominals with respect to the binding theory. See also Finer (1987) and Chien and Wexler (1990) for other such schemes.

can be considered openly. I argue later that the real costs associated with deductive algorithms rapidly escalate to unacceptable levels when nontrivial systems of parameters are considered. Thus, although one can imagine a wide variety of deductive procedures that can handle cases like (5) correctly, all the special procedures necessary to account for the wide variety of problems the learner must resolve will result in an unacceptably complex system. I propose an alternative that simulates deductive processes by exploiting the properties of populations of hypotheses; if we can live more cheaply by avoiding deductive costs, so much the better.

If we consider the properties of a P&P approach a bit more closely, we can see that the problem of relevance threatens to repeat itself in more subtle forms. Before considering this point in more detail, let us consider how P&P grammars generalize principles across construction types. Much of the appeal of this approach to language is that it has freed the study of grammar from construction-specific, language-particular rules. I argue in the next subsection that this generality entails certain costs for a deductive approach to learning. Consider, for example, a construction like subject to subject raising in English.

(6) John$_i$ seems [t_i to be late]

In the P&P view, there is no particular rule in the grammar of English that is responsible for this construction. Instead, a network of principles, parameters, and properties of lexical items interact to derive (6). θ-theory requires *John* to bear a semantic role; as a result, it must be base generated in a position that will receive such a role. Lexical properties of *seem* are such that it cannot assign a θ-role to its subject. Hence, *John* must be base generated as the subject of *to be late*. Case theory requires that *John* receive Case; in order to receive Case, it must occupy a Case position at S-Structure. But *to be late* is an infinitive and so cannot assign Case to its subject. So, *John* must move to a Case position. But movement of *John* leaves a trace that must stand in a particular structural relation, proper government, with some element in order to be licensed. In short, Case theory, θ-theory, government theory, and lexical properties "conspire" to derive (6). Such conspiracies have come to replace the less adequate notion of language-particular (and construction-specific) rule. This type of interaction, where a variety of components interact to derive the properties of grammatical constructions, has become the norm in the literature, as an examination of Chomsky (1981a, 1985, 1986) and the references cited there can attest. From the point of view of grammatical theory, the move is a salutary one because it permits a rigorous approach to the explanation of word order and relative acceptability of sentences.

A P&P grammar, then, can be viewed as a special logic for "proving" the

grammaticality of sentences. That is, the principles and their associated parameter settings can be viewed as a set of axioms that, taken together, imply the grammaticality of a set of strings. The principles and parameters of linguistic theory receive much of their appeal in that they provide each other with predictive power in the sense that they interact to derive sentences. To take but one example, the fact that English allows for Exceptional Case Marking (ECM) structures, as in (7), has a number of consequences not just for the theory of Case, but also for the binding theory, the distribution of empty categories, and the distribution of pleonastic elements. Exceptional Case Marking structures allow for the "exceptional" government of the subject of an infinitival clause by a verb higher in the parse tree. Thus, *John* in (7) is the structural subject of the infinitive *to be late.* The verb *believe,* governs and assigns Case to the subject of the infinitival. This exceptional government has a complex set of consequences for other components of the grammar.

(7) Bill believes [John to be late]

The analysis where *believe* governs and assigns Case to *John* immediately predicts the following grammaticality judgments:

(8) a. Bill$_i$ believes himself$_i$ to be late.
b. *Bill$_i$ believes him$_i$ to be late.
c. Who believes whom to be late.
d. Bill believes it to be raining.
e. Bill believes there to have been a riot.

We can contrast the case of *believe* with the superficially identical case of *persuade.* In this case, the noun phrase immediately following *persuade* is the true object of the verb and not the structural subject of the embedded infinitival. That is, *persuade* is a control verb as opposed to *believe,* which is an ECM verb. The fact that *persuade* is a control verb and not an ECM verb predicts a slightly different range of grammaticality judgments.

(9) a. Bill$_i$ persuaded himself$_i$ to be late.
b. *Bill$_i$ persuaded him$_i$ to be late.
c. Who persuaded whom to be late.
d. *Bill persuaded it to be raining.
e. *Bill persuaded there to be a riot.

Notice that (9d) and (9e) are unacceptable, as opposed to the minimally different (8d) and (8e). This is because the elements *it* and *there,* being

expletive elements, have a special grammatical status and are restricted to occurring in the position of a structural subject.

The principles, parameters, and lexical properties interact in such a way that any given property can have wide-ranging consequences throughout the grammar. In doing so, each property makes predictions across a variety of constructions and across languages, and thus the principles and parameters provide each other with predictive power. As grammatical analyses are refined, the resulting set of principle and parameters receives empirical support from this increase in predictive power.

2.1 Deduction in a Computationally Bounded System

Given the complex nature of the interactions possible in a P&P system, it should be clear that a learner cannot blindly reset parameters. He or she must, in some sense, be aware of the potential effects of setting parameters and must have some means of moving toward the target, based on experience with the input text. In effect, the learner must be able to distinguish between hypotheses in such a way as to "predict" which hypothesis will optimize the ability to represent the input sequence. It has been suggested (e.g., by Roeper & Nishigauchi, 1987) that learners accomplish this by using the logical structure of the grammar to deduce the proper set of parameter settings for the target grammar. Considered from a more directly computational perspective, however, this proposal does not seem quite so promising. To be precise, the problem lies in the fact that the move from construction-specific rules to principles and parameters has the immediate consequence that the principles are not insulated from each other but interact to derive the language.

But the fact that principles and parameters interact in diffuse and complex ways presents the learner with a new kind of problem. The learner will not necessarily have direct evidence of the influence of a single parameter on the target language. Instead, he or she will have to assume that the parameters to be set are taking part in complex conspiracies to derive the sentences being encountered. Given that the learner has made an error, there is immediate information that at least one parameter in the hypothesis grammar is set to an incorrect value, but the learner may not be able to detect which parameter is at fault.

To see one example of this, consider the problem of acquiring ECM in English.

(10) John believes [$_{IP}$ Bill to have left]

In an ECM structure, the verb has the property of being able to assign abstract Case across a clausal *(IP)* barrier. ECM structures are subject to

cross-linguistic variation; their presence in the language, therefore, is the result of a parameter that must be set by the learner. Suppose that, in the unmarked state, an embedded *IP* is always dominated by *CP*. If so, then an embedded subject is always protected from government by an element external to the *CP;* overt phonological subjects of infinitives will be ruled out by the Case Filter, which requires overt phonological *NP*s to have abstract Case, in the absence of any independent means of assigning Case internal to the infinitival *CP* (Chomsky, 1981a). In terms of the present framework, we could express the ECM parameter as:

(11) ECM: A verb may subcategorize for an infinitival *IP*. {*yes, no*}

Setting the ECM parameter would involve assigning a truth value to the statement in (11).

Suppose that the learner assigns the structure in (12) to the example in (10).

(12) John believes [$_{CP}$ [$_{IP}$ Bill to have left]]

Because *Bill* has no means of receiving Case, the Case Filter will block (12) as a well-formed structural description. Note that the learner must deduce from a violation of the Case Filter that the verb *believe* can govern, and assign abstract Case to, the embedded subject position. In other words, the learner must assign the truth value *1* to (11) and allow the verb *believe* to subcategorize for an infinitival IP.

To complicate matters, it appears that some languages—Modern Irish, for example (see Chung & McCloskey, 1987, and the references cited there)—allow the subject of infinitive clauses to receive Case structurally.

(13) Is cuimhneach leo [iad a bheith ar seachrán].
Cop mindful with-them them be (-Fin) lost
'They remember being lost.'

Thus, Case theory must be parameterized to allow for the possibility of structural Case assignment to the subject position of nonfinite clauses. The relevant parameter, which I refer to as SCM (Structural Case Marking), could be written as:

(14) SCM: The language allows subjects to receive abstract Case by virtue of their structural position. {*yes, no*}

The question now arises of how the learner can distinguish between ECM and SCM, given the presence of an example like (15) in the input text.

(15) John considers Bill to be smart.

Notice that either hypothesis will account for (15). If ECM is involved, then the verb *consider* assigns abstract Case to the noun phrase *Bill*. If SCM is involved, then *Bill* receives abstract Case by virtue of its position as subject of *to be smart*. There is no property specific to (15) that will decide the issue for the learner. The choice of parameters will have a range of consequences. First, the distribution of overt subjects will be broader in a language with an SCM Case marking system like that of Irish than in a language with ECM. An SCM language will allow overt subjects for nonfinite clauses where the overt subject is not governed externally (see Chung & McCloskey, 1987, for discussion). A language with ECM will have a different distribution of anaphors than a language with SCM Case marking.

(16) a. They believe [each other to be ill]
b. *Shíl siad [a cheile a bheith breoite].
think (Past) they each-other be(-FIN) ill
'They thought that each other was ill.'

In an ECM language, an anaphor in the embedded subject position may have an antecedent in the superordinate clause because the verb of that clause governs the anaphor. This cannot happen in an SCM language such as Irish because the verb of the superordinate clause cannot govern the embedded subject position and, hence, does not seem to influence the embedded subject's governing category.

Consider the case where the learner is exposed to English and encounters a sentence the analysis of which presupposes that the ECM parameter is set to *yes*. It is possible that the learner will incorrectly set the SCM parameter to *yes*. At this point, the learner can give an analysis to sentences like (15), but not the correct analysis. In particular, the learner will fail to assign a well-formed analysis to (16a). At this point, he or she can conclude either that the language involves ECM and not SCM, or that the language permits long distance anaphora, as discussed earlier, or that the language permits both ECM and SCM. The second and third alternatives, of course, would be fatal for the learner as they involve languages that are supersets of the target (see later discussion of superset languages). Notice that, by the Single-Value Constraint, the learner cannot directly set SCM to *no* and ECM to *yes*. Instead, the learner must take two steps: first backtracking by resetting SCM to *no* and then testing a new hypothesis by setting ECM to *yes*. Thus, although the Single-Value Constraint seems to reduce the learner's space of possible hypotheses, it makes the problem of scheduling trials for new hypotheses relatively more complex, particularly if the learner

is to avoid a cycle of infinitely going back and forth in a series of hypotheses.

It appears that in order to distinguish between two possible but conflicting parameter settings the learner must consider their effects vis-à-vis other components of the theory. One might imagine, then, that the learner would deduce a number of such effects, generate the relevant type of structure, and save it in a special storage unit. He or she would then monitor the input for confirmation of the prediction. Furthermore, when conflicting hypotheses are possible, the learner must be provided with protocols for scheduling hypothesis generation, in order to avoid accidentally entering a superset language.

Idealizing the situation, the learner is presented with a sentence, s_m, that the learner cannot represent. As we have seen, it is possible that s_m can be represented by a variety of different grammars with different parameter settings (different possible grammars).

(17) a. $s_m \in L_1 = L[p_i(1), p_j(0), p_k(0)]$
b. $s_m \in L_2 = L[p_i(0), p_j(1), p_k(0)]$
c. $s_m \in L_3 = L[p_i(0), p_j(0), p_k(1)]$

$L[p_i(1), p_j(0), p_k(0)]$ is interpreted as "the language that results from setting the parameters p_i to *1 (yes)*, p_j to *0 (no)*, and p_k to *0 (no)*." (17) indicates that a given example may be compatible with some set of different parameter settings, as shown in Figure 1, where the lines represent the derivation relationship between a sentence and a grammar.

Although the grammars (17) may all be capable of representing s_m, they are not equivalent. They will differ on the grammaticality predictions they make with respect to other strings, s_n, s_p, s_q, and so on.

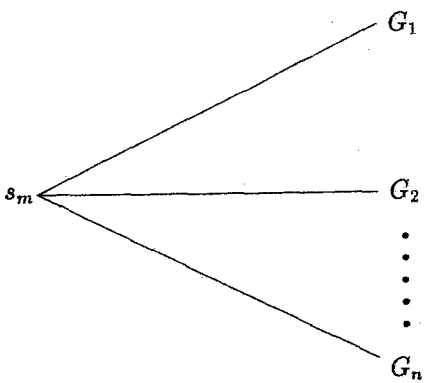

FIGURE 1 The derivation relationship between sentence and grammar.

(18) a. $s_n \in L_1 \,\&\, s_n \notin L_2 \cup L_3$
b. $s_p \in L_2 \,\&\, s_p \notin L_1 \cup L_3$
c. $s_q \in L_3 \,\&\, s_q \notin L_1 \cup L_2$

The target language can only be one of (18a–c). Thus, the learner has a stake in distinguishing among all these alternatives because, if the wrong alternative is selected, the learner has not correctly learned the target language. How are the complex interactions possible in a P&P theory untangled to find which parameter must be reset? A number of components will have interacted to derive the particular example, and each of these components may contain a number of parameters. Resetting any one of these parameters will have a cascade of effects throughout the language derived by the learner's new hypothesis grammar.

It does not seem likely that the learner can trace through this complex set of consequences and zero in on the correct parameter setting within acceptable time limits and using a suitably bounded amount of memory. More explicitly, suppose that UG contains a set of 30 binary parameters and that the learner is only concerned with interactions among parameter values. Although 30 seems like a tractable set of parameters, the learner will have to consider an enormous set of possible interactions among the parameter settings. For example, the learner will have to consider the set of consequences that follow from one parameter taken alone, plus any two parameters taken together, plus any three parameters interacting together, and so on.

The number of interactions among parameter settings that are possible a priori can be estimated more formally. Where n is the number of parameters and k is the number of possible combinations, the learner will have to consider all combinations of n parameters taken k at a time as k ranges from 1 to n or:

(19) $\Sigma_{k=1}^{n} \binom{n}{k}$

where $n = 30$. This number is equal to $2^{30} - 1$, roughly the size of the hypothesis space to begin with. It is therefore difficult to see how a deductive learner who attends to the potential interactions among parameter settings would be any better than an enumerative search through the possible grammars, a technique we rejected earlier as too costly. Notice that this number is a lower boundary, as the learner will also have to consider other potential interactions between parameter settings and fixed components of the grammar.

If such a deductive learning algorithm is to succeed, it must be provided with a wholly automatic means of distinguishing the relevant set of

interactions from the irrelevant cases; in addition, the set of relevant cases must be proven to be small enough that the learner can reasonably manage them. I show later that a learning algorithm exists that can correctly converge on the target grammar without the brittleness and computational costs associated with a deductive learner. In particular, the learner developed in Clark (1990b) can simulate the crucial deductions without actually performing them explicitly (see also Clark, 1990a).

In considering the learning problem, it is important to recall that the learner is computationally bounded; in a deep sense, the problem of learnability is directly tied to computational considerations. The learner has finite resources in terms of time and memory and must use those resources in such a way that convergence on the target is guaranteed. Empirically, a learner cannot take indefinite periods of time before converging to the target grammar, nor does a learner have a perfect memory for past sequences in the input text. In all likelihood, there is only limited work space in which to store predictions and deductions. Furthermore, the learner is given little information about the proper analysis to be accorded to the input data. He or she has only limited information about the proper structural analysis for any given datum,[7] and little to no access to input that is ill-formed with respect to the target. Given the apparent complexity of the deductive approach, it is far from clear that a tractable learning algorithm can be given that models such a learner. The question then arises of how children are able to learn their mother tongue quickly, accurately, and automatically (pathological cases aside). Given the robustness of child language acquisition, they must have some inexpensive means of unraveling the complex grammatical interactions that they encounter. The formal theory of language acquisition is obliged to explore these techniques.

2.2 Subsets and Other Traps

Beyond efficiency considerations, certain formal relationships between parameter values and, I argue, between parameters show that the learner cannot use a brute-force search technique to converge on the target. In addition, a learner who attempted to use deductions must be structured so as to avoid the traps set by these relationships. As pointed out by Berwick (1985) and Wexler and Manzini (1987), certain parameters may fall into subset relations; that is, the language that results when a certain parameter, p_x, is set to *0* is a proper subset of the language that results when p_x is set to *1*, as in (20) and Figure 2.

[7]Following Morgan (1986), I grant that intonation, for example, can provide the learner with some information about structural analyses. I maintain, however, that the proper set of parameter settings are still massively underdetermined even if the learner has access to phonological cues.

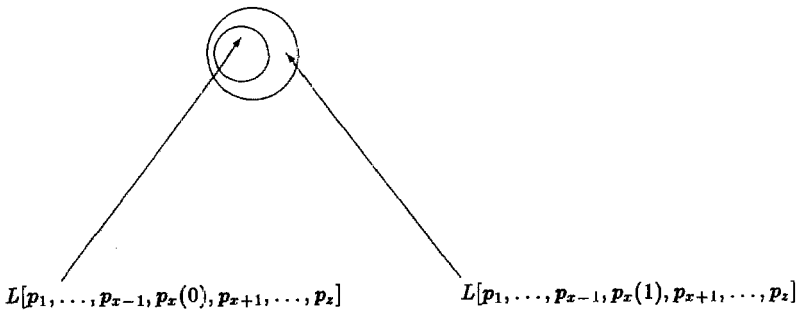

FIGURE 2 Subset parameters.

(20) *Subset Parameters*
$L[p_1, \ldots, p_{x-1}, p_x(0), p_{x+1}, \ldots, p_z] \subset L[p_1, \ldots, p_{x-1}, p_x(1), p_{x+1}, \ldots, p_x]$

This type of situation arises with the long distance anaphora parameter discussed earlier as well as with the bounding nodes for classical subjacency. Consider the two following abstract representations involving anaphora. The square brackets represent domains within which a short distance anaphor must find its antecedent.

(21) a. $[GC_1 \ldots NP_i \ldots [GC_2\ NP_j \ldots \text{anaphor}_j \ldots]]$
b. $[GC_1 \ldots NP_i \ldots [GC_2\ NP_j \ldots \text{anaphor}_i \ldots]]$

In a language such as English, which has only short distance anaphors, only (21a) will be grammatical; (21b) will be ruled out because the anaphor does not have its antecedent within its local domain. In a language such as Korean, which allows long distance anaphora, both (21a) and (21b) will be grammatical. Thus, Korean is a superset of English with respect to anaphora. This sort of relationship between parameter values occurs in several places in the grammar and sets a real trap for the learner. In essence, all the sentences that are grammatical in the subset will also be grammatical in the superset language. Thus, if the learner guesses the superset language, then no further evidence will contradict the learner's hypothesis; each new sentence will be compatible with the hypothesis and the learner will never have grounds to retract this (incorrect) hypothesis. Nevertheless, the learner has not correctly acquired the target grammar. Thus, he or she must guess the minimal language compatible with the set of input sentences encountered. Given that the learner has no reliable access to negative evidence, it appears that the learner must guess the smallest possible language compatible with the input at each step of the learning procedure. This is the content of the Subset Condition of Berwick (1985), which is intended to circumvent

the sort of trap posed by subset parameters. This again shows that in the general case the learner cannot rely on a simple search of the hypothesis space to converge on the target grammar.

A more dangerous possibility arises if we consider that sets of parameters might interact in such a way as to generate superset languages. That is, when considered individually the parameters in question may not necessarily generate superset languages, but when they act in a group they do generate a superset language. This is the *shifting* relation of Clark (1990b).

(22) *Shifting*
Two parameters, x_i and x_j, cause a shift at values $x_i(1)$ and $x_j(1)$ just in case:

a. $L[\phi_n(x_1, \ldots, x_i(1), \ldots, x_j(0), \ldots, x_n)] \not\subset$
$L[\phi_n(x_1, \ldots, x_i(0), \ldots, x_j(1), \ldots, x_n)]$

b. $L[\phi_n(x_1, \ldots, x_i(0), \ldots, x_j(1), \ldots, x_n)] \not\subset$
$L[\phi_n(x_1, \ldots, x_i(1), \ldots, x_j(0), \ldots, x_n)]$

c. $L[\phi_n(x_1, \ldots, x_i(1), \ldots, x_j(0), \ldots, x_n)] \subset$
$L[\phi_n[x_1, \ldots, x_i(1), \ldots, x_j(1), \ldots, x_n)]$

d. $L[\phi_n(x_1, \ldots, x_i(0), \ldots, x_j(1), \ldots, x_n)] \subset$
$L[\phi_n(x_1, \ldots, x_i(1), \ldots, x_j(1), \ldots, x_n)]$

In other words, a shift occurs given two parameters that generate superset languages when they are both set to some particular value. Notice, crucially, that if the language generated by setting x_i to *0* is a subset of the language generated by setting x_i to *1*, this relationship is preserved in the shifted language. In brief, a learner could obey the Subset Condition on the microscopic level (with respect to a single parameter) while violating it on the macroscopic level (due to shifting interactions between parameters). Of particular importance for learning theory, it is possible that the learner could accidentally enter a superset language due to the interactions between parameters. In order for the learner to avoid these higher level violations of the Subset Condition, he or she would have to calculate interactions between parameter settings. But this would become increasingly difficult as the number of parameters that could "conspire" to generate a shifted language increased.[8]

[8]Notice that the shifting relation obeys the Independence Principle of Wexler and Manzini (1987, p. 65), "The subset relations between languages generated under different values of parameter remain constant whatever the values of the other parameters may be."

The Independence Principle is intended to prevent subset relations of a parameter from wobbling about due to the effects of other parameter settings. As can be seen from Figure 3, parameters can interact to generate shifted languages while obeying independence. With respect to any given parameter, the subset relations remain constant; it is the interaction between parameters that creates a shift.

Figure 3 illustrates the most simple case of shifting involving superset parameters. In this example, we have two parameters, p_1 and p_2, that interact to generate a shifted language, $L[p_1(1), p_2(1)]$. In Figure 3, dominance indicates the subset/superset relation. In this case, both p_1 and p_2 are superset parameters; any language with p_1 set to *0* is a subset of a language with p_1 set to *1* and any language with p_2 set to *0* is a subset of a language with p_2 set to *1*. Note that $L[p_1(1), p_2(0)]$ and $L[p_1(0), p_2(1)]$ are not in the superset relation with each other. The language $L[p_1(1), p_2(1)]$, however, properly contains the other three possible options. This type of shifting relation can easily be circumvented by the learner because both parties of the conspiracy, p_1 and p_2, are superset parameters. Thus, a learner who obeys the Subset Condition can easily sidestep this type of trap. As we see later, the learner will be reluctant to posit the language $L[p_1(1), p_2(1)]$ and will only do so if faced with a significant amount of empirical prodding in the form of failed parses.

A more difficult case is illustrated in Figure 4. In this case, only one of the parameters, p_1, is a superset parameter. One might imagine that p_1 regulates the option of having left-dislocation of a constituent. Left-dislocation is the process of fronting a constituent as in (23):

(23) Jean, je l'ai vu.
Jean I him-have seen
'Jean, I saw him.'

In this case, *Jean* has been placed in clause-initial position and a clitic pronoun has been left in its place.

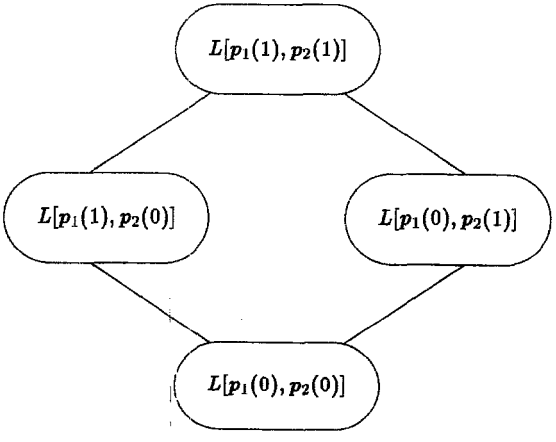

FIGURE 3 Dominance indicating the subset/superset relation.

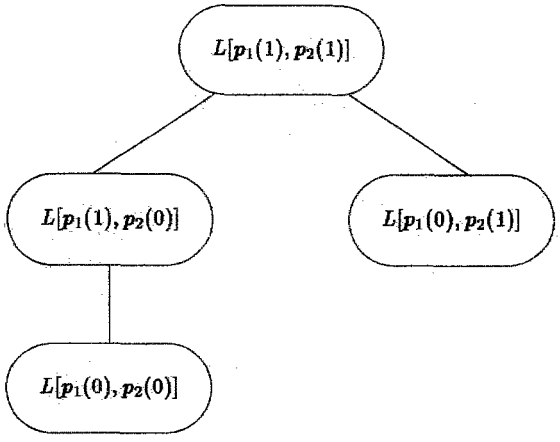

FIGURE 4 The case of one superset parameter.

The parameter, p_2, does not generate languages in the superset relation. For example, one might take p_2 to be a parameter that regulates verb-second (V2) phenomena in matrix clauses. The process of V2 places the tensed verb after the first constituent in root clauses. German is a case of a V2 language, as (24) illustrates:

(24) das Mädchen wird 20 Stunden arbeiten.
the girl will 20 hours work
'The girl will work 20 hours.'

The base word order of German is S(ubject) O(bject) V(erb). The process of V2 places the tensed verb after the first constituent. German allows nonsubjects to occur in first position, resulting in orders like:

(25) S-V-O
Adverb-V-S-O
O-V-S

Suppose, now, that p_1 and p_2 interact in such a way that, when both are set to *1*, the language allows left-dislocation of a constituent over the V2 structure of the foot clause. The resulting language has all of the normal V2 orders plus clauses with an additional constituent left-dislocated before the normal V2 order. Clark and Roberts (in press) argue that Middle French was (briefly) a language of this type. Such a language would be a shifted

language.[9] As argued there, learners tend to avoid hypothesizing a grammar that generates a shifted language, indicating that shifting is extremely marked. Thus, shifting is a case where independent parameters may interact in such a way as to have consequences for the learner.

To see the problem posed to the learner, take the case where the target language is V2 without left-dislocation. Suppose that the learner, during an early phase of the learning cycle, erroneously set p_1 to *1*, allowing left-dislocation of a constituent in response to the presence of nonsubjects in clause-initial position. This hypothesis, however, is inadequate to account for all the root V2 orders that the learner encounters. In response, the learner set p_2 to *1*, allowing for the possibility of V2, but did not reset p_1 to *0*. In this case, the learner has now entered a shifted language. Due to the interaction between p_1 and p_2, all the target orders will be consistent with the learner's hypothesis that, nevertheless, overgenerates. It must be guaranteed that such a hypothesis will be treated in such a way that the learner can retract the overgeneral hypothesis without access to direct negative evidence. But now we have returned to the problem of deducing the empirical effects of interactions between parameter settings.

Summarizing the argument so far, I have argued that a brute force enumerative search provides no solution for a formal theory of language learnability both because it is too costly in terms of time and because it provides no obvious solution to the learner's problem of avoiding fatally hypothesizing a superset of the target language. A deductive learner provides a prima facie attractive alternative because the learner is provided with some intelligence (in the form of deductive structure) that can be used to guide his or her search. Thus, a deductive learner can avoid many of the traps that undo an enumerative search. The complex interactions between parameter settings and other components of the grammar, however, mean that such a learner is faced with an imposing task if he or she is to unravel the interactions and accurately lock onto the target parameter settings. As a result of these interactions, a deductive learner faces a computational problem that is at least as costly as an enumerative search. Thus, although a deductive learner has attractive features, he or she does not represent a real improvement over an enumerative learner.

[9]The situation described in Clark and Roberts (in press) is, in fact, more complex than the text implies. The shifting relation described there relies on the interaction of three parameters: V2, left-dislocation, and the presence or absence of clitic pronouns. See Clark and Roberts (in press) for a discussion of shifting and how it drives diachronic change via acquisition. An anonymous reviewer for *Language Acquisition* pointed out that the presence of (clitic) pronouns was a clue for left dislocation as V2 does not require that the topicalized element be linked to a pronoun. Indeed, as Clark and Roberts (in press) argue, French lost V2 and retained left dislocation structures, thus avoiding the shifted language.

3. NATURAL SELECTION AND THE LEARNABILITY PROPERTY

In this section, I formulate a learning paradigm that has many of the attractive properties of deductive learning but that avoids the computational costs associated with deduction. The learner developed here will not actually perform deductions directly but will simulate them using a technique that, in its essential details, is like natural selection. This type of learner is both robust and efficient: robust in the sense that learning can be achieved in an environment containing noisy, equivocal data; efficient in the sense that the target can be located without the computational costs (or potential brittleness) associated with a deductive learner or with an enumerative search. I first turn to some general considerations regarding the behavior of a computationally bounded agent.

As noted, an apparently manageable set of 30 binary parameters would yield a hypothesis space of 2^{30} grammars for the learner to search. As we have seen, it is not practical for the learner to attempt a brute force search of a space this size, even if we could guarantee that such a search would eventually locate the target despite the traps laid by the subset problem and shifting. Similarly, the learner could not "hop" through the space at random. The probability of selecting the target at a random hop in this space would be $\frac{1}{2^{30}} (= \frac{1}{1,073,741,824})$, which yields too low a probability that the learner would succeed at any given step. Furthermore, the learner could hop onto a language that is a superset of, or shifted with respect to, the target. Given that hypothesis change is error driven, the consequences of such a careless hop would be fatal.

If the learner is to succeed at the task, there must be some way of distinguishing between a good hypothesis and a less than good hypothesis; the learner must be able to find a way of improving guesses with time until he or she systematically converges on the target. Consider, however, the early steps in the acquisition process. At this point, the learner may be quite far from the target, far enough away that resetting any one parameter may not result in a clearly visible improvement in the hypothesis. The problem of guiding the learner's hypotheses in the right direction under these conditions is far from trivial.

The picture that emerged in the preceding section is that of the learner as a bounded agent. There are only finite computational resources with which to work. The learner's memory is limited and, presumably, space for performing computations is quite limited. The learner is under time pressure to arrive at the correct grammar. Finally, the data available to the learner are limited. He or she cannot inspect all the sentences generated by the target grammar, and there is no systematic access to negative evidence. In short, the learner in some basic sense lacks the resources to make rational

decisions; there is always the chance that a bad choice will be made, whereas, had the learner only computed a little longer or waited for more information, a much better choice could have been made. Instead, the learner must have some means of simulating rationality.[10] A bounded-decision procedure attempts to make a satisfactory decision in large, complex situations where true rational decision is not feasible due to computational limitations on the agent. At any given point, the decision made by a bounded agent may not be the globally optimal decision. It will on average, however, do fairly well given its limits.

Crucially, a bounded agent is often not required to find the globally optimum solution over time; instead, it must find the best solution given its computational limitations. In the case of language learnability, we do require that, in the long run, the learner arrive at the target, which in the context of the learning problem, represents the global optimum.[11] We are then faced with an apparent contradiction. The learner has all the properties associated with a bounded agent that cannot be guaranteed necessarily to find a globally optimal solution, but the logic of the language learnability question requires that the learner find the global optimum.

The solution to the problems posed here is that the fundamental mechanism underlying language learnability is of a kind with natural selection (Darwin, 1859), and that the learnability property can be modeled using techniques drawn from population genetics.[12] Natural selection operates in such a way as to provide near-optimal solutions using a great economy of means; as such, it provides a plausible, albeit unexpected, model of the learnability property. Nevertheless, it should be emphasized that the model developed works on analogy with natural selection and is not intended as a discussion of the genetic underpinnings of the language faculty. The particular computational framework I adopt is that of GAs (see Belew, McInerney, & Schraudolph, 1990; Booker, Goldberg, & Holland, 1990; Goldberg, 1989; Holland, 1975; Schaffer, 1989; Schraudolph & Belew, 1990, for extensive discussion and a wide variety of applications of

[10]This is the important notion of *bounded rationality* or *satisficing* (see the articles collected in Simon, 1982).

[11]In fact, though, we may wish to loosen this requirement in practice. That is, it may be preferable to require that the learner arrive at a grammar near, but not identical to, the target. The issue hinges on the problem of diachronic change, how languages change over time. If language change is ultimately related to learnability, then "near misses" may provide a good model of how diachronic change works. See Clark and Roberts (in press) for an analysis of diachronic change relative to the learnability framework discussed here.

[12]Good recent discussions of evolution can be found in Dawkins (1983, 1986), where many of the properties of natural selection as an "engineering technique" are discussed. A classic work, which tied evolution to population genetics, is that of Haldane (reprinted in Haldane, 1990). Falconer (1989) and Spiess (1989) provide excellent coverage of mathematical models in population genetics.

this framework). I now turn to a formalization of the learning problem for natural language and an explicit theory of the learnability property for a P&P approach to grammar.

3.1 Genetic Algorithms and Language Learnability

As noted, parameters are finite vectors along which natural languages may vary; the learner is faced with the problem of searching a finite space of possible grammars, rather than with the more difficult problem of inducing a set of rules that lies at an undetermined point in an infinite hypothesis space. Formally, I characterize the learning problem by means of the following (see Clark, 1990b for a more detailed exposition):[13]

$$(26)\ \gamma[\phi_n(\varphi(\sigma_i))] = P_m$$

In (26), the input text is represented by σ. The input text is taken by the learner, φ, as its sole argument. The learner, φ, is a relation between input texts and parameter settings. A sequence of n parameter settings delivered by φ can be mapped by the function ϕ_n onto G_i, a grammar for the input text, σ_i. Once a grammar is delivered by ϕ_n, it can be used to determine a parser, P_m, for the grammar G_i. The relation, γ, is a "compiling" function that maps a grammar onto a parsing device compatible with that grammar. The learning problem for natural languages is, then, a relation between input texts and parsing devices. Notice that the learner, on this approach, does not directly hypothesize grammars (as was the case with Wexler & Culicover's, 1980, model of learnability). Instead, the learner produces a set of parameter settings that, as we soon see, can be represented as a simple index. The grammar results from interpreting the sequence of parameter settings produced by the learner relative to UG (ϕ_n in (26)).

Genetic algorithms mimic natural selection by representing hypotheses about a problem in a way that is similar to the way in which genetic material is represented. Hypotheses are then tested against the problem space, with the most fit hypotheses contributing to the formation of new hypotheses via reproduction (the combination of preexisting hypotheses to form new hypotheses in a way that is similar to the biological recombination of DNA present in mating). By "breeding" the most fit hypotheses, testing them

[13]The formal method of characterizing the learning problem found here is very much influenced by the work of Osherson et al. (1986).

against the problem space, and pruning the least fit, a genetic algorithm can efficiently search large spaces and find optimal solutions.[14]

A genetic algorithm consists of the following components:

(27) a. A *representation* of hypotheses in terms of *strings,* similar in structure to genetic material.
b. A *crossing-over* operator. This mechanism combines two hypotheses and produces a new hypothesis by combining parts of each to the parent's genetic material.
c. A *mutation* operator. This mechanism randomly alters an offspring's genotype to produce a new hypothesis close to, but not identical with, the parent's genetic endowment.
d. A *measure of fitness* of hypotheses in terms of their performance in an environment.
e. A *reproductive mechanism* that allows a hypothesis to produce offspring; in general, the more fit a hypothesis is, the greater the likelihood that it will reproduce.

Briefly, the algorithm will begin from a population of hypothesis strings that can be mapped onto parsing devices. These parsers will be tested against an input sentence and their performance judged by the measure of fitness. The resulting measures will be used as a raw probability for reproduction; more fit hypotheses are more likely to be selected for reproduction and so contribute more frequently to the creation of new hypotheses than less fit hypotheses. Reproduction itself blindly combines parts of old hypothesis strings to create new hypotheses. These are added to the population and the cycle is run again. In addition, I assume there is an operation of *death* that occasionally deletes a relatively unfit hypothesis from the population. Thus, unfit hypotheses are eventually purged from the pool of hypothesis strings and stand no chance of contributing to the formation of new hypotheses.[15] Repeated application of the learning cycle, plus occasional pruning of unfit hypotheses, eventually result in a sole surviving hypothesis. If this hypothesis matches the target, then the learner has successfully converged.

[14]Space prevents a comprehensive discussion of this class of algorithms. I do not explore some of the potential genetic operators, nor am I concerned with certain aspects of the optimal representation of the hypothesis space. See Goldberg (1989) for a general introduction and Clark (1990b) for an application to the learnability problem for natural languages. Holland (1975) provided a rigorous mathematical exposition of this class of algorithms.

[15]See Clark (1990b) for a more complete discussion of the algorithm. The algorithm itself has been implemented in Lisp and tested on search spaces of 2^{30} possible solutions. Chapter 3 of Clark (1990b) provides a detailed discussion of this simulation.

Most crucial for our purposes are the representations of hypotheses in terms of strings and the notion of a fitness metric. We must be capable of mapping between our problem space and a representation of that hypothesis space in terms of strings. Fitness will be measured in terms of the performance of parsers relative to a stream of input data; the actual algorithm will operate on the string representation of the hypotheses. We must, then, have a translation function that relates our hypotheses (strings) to the parsers that they represent, as in Figure 5.

We conceive of the learner, φ, as operating on strings of parameter settings. The translation function in Figure 5 then maps the learner's hypothesis strings onto parsing devices. Thus, the translation function is comparable to both the function ϕ_n, which maps sequences of parameter settings onto grammars, and the function γ, which maps grammars onto parsers. In a sense, the hypothesis strings represent genotypes for parsing devices, whereas the translation functions (ϕ_n and γ) map genotypes onto phenotypes. These are then tested against the linguistic environment for relative fitness, and the results are used as a basis for forming new hypotheses.

Let us turn to a more careful consideration of the representation of hypotheses. We saw in the previous section that parameters can be reformulated as simple propositions to which the learner can attribute a truth value (a *yes* or a *no*). Let us take a further example of a more complex parameter that appears to be difficult to express in this binary form. The bounding nodes for classical subjacency (Chomsky, 1977) provide a good example of such a parameter, because there are several possible bounding nodes. The subjacency principle regulates the formation of long distance dependencies of the type found in *wh*-questions in English. Here, subjacency is taken as an invariant property of natural languages, whereas the bounding nodes may be contingently selected from a restricted set.

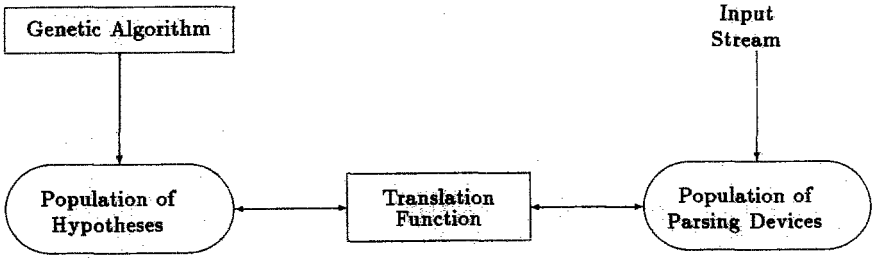

FIGURE 5 A translation function.

(28) *Subjacency*
No rule may involve X and Y in the configuration:
. . . X . . . [α . . . [β . . . Y . . . β] . . . α] . . .
(order irrelevant)
where α and β are bounding nodes;
$\alpha, \beta \in \{NP, IP, CP\}$.

IP corresponds to the traditional notion of a clause, whereas *CP* corresponds to a clause plus a subordinating conjunction. I suppose that English selects *IP* and *NP* as its bounding nodes, leading to the following set of judgments (*t* indicates the locus for a missing constituent that corresponds to the *wh*-phrase at the beginning of the sentence):

(29) a. Who did John see *t*?
b. *What did John wonder who gave *t* to Mary?
c. *What did John say why Bill wondered who gave *t* to Mary?

The examples in (29b) and (29c) are ungrammatical because displacement of the *wh*-phrase must take place in a single step across two *IP* nodes, thus violating the English version of subjacency. Italian makes a different choice of bounding nodes (Rizzi, 1982), selecting *CP* instead of *IP*. This has the effect of allowing questions of the same structure as (29b) while still excluding questions of the same structure as (29c). Thus, although both Italian and English have the subjacency property, the effects of the property in the two languages are slightly different.[16] Further, I assume that all languages must select at least one bounding node.

We can reformulate the bounding node parameter by thinking of parameters as descriptive statements that may be either true or false of a given grammatical system. From this perspective, we could rewrite the parameter for the bounding nodes as a series of three statements.

(30) a. *IP* is a bounding node for subjacency.
b. *CP* is a bounding node for subjacency.
c. *NP* is a bounding node for subjacency.

The learner's task would be to scan the input data and attempt to assign truth values, 1 for *true* and 0 for *false* to each of the three propositions in (30). The learner's hypotheses could then be taken as strings of *0*s and *1*s corresponding to the truth value associated with each parameter. For

[16]Note also that the choice of bounding nodes for subjacency is yet another case of a superset parameter. The Italian choice yields a superset of the English choice.

example, the string *100* could correspond to the hypothesis that *IP* is a bounding node for subjacency, but neither *CP* nor *NP* is. Thus, it is relatively natural to represent parameter settings in terms of strings. Notice that this provides an index, in the form of a binary number, for each logically possible system of parameters. The constraint that a language must select at least one bounding node could be captured by stipulating that the sequence *000*, representing the bounding node parameters, is fatal; that is, there are holes in the enumeration of the possible grammatical systems wherever a system without bounding nodes would occur.

Given our previous discussion, we could construct the following small hypothesis space for a learning device:

(31) a. The language allows long distance anaphors.
b. Long distance anaphors must be bound by a subject.
c. The domain in which a long distance anaphor must be bound is defined by indicative tense.
d. A verb may subcategorize for an infinitival *IP*.
e. The language allows subjects to receive abstract Case by virtue of their structural position.
f. *IP* is a bounding node for subjacency.
g. *CP* is a bounding node for subjacency.
h. *NP* is a bounding node for subjacency.

Although this space contains relatively few parameters, it defines a space of 256 logically possible grammars (assuming all other grammatical features are fixed). The hypothesis strings can be built beginning with (31a) and continuing in order to (31h). Thus, *11000010* would denote a language that allows long distance anaphora ((31a)) that are subject-oriented ((31b)) and that has *CP* as its sole bounding node for subjacency ((31g)). These strings of binary numbers can be taken as the indices which output by the learner, φ.

In terms of an actual parsing framework, there would be a fixed central algorithm, corresponding to UG. Within this algorithm would be various flags, indicating points where code must be inserted for the parser to function. The *0*s and *1*s in the hypothesis strings could be interpreted as pointers to the parameterized code. Upon receiving an hypothesis string, the machine would look up the various pieces of code indicated by the *0*s and *1*s and systematically substitute the code it finds for the flags in the main algorithm. The result would be a special parsing device designed to analyze the language enumerated by the hypothesis string. Thus, a "self-constructing" parser would be the ensemble of the core algorithm, the parameterized code, and a learning device that would select the appropriate hypothesis string in response to the input text. We then have a straightfor-

ward model of the *translation function* required by the genetic algorithm to relate hypothesis strings to parsing devices. Recall that this translation function, itself, corresponded to the functions φ and ϕ_n in the formalization of the learning problem in (26).

Having defined a representation in terms of strings (which can be mapped efficiently onto parsing devices), we can now turn to methods of combining and modifying existing hypothesis strings. The crossing-over operator combines two hypothesis strings to create new hypotheses. For example, suppose that the two hypotheses in (32) have been selected for reproduction.

(32) a. 10001110
b. 11010000

Suppose we "cut" both strings after the fourth position in the bit string.

(33) a. 1000–1110
b. 1101–0000

The first part of string (a) is then recombined with the second part of string (b), and the first part of string (b) is recombined with the second part of string (a).

(34) a. 1000–0000
b. 1101–1110

And thus two new "offspring" hypotheses that have inherited genetic material (hypotheses about settings of particular parameters) from each parent are created. It should be noted that fitness interacts in a crucial way with the crossing-over operation. Highly fit hypotheses are more likely to be selected to take part in crossing over and are therefore more likely to pass the parameter settings that made them fit on to new generations of hypotheses.

The mutation operator similarly creates new hypotheses on the basis of existing ones. In essence, it must slightly alter a hypothesis string in order to create a new, but "nearby," hypothesis. We can do this by flipping a randomly selected bit position in a hypothesis string by the following rules:

(35) a. $0 \rightarrow 1$
b. $1 \rightarrow 0$

Thus, selecting the third position of the following hypothesis for mutation would yield a "mutant" that is nearly identical to its parent structure:

(36) 10001110 → 10101110

The mutation operation can be viewed as a means of searching the immediate hypothesis space surrounding a parameter string. The fitness associated with a hypothesis is taken as the probability that it will be selected to produce a mutant offspring. Mutation itself occurs with a relatively low probability on a learning cycle. One effect of mutation is that the learner can experiment with near-optimal hypotheses that approximate, but do not correspond to, the target. This can push the learner off of a false optimum, thus distinguishing GAs from traditional hill-climbing searches in computer science (see Goldberg, 1989, for extensive discussion of this point).

3.2 The Fitness Metric

Having seen how hypotheses can be represented in terms of strings and how these can be combined systematically to form new hypotheses, there remains the problem of defining the relative fitness of a hypothesis with respect to a linguistic environment. Ultimately, we want the learner to become better able to represent the input data. Following Wexler and Culicover (1980), Pinker (1984), Berwick (1985), and many others, I adopt the notion that learning is error driven. The learner should change his or her hypothesis on the basis of evidence from the external environment, and the new hypothesis must be better able to account for this evidence. In a sense that must be made to be precise, new hypotheses should, on the average at least, be an improvement over the old ones.

In line with the characterization of the learner provided earlier, let us suppose that the crucial property of a failed parse is that it violates at least one principle of core grammar. In particular, I suppose that a parser consists of a number of grammatical modules (Case, binding, $\overline{X}$-theory, and so on) that operate in tandem to produce a full syntactic representation (Berwick, 1987; Fong, 1990; Gibson, 1991). When a principle in one of these modules is violated, then the current grammar cannot assign a well-formed representation to the input. In this case, let us assume that the offended component will signal a violation. With this in mind, we adopt the following notion of *improvement* of one hypothesis with respect to another:

(37) An hypothesis A is an improvement over an hypothesis B if, given an input datum, s_i, A signals m violations of core grammatical principles, whereas B signals n violations and $m < n$.

Intuitively, a parser that signals three violations on a parse is rather better than one that signals four violations and a parser that signals two violations

is superior to one that signals three. Crucially, parsers need not perform perfectly in order for the performance to be distinguished.

We will suppose, then, that the various modules of the parser are connected to a summation function, Σ, in the manner shown in Figure 6. Each module can signal a violation to the function Σ, which then sums up the number of violations and passes the number on to the learner. Notice that the learner has no access to which grammatical principles have been violated; he or she only receives a number representing the sum of the violations for each parse. Thus, for any given failed parse, the learner cannot look into the grammatical components to diagnose what the problem is. The core basis for comparison between two hypotheses will be whether one hypothesis triggers fewer violations than the other.

However, grammatical violations cannot be the only basis for comparing hypotheses available to the learner. As noted, the learner must be able to distinguish between hypotheses that generate a superset language and those that do not. If a superset hypothesis and a subset hypothesis can both

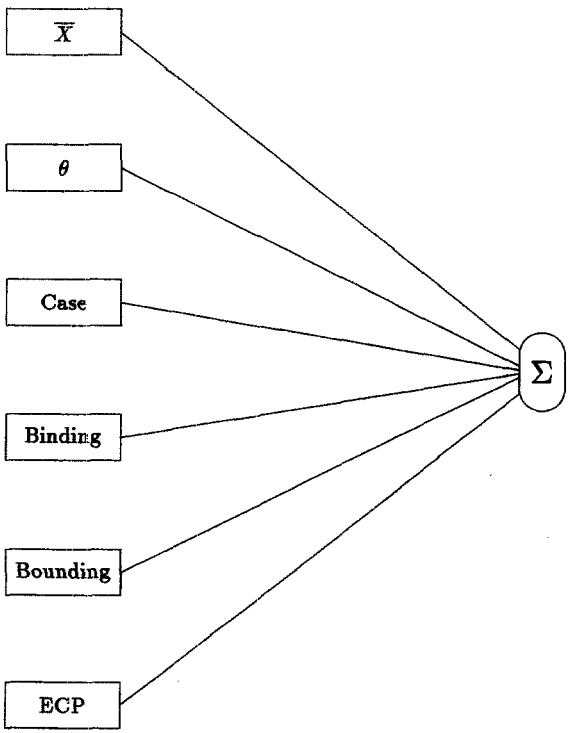

FIGURE 6 A summation function.

account for an input datum, then, all things being equal, the learner should prefer the latter to the former. Thus, any fitness metric should be such that it generally rates a subset hypothesis more highly than a superset hypothesis just so long as the subset hypothesis is empirically adequate (does not fail to parse the input data).

Finally, I assume that the learner can take into account the overall "elegance" of its hypotheses. All else being equal, the learner will prefer hypotheses that lead to more compact representations. Compactness can be defined in terms of such factors as the number of nodes required to cover the input string, the length of the chains associated with arguments and operators, or both. For the moment, we assume that the measure of elegance is a raw node count from each parse plus a measure of chain length that takes into account displaced constituents as in *wh*-questions.[17]

With these factors in mind, I have suggested the following as a fitness metric, defined over a population of parsing devices relative to an input sentence.[18] It should be noted that hypotheses are judged indirectly by means of the parsing devices they determine, just as a genotype is judged through its expression as a phenotype. I present the metric here and, later, turn to a more detailed exposition of the role that each term plays in distinguishing between hypotheses.[19]

(38) *The Fitness Metric*

$$\frac{(\Sigma_{j=1}^{n} v_j + b\,\Sigma_{j=1}^{n} s_j + c\,\Sigma_{j=1}^{n} e_j) - (v_i + bs_i + ce_i)}{(n-1)\,(\Sigma_{j=1}^{n} v_j + b\,\Sigma_{j=1}^{n} s_j + c\,\Sigma_{j=1}^{n} e_j)}$$

In (38), n is the size of the population of parsing devices. The term $\Sigma_{j=1}^{n} v_j$ represents the number of violations signaled by all the n parsing

[17]The relative elegance of grammatical representations is related to Chomsky's (1989) "economy of derivations." For the present purposes, I assume that overall economy can be estimated from S-Structure representations by counting the nodes required and measuring the lengths of the syntactic chains. An anonymous reviewer noted that, in principle, the crude node-counting metric proposed here is problematic for cases of exceptional government. For example, in the absence of θ-theory, the learner may prefer a movement analysis to a control analysis for *John wants to leave*. In the movement analysis, *John* undergoes NP movement from the embedded to the matrix clause, whereas the control analysis involves extra structure to protect the PRO in the embedded subject position from external government. Thus, the latter (correct) analysis might be disfavored relative to the former (incorrect) analysis. Fortunately for me, I can appeal to θ-theory in this case. Nevertheless, the reviewer is quite right to have questioned whether node counting does justice to the intuition that elegance and economy are related. For the present, I leave the question open.

[18]See Clark (1990b) for an earlier version of this metric and Clark and Roberts (in press) for the updated model applied to a case study of diachronic change.

[19]The fitness metric defined in (38) operates over a population of distinct hypotheses. When all hypotheses are identical (i.e., when there is only one hypothesis), then its fitness is, by convention, 1. See the later discussion of convergence and performance.

devices in the population, and the term v_i represents the number of violations signaled by parser P_i. The remaining terms serve to distinguish subset hypotheses from superset hypotheses and to favor hypotheses that lead to elegant representations. $\Sigma_{j=1}^{n} s_j$ is the number of the parameters set to superset values in the population, and s_i the number of parameters set to superset values in hypothesis h_i. Thus, the metric in (38) encodes a version of the Subset Condition. $\Sigma_{j=1}^{n} e_j$ is the measure of the general elegance of the analyses returned by the entire population, in terms of a measure of the size of each parse tree, a raw node count, whereas e_i is the measure for the parser P_i.

The metric in (38) assumes that the factors (grammatical violations, the Subset Condition, and elegance) do not carry equal import for the learner. Grammatical violations drive the learner in the most severe fashion. The constant, b, is a "superset penalty" for guessing too large a language. This term serves to scale Subset Condition violations so that they are less severe than raw failure to parse. The learner is then still able to hypothesize superset languages if necessary. Similarly, the constant c is a scaling factor for the elegance of the representation, again serving to make elegance a less important factor than grammatical violations. We leave the question of the exact values of these constants open, assuming only that $1 > b, c > 0$. The scaling constants are especially important in the early stages of learning. At this point, none of the learner's hypotheses are likely to do well relative to the input stream. As a result, the learner's major basis for distinguishing between hypotheses would be factors like the Subset Condition. This can lead the learner to rule out too hastily the possibility that the target settings involve superset values. Similar considerations hold for the elegance factor.

It is perhaps useful to consider the contribution of each of the factors, using some hypothetical examples. Let us turn first to the way in which the fitness metric treats grammatical violations. For the population, this is the term $\Sigma_{j=1}^{n} v_j$ in the fitness metric; for the individual parsing device, it is the term v_i. Suppose we have the three parsing devices, P_1, P_2, and P_3. Running these on an input sentence yields:

(39) a. P_1 returns one violation, covering the input with 15 nodes.
b. P_2 returns two violations, covering the input with 15 nodes.
c. P_3 returns three violations, covering the input with 15 nodes.

Running the results in (39) through the fitness metric gives the results in (40), with $b = 0.02$ and $c = 0.05$ (we ignore here the contribution of the subset factor by assuming that none of the hypotheses underlying the parser contains superset settings):

(40) a. P_1 receives a fitness rating of 0.393939.

b. P_2 receives a fitness rating of 0.333333.
c. P_3 receives a fitness rating of 0.272727.

Thus, parser P_1 is judged the most fit, P_2 the next most fit, and P_3 the least fit. Notice that the learner does not receive information about which grammatical principles are violated. Such information is not needed in order to distinguish between the hypotheses at hand. Instead, the learner needs only to observe the performance of the hypotheses in an external manner, without information as to their inner workings. The learner will base new hypotheses on those old ones that are relatively more fit, thus passing on the parameter settings that made those hypotheses fit to future generations. Those parameter settings that avoid grammatical violations relative to the input text will be preserved, and those that tend to generate violations will gradually disappear.

Notice, finally, that the fitness metric can correctly distinguish between the performance of various hypotheses even when none of the hypotheses performs correctly on a given input datum. The metric encodes the notion that one hypothesis, though flawed, can be less flawed than another. This is especially important in the early stages of learning when none of the learner's hypotheses is likely to resemble the target. Thus, all of his or her hypotheses will behave incorrectly. Because fitness is capable of distinguishing between incorrect hypotheses, the learner will be able to construct better hypotheses using the genetic operators associated with reproduction. With time, then, the learner's hypotheses will become more survivable.

More generally, the following theorem is implied by the fitness metric:

(41) *The Violation Theorem*
Given an input datum, s_i, and two hypotheses, h_1 and h_2, that are identical except that the parsing device that h_1 underlies returns n violations on s_i, whereas the parsing device that h_2 underlies returns m violations on s_i, then h_1 will be judged more fit than h_2, where $n < m$.

Proof: We are given that h_1 is responsible for n violations and that h_2 is responsible for m violations on datum s_i, where $n < m$. Thus, there is a positive integer k such that $m = n + k$. Otherwise, the two hypotheses behave identically only on s_i. That is, h_1 and h_2 both contain g superset settings, $g \geq 0$, and they both cover the tree in l nodes, $l > 0$. In this case, the fitness metric returns, for h_1, $(1 - n + bg + cl)/(2n + k + 2bg + 2cl)$. For h_2, the fitness metric returns $(1 - n + k + bg + cl)/(2n + k + 2bg + 2cl)$. Because $(1 - n + bg + cl)/(2n + k + 2bg + 2cl)$ is greater than $(1 - n + k + bg + cl)/(2n + k + 2bg + 2cl)$, h_1 is

judged more fit than h_2. As the values for n, k, g, and l were selected arbitrarily, the result holds generally.

It follows from the Violation Theorem that the fitness metric will, in general, assign higher fitness to hypotheses that lead to fewer grammatical violations on the input stream. Because selection is based on fitness, the learner will tend toward those parameter settings that minimize the number of grammatical violations.

Let us turn now to the contribution of the superset penalty, the term $\Sigma_{j=1}^{n} s_j$ for the entire population and the term s_i for a single parsing device. Suppose that P_1 and P_2 both signal no violations of any grammatical principles and both cover the input in 20 nodes. Suppose further that P_2 contains a superset setting for one parameter and that P_1 contains no superset settings. The fitness metric will then return:

(42) a. P_1 receives a fitness rating of 0.50495.
b. P_2 receives a fitness rating of 0.49505.

Notice that the "smallest hypothesis," in this case the one underlying P_I, is judged more fit than the one that violates the Subset Condition. Thus, the fitness metric can distinguish both between hypotheses that are unequal in their parsing powers and between hypotheses that are equal in parsing power but differ with respect to the Subset Condition.

A general theorem regarding subset/superset settings follows from the fitness metric:

(43) *The Superset Theorem*
Given an input datum s_i, and two hypotheses, h_1 and h_2, that are identical with respect to their behavior on s_i except that h_1 contains n superset settings whereas h_2 contains m superset settings, then h_1 will be judged more fit than h_2, where $n < m$.

Proof: In this case, we are given that h_1 and h_2 behave identically on datum s_i except that h_1 contains n superset settings, whereas h_2 contains m and $n < m$. Then there is a positive integer k such that $m = n + k$. Because their behaviors are in other respects identical on s_i, both h_1 and h_2 return g violations and cover the input in l nodes. Thus, for h_1, the fitness function returns $(1 - g + bn + cl)/[2g + b(2n + k) + 2cl]$ and, for h_2, it returns $[1 - g + b(n + k) + cl]/[2g + b(2n + k) + 2cl]$. Because $(1 - g + bn + cl)/[2g + b(2n + k) + 2cl]$ is greater than $[1 - g + b(n + k) + cl]/[2g + b(2n + k) + 2cl]$, h_1 is judged more fit

than h_2 by the fitness metric. Since the values for n, k, g, and l were selected arbitrarily, the result holds generally.

The Superset Theorem implies that the learner will tend toward the hypothesis that generates the smallest language possible (while still avoiding grammatical violations as the Violation Theorem requires). Note that (43) has the interesting result that the learner can retract overgeneral hypotheses on the basis of positive-only input. That is, this model gives a precise formulation of how indirect negative evidence could contribute to the acquisition of syntactic knowledge.

We turn, finally, to the contribution of the elegance factor; this is the term $\Sigma_{j=1}^{n} e_j$ for the entire population and e_i for individual parsing devices. Consider two hypotheses, P_1 and P_2, both of which return no violations, contain no superset settings, but cover the input with trees of different elegance. Suppose that P_1 is able to cover the input with 17 nodes, whereas, P_2 covers the input with 18 nodes. The results of the fitness metric are then:

(44) a. P_1 receives a fitness rating of 0.514286.
b. P_2 receives a fitness rating of 0.485714.

The first hypothesis is preferred by the fitness metric because it is able to span the input in a more elegant way than the second hypothesis.

In order to see the importance of this factor, consider the case where the target is SVO. Suppose that hypothesis h_1 treats the subject as being in the Spec of *IP* at S-Structure, whereas hypothesis h_2 treats the subject as having moved to the Spec of *CP*, attracting the main verb with it. For a simple clause, h_1 and h_2 will return the following structures:

(45) a. h_1: $[_{IP}\ DP\ [_{\bar{I}}\mathrm{I}\ VP]]$
b. h_2: $[_{CP}\ DP_i\ [_{\bar{C}}\mathrm{V}_j\ [_{IP}\ t_i\ [_{\bar{I}}t_j\ VP]]]]$

By assumption, both h_1 and h_2 can account for the input stream. Notice, however, that h_2 involves systematically longer chains than h_1 as the former always involves movement of the subject to the Spec of *CP*, with subsequent attraction of the verb to C^0, whereas the latter does not. The representations returned by h_1 are simpler than those returned by h_2. Because the learner, via the fitness metric, can take into account the general elegance of representations, he or she can successfully distinguish between h_1 and h_2. Notice, however, that elegance is defined quite simply as a count of the nodes in the tree covering an input item plus the lengths of the chains in the representation. This case corresponds to the difficult case of shifting discussed earlier involving the interaction between V2 and left-dislocation;

thus, the learner developed here can successfully avoid this type of trap without recourse to direct deductions based on the interactions between parameters.[20]

More generally, the following theorem is implied by the fitness metric:[21]

(46) *The Elegance Theorem*

Given an input datum, s_i, and two hypotheses, h_1 and h_2, that are identical except that the parsing device that h_1 underlies covers s_i with n nodes (plus chains), whereas the parsing device that h_2 underlies covers s_i with m nodes (plus chains), then h_1 will be judged more fit than h_2, where $n < m$.

Proof: We are given that h_1 is identical to h_2 with respect to their behavior on input datum s_i except that h_1 is responsible for a representation of s_i with an elegance of n, whereas h_2 is responsible for a representation of s_i with an elegance of m and $n < m$. Thus, there is a positive integer k such that $m = n + k$. As h_1 and h_2 are otherwise identical, then both return g violations each and both contain l superset settings each. For h_1, the fitness metric will therefore return $(1 - g + bl + cn)/[2g + 2bl + c(2n + k)]$. For h_2, the fitness metric returns $[1 - g + bl + c(2n + k)]/[2g + 2bl + c(2n + k)]$. As $(1 - g + bl + cn)/[2g + 2bl + c(2n + k)]$ is greater than $[1 - g + bl + c(n + k)]/[2g + 2bl + c(2n + k)]$, h_1 is judged more fit than h_2. As the values for n, k, g, and l were selected arbitrarily, the result holds generally.

The fitness metric can be seen as working as follows. The population of parsing devices specified by the learner's hypothesis strings is run against each input item. The term $\Sigma_{j=1}^{n} v_j + b\,\Sigma_{j=1}^{n} s_j + c\,\Sigma_{j=1}^{n} e_j$ yields the total number of violations, superset settings, and the total elegance of representations of the entire population, with the various factors weighted appro-

[20]It is interesting to note that this framework predicts that children should be relatively slow to acquire V2 phenomena. In fact, it has been reported that children acquiring German often produce root SOV orders without movement of the tensed verb to second position. This is despite the fact that all sentences they hear in the input show V2. Of course, a more precise account of this phenomena within the GA framework must be developed, although section 5 develops the essential details of the analysis. See Mills (1985) for discussion of the empirical data regarding acquisition of German.

[21]An anonymous reviewer notes that the fitness metric will tend to disfavor derivations that involve vacuous movement (see Chomsky, 1986, and Travis, 1984). The issue is delicate because in principle at least, the learner could potentially be driven to hypothesize a sequence of parameter settings that entail vacuous movement on the basis of advantageous representations for other structures. That is, vacuous movement would be acquired as a side effect of other pressures.

priately by the constants b and c. Dividing this term by n, the size of the population, would give the average number of undesirable properties for the entire population. Consider next the term $v_i + bs_i + ce_i$. This yields the number of unhealthy properties each individual parsing device carries. As this term grows in relation to the population average, the relative fitness of the parsing device decreases. If this term decreases with respect to the population average, then the parsing device is judged relatively more fit.

3.3 The Learning Algorithm

Having defined the fitness metric and having seen some of its properties, we are now in a position to specify the learning algorithm. Notice that the algorithm, presented here in pseudo-code, does not perform deductions as described in section 2. The learner is blissfully unaware of the consequences of his or her actions at any given step. However, the learner is sensitive to the relative fitness of hypotheses and uses this measure to guide it learning; thus, the algorithm is not a random walk through the space of possible grammars.

(47) *The Learning Algorithm*

1. Generate at random a population of distinct hypothesis strings.
2. Compile the hypothesis strings to their corresponding parsing devices.
3. Input a sentence from the text s_i and run the parsing devices, placing the results associated with each parsing device in a memory store.
4. Measure the fitness against the parse outputs in the memory store; associate each hypothesis string with the fitness of its corresponding parsing device.
5. Apply the genetic operators:
   (a) Select two hypothesis strings from the population and, using the crossing-over operator, mate them. For each hypothesis string, its probability of being selected for mating is equal to its fitness.
   (b) With a probability $p > 0$, perform mutation on a randomly selected string.
   (c) With a probability $p > 0$, eliminate the hypothesis (or hypotheses) with the least fitness from the population.
6. If the population consists of a single hypothesis string equal to the target sequence of parameter settings, then exit; otherwise, go to step 2.

The opportunity to reproduce (i.e., to be selected for the crossing-over operation and mutation) is a direct function of relative fitness. The algorithm in (47) assumes that the fitness associated with a hypothesis corresponds transparently to its proportion of the general population. In an environment with random mating, then, those hypotheses with a high proportion in the population are more likely to meet and reproduce. The fitness ratings are used to simulate a weighted roulette wheel, the results of which undergo the crossing-over and mutation operations. In other words, successful hypotheses will receive a high fitness rating. The fitness rating corresponds to the probability that the hypothesis will get to reproduce. Thus, the fittest hypotheses will reproduce more frequently and pass on their parameter settings to new hypotheses. Cumulatively, then, the population will tend toward the optimal set of parameter settings for the target.

The preceding discussion entails that the most fit hypotheses are the most likely to contribute to the formation of new hypotheses. These hypotheses have the greatest opportunity to pass on the parameter settings that made them fit to new hypotheses. Pruning weak hypotheses at random intervals ultimately prevents them from contributing their inferior parameter settings to the general pool. Thus, fit parameter settings tend to take over, whereas unfit parameter settings are purged. By iterating the process of parsing, judging fitness, reproduction, and death, the learner is able to incrementally approach the target grammar. The mutation operation serves to maintain population diversity until the learner converges on the target grammar. Thus, the learner can be driven toward the target by optimizing hypothesis performance via the interaction of fitness, reproduction, and death. The learner can explore the neighborhood around his or her best hypotheses via mutation. Furthermore, because of the fitness metric, the learner can evaluate hypothesis performance in a way that circumvents direct deductions from parameter settings.

3.4 Parameter Expression and Triggers

Let us now turn to a more careful consideration of the relationship among the input stream, fitness, and hypothesis formation. Recent work on language learnability has often made reference to the notion of a triggering datum relative to a particular parameter (see Lightfoot, 1989, and the references cited there). Intuitively, a triggering datum for a parameter is an input sentence that will cause the learner to set that parameter to a particular value. As we have seen, the GA framework simulates causal relations between the input data and hypothesis formation indirectly by means of fitness and natural selection. It is therefore of some interest to consider how the notion of a trigger can be formally construed within the present framework. Let us start with a simple example.

(48) John loves Mary.

Notice that certain parameters must be set in a particular way if the sentence is to be parsed. For example, the verb *love* must assign Case and a θ-role to the right so that its object *Mary* will occur after it. Furthermore, *John* must receive nominative Case under Spec-Head agreement with T^0. If nominative Case were assigned under government, (48) would violate the Case Filter because *John* would be caseless. Thus, a class of parsers with the parameters for Case assignment and θ-role assignment set in a particular way will successfully represent the sentence in (48). Let us assume that the following parameters are the relevant ones for successfully parsing (48):

(49) a. Nominative Case is assigned under Spec-Head agreement with T^0.
b. Nominative Case is assigned under head-government.
c. Accusative Case is assigned under head-government from left to right.
d. Internal θ-roles are assigned under head-government from left to right.

Other parameters (e.g., bounding nodes and those that regulate conditions on anaphora) are irrelevant. Thus, the class of parsers that accept (48) can freely vary with respect to these properties. If we add the parameters in (49) to the system developed earlier, we get the following set:

(50) a. Nominative Case is assigned under Spec-Head agreement with T^0.
b. Nominative Case is assigned under head-government.
c. Accusative Case is assigned under head-government from left to right.
d. Internal θ-roles are assigned under head-government from left to right.
e. The language allows long distance anaphors.
f. Long distance anaphors must be bound by a subject.
g. The domain in which a long distance anaphor must be bound is defined by indicative tense.
h. A verb may subcategorize for an infinitival *IP*.
i. The language allows subjects to receive abstract Case by virtue of their structural position.
j. *IP* is a bounding node for subjacency.
k. *CP* is a bounding node for subjacency.
l. *NP* is a bounding node for subjacency.

From (50), we can construct hypotheses strings by starting in order from (50a) and continuing to (50l). This gives a space of 2^{12} or 4,096 logically possible languages.

Notice that many of the grammars in the hypothesis space will be capable of assigning a well-formed representation to the sentence in (48). Using the symbol "□" to represent a "don't care" symbol, any parser with a string of the form:

(51) 1 □ 1 1 □ □ □ □ □ □ □ □ □

will be capable of assigning a well-formed representation to (48); that is, 2^9 (= 512) grammars in the hypothesis space will correctly represent (48). Notice that any hypothesis of the form:

(52) 1 □ 0 1 □ □ □ □ □ □ □ □ □

will fail to assign a well-formed representation to (48) because it has a *0* in the third position. It follows from the Violation Theorem that, given the input in (48), the fitness metric will favor hypotheses that are compatible with the "schema" (Holland, 1975) in (51) and disfavor those hypotheses, like those of the form in (52), which differ from it. Differential reproduction of the fit will ensure that those adaptive parameter settings (51) will come to dominate in the population, to the disadvantage of form like (52) that contain nonadaptive parameter settings. If data like (48) occur frequently in the input text, then the learner will be under significant pressure to set the first, third, and fourth parameters to *1* throughout the population.

The preceding suggests we can formalize the intuitive notion of trigger in the following way. First, we say that any given input sentence *expresses* certain parameters.

(53) *Parameter Expression*
A sentence ρ expresses a parameter p_i just in case a grammar must have p_i set to some definite value in order to assign a well-formed representation to ρ.

When a given datum expresses some parameter value, the learner will be under pressure to set that parameter to the value expressed by the datum, as we saw earlier. The causal relationship between parameter setting and the input stream is mediated by the fitness metric and reproduction, because the fitness metric will prefer hypotheses with the correct setting to those that lack that setting. Thus, we have captured the formal properties that underlie the intuitive notion of *triggering datum.*

(54) *Trigger*
A sentence ρ is a trigger for a parameter p_j if ρ expresses p_j.

Thus, a trigger for a parameter is any input datum that expresses that parameter.

We can now imagine a method of abstractly encoding the stream of input data in string form. In particular, suppose that the function ψ maps a sentence onto the set of sequences of parameter settings that are compatible with that sentence. For example, suppose that there are six binary parameters in UG and that a given input sentence, s_m, can be accounted for by grammars with the second and third parameters set to *0* and the fifth parameter set to *1*. Applying ψ to s_m would give the following set of parameter strings:

(55) $\psi(s_m)$ = {00001, 10001, 00011, 10011}

Again, using "□" as a variable to range over *0* and *1*, the set of strings in (55) could be replaced with a cover term, as in (56).

(56) {00001, 10001, 00011, 10011} = [□ 0 0 □ 1]

I refer to the sequence [□ 0 0 □ 1] as the *p-encoding* for s_m; the p-encoding of a sentence may be thought of as a "pure" representation of the parameters expressed by the sentence. We can now define a relation, ψ', that maps an input datum onto its corresponding p-encoding. For example, given the system of parameters defined earlier and the input datum "John loves Mary," we would have the following:

(57) ψ' (John loves Mary) = [1 □ 1 1 □ □ □ □ □ □ □ □]

Notice that, in principle, one could replace the sentences in an input text with their p-encodings and so study the frequency of expression for various parameters and the overall structure of the text relative to parameter expression. This is the technique used in Clark (1990b) to simulate parsing so that the learnability properties of large systems of parameters could be studied efficiently. Briefly, the input text was encoded as a sequence of p-encodings, and parses were simulated based on the sequence of p-encodings. As discussed earlier, the learner received only the output of the fitness metric. Although the text was severely limited in complexity (see Clark, 1990b, for details), the learner was able to converge in large hypothesis spaces containing an array of superset/subset parameters.

It must be noted that there is no requirement that parameters be

expressed unambiguously by any particular input datum. As discussed in Clark and Roberts (in press), conflicting parameter settings could account for any given datum. The crucial factor is parameter expression over time; although it may be the case that each datum ambiguously expresses a number of parameters, it must be the case that the target sequence of settings is expressed with sufficient frequency that the fitness metric will be able to sift through the ambiguities in the text and allow the learner to lock on to the target. The relationship between frequency and parameter setting is discussed more fully in the next section.

Summarizing the results of this section, we have developed a formal system for the representation of parameters and a genetic algorithm that allows the learner to find the target in the potentially immense space of possibilities without appeal to deductive mechanisms. The key ingredient is a form of natural selection. The learner is able to distinguish fit hypotheses from unfit hypotheses in a relatively superficial manner. Differential reproduction and elimination of unfit settings permit an efficient search of the hypothesis space. As we have seen, the fitness metric (which distinguishes hypotheses on the basis of grammatical violations, subset/superset relations, and the compactness of representations) combined with the genetic operators allow a precise formal model of the causal relationship that exists between the input text and parameter setting.

Finally, we have established that there is an important relationship between parameter-expression and the fitness metric. Ultimately, the fitness associated with a hypothesis governs its probability of being selected for reproduction. The more fit a hypothesis is, the more likely it is to pass on those parameter settings that made it fit.

Consider, now, parameter expression. When a parameter is expressed, those hypotheses that have the correct value for that parameter will be judged more fit than those that lack the proper value. If a parameter is expressed frequently, then those hypotheses bearing the correct value will have more opportunity to be selected for reproduction, and the appropriate parameter setting will tend to dominate in the population. Furthermore, those hypotheses bearing the incorrect will have a lower fitness rating and will tend to reproduce less frequently to the point where the parameter values that made them unfit are washed from the population. Thus, parameter settings that are expressed frequently will tend to be set quickly and efficiently by the learner. However, parameters that are expressed infrequently will tend not to affect a hypotheses fitness in the same way. The learner will have correspondingly less stake in setting the parameter correctly, and it will take the learner longer to set the parameter. In the next section, we formalize these relationships more fully and consider the relationship between convergence on the target sequence of parameter settings and overall system performance.

4. CONVERGENCE AND PERFORMANCE

There are two related problems that a learning system for natural languages must address. First, and quite generally, a learning system should show improved performance over time. Although it may be too strong to require that a system's performance at time $t + 1$ be better than its performance at time t, it is reasonable to require that in the average, its performance improve. Second, a successful learning system for natural languages must converge to the correct sequence of parameter settings. That is, if p_a is the target sequence and $h(t_i)$ is the learner's hypothesis at time t_i, then there must be a time, t_j, such that $h(t_j) = p_a$ and for all $t_k > t_j$, $h(t_k) = p_a$. The issues of convergence and performance are clearly related in the sense that improving the system's performance should bring the system closer to the target parameter setting. Recall, though, that the number of parameters is finite. This, in turn, implies that the room for improvement in system performance is itself finite. At the limit, then, if the system's performance does not degrade, the sequence of parameter settings hypothesized by the learner will be indistinguishable from p_a and will tend to remain at that hypothesis.

4.1 Defining Convergence

Let us first turn to the evaluation of system performance.[22] If $\mathcal{L}(t_i)$ is a characterization of the learner's state at time t_i, then we want a measure, μ, such that $\mu(\mathcal{L}(t_i))$ is an evaluation of the performance of $\mathcal{L}$ at time t_i. For present purposes, I assume that $\mu : L \rightarrow [0, 1]$; in other words, μ is a mapping from learning systems to the reals between 0 and 1, inclusive, where 1 is perfect performance. We then want to establish a metric of average system performance over time, where T is the sum of the time units, t_i, from the initiation of learning to the present.

(58) *A Measure of System Performance Over Time*

$$\bar{\mu}(\mathcal{L}, T) = \frac{1}{T} \sum_{t=1}^{T} \mu(\mathcal{L}(t))$$

A system improves if $\bar{\mu}$ tends toward 1 as T grows.

(59) *Improvement Over Time*

$$\lim_{T \rightarrow \infty} \bar{\mu}(\mathcal{L}, T) = 1$$

[22]The discussion of performance and convergence here owes a great deal to Holland's (1975) discussion of schemata.

That is, as T approaches infinity, the measure of system improvement should approach 1. Naturally, $\bar{\mu}$ will never reach 1; however, if the system has attained the target, μ should return 1 (perfect performance on one time step) and this should draw $\bar{\mu}$ to 1. As $\mu(\mathcal{L}(t_i)) = 1$ implies that the performance of $\mathcal{L}$ at t_i is perfect, the preceding can also be used to establish that $\mathcal{L}$, a learning system in the set of possible learning systems, L, converges.

(60) *Convergence*
A learning system, $\mathcal{L} \in L$, converges to a target p_a just in case:

$$\lim_{T \to \infty} \bar{\mu}(\mathcal{L}, T) = 1$$

where $\bar{\mu}(\mathcal{L}, T)$ is a measure of average system performance over time.

Notice that the definition of convergence in (60) does not strictly require that the learner lock on to the target and conjecture only that hypothesis. The learner can alter his or her guess occasionally but will remain on the target an overwhelmingly large number of times, given that the system detects errors in performance and seeks to optimize the learner's behavior. Hence, the definition in (60) is adequate for the present purposes.

The question, then, is how best to calculate $\mu(\mathcal{L}(t_i))$. Several techniques suggest themselves. For example, one might use the fitness metric to estimate μ, a possible strategy being to take the average fitness of the hypotheses in $\mathcal{L}$ at t_i. It is important to recall that the relative fitness of a hypothesis at any given step in the learning cycle, t_j, is calculated relative to the performance of the other hypotheses present in the population at t_j. In particular, the following holds:

$$(61)\ \sum_{i=1}^{n} \frac{(\sum_{j=1}^{n} v_j + b \sum_{j=1}^{n} s_j + c \sum_{j=1}^{n} e_j) - (v_i + bs_i + ce_i)}{(n-1)(\sum_{j=1}^{n} v_j + b \sum_{j=1}^{n} s_j + c \sum_{j=1}^{n} e_j)} = 1$$

Given (61), as long as the population size n is greater than 1, then the average fitness of the population in $\mathcal{L}$ will be less than or equal to 0.5 at most. After the learner has converged, and a single hypothesis remains, its fitness will be 1 by definition. This will cause $\bar{\mu}(T)$ to approach 1 in the limit.

The fitness metric is an inherently relational notion; it measures the performance of a hypothesis relative to other hypotheses at any given step in the learning procedure. Crucially, a hypothesis need not be correct in order to receive a high fitness rating. It is sufficient for it to perform better than the population average. Consider the case where the learner has failed to converge, but the population size has been reduced to a single hypothesis

and the learner retains this hypothesis. Since μ is calculated on the basis of fitness, $\bar{\mu}(\mathcal{L}, T)$ will approach 1 in the limit although the learner has failed to arrive at the target sequence of parameter settings, p_a. In order to push the notion to its logical extreme, we can take the case where the learner's one remaining hypothesis does not lie near the target in the hypothesis space. In this case, the performance of the learner's hypothesis, with respect to its ability to parse the input stream, may be poor although $\bar{\mu}(\mathcal{L}, T)$ will approaches 1 in the limit. This result is unsurprising given that the fitness metric measures relative, and not absolute, performance. Although such an extreme case will not happen in the type of system considered here, it still suggests that the fitness metric does not provide the appropriate means for evaluating absolute system performance.

Another technique for measuring system performance would be to consider the degree of similarity between the learner's most fit hypotheses at any time and the target sequence of parameter settings. In this case, the performance measure would compare, parameter by parameter, the best hypothesis or hypotheses against the target sequence of parameter settings p_a. The result would simply be the percentage of parameters set correctly in the hypothesis string. A hypothesis would receive 1 if all the parameters are set correctly and 0 if none of them are. Call this measure μ_0. This is a very strong measure of system performance as the learner must approach absolute identity with the target. We now define $\bar{\mu}_0(\mathcal{L}, T)$ as:

(62) *Performance Under Strong Identity*

$$\bar{\mu}_0(\mathcal{L}, T) = \frac{1}{T} \sum_{t=1}^{T} \mu_0(\mathcal{L}(t))$$

As $\bar{\mu}_0(\mathcal{L}, T)$ will approach 1 in the limit only if the learner's hypothesis is identical to the target, this suggests the following:

(63) *Strong Convergence*
A learning system, $\mathcal{L} \in L$, converges to a target p_a just in case:

$$\lim_{T \to \infty} \bar{\mu}_0(\mathcal{L}, T) = 1$$

where $\bar{\mu}_0(\mathcal{L}, T)$ is a measure of average system performance over time.

A weaker approach to convergence would be to consider only the behavior of the parsing device determined by the learner's best hypothesis. In particular, we might let μ_1 measure the ability of the best parse to

represent the input stream without violations.[23] It is a conceptual possibility that different combinations of parameter settings could conspire to derive the same language with distinct structural representations assigned by the associated parsing devices. $\bar{\mu}_1(\mathcal{L}, T)$ would then be defined as:

(64) *Performance Under Acceptable Parses*

$$\bar{\mu}_1(\mathcal{L}, T) = \frac{1}{T} \sum_{t=1}^{T} \mu_1(\mathcal{L}(t))$$

Because $\mathcal{L}$ need only arrive at a hypothesis compatible with the language generated by the target parameter setting, we define the following:

(65) *Weak Convergence*
A learning system, $\mathcal{L} \in L$, converges to a target p_a just in case:

$$\lim_{T \to \infty} \bar{\mu}_1(\mathcal{L}, T) = 1$$

where $\bar{\mu}_1(\mathcal{L}, T)$ is a measure of average system performance over time.

Weak convergence allows the learner to settle upon a hypothesis that works for the input stream, although that hypothesis may differ from the target in certain systematic ways, just so long as the resulting parsing device accepts the input.

At this stage of the inquiry, it is not yet possible to determine whether weak or strong convergence is appropriate for the study of parameter setting. In particular, it is unknown whether distinct combinations of parameter settings determine the same output language. A reasonable starting assumption is that strong convergence is the appropriate criterion; a system that strongly converges to the target also obeys weak convergence.

4.2 Three Learnability Properties

Having defined convergence in terms of the performance of the learning system, let us now consider a fundamental observation regarding the frequency of parameter expression and convergence. Let us take $(x_1, \ldots, x_{n-1}, 1, x_{n+1}, \ldots x_m)$ as the target sequence of parameter settings. That is, the target consists of a sequence of m parameter values

[23]Most plausibly, $\bar{\mu}_1$ would consider behavior over several parses. In this case, $\bar{\mu}_1$ might assign 1 to $\mathcal{L}$ if no parses fail and percentage of correct parsers over the sequence of input items parsed otherwise.

with *1* in the *n*th position, where $n \leq m$. In order for the learner to arrive at the value *1* for the *n*th parameter, it must be the case that this parameter is expressed by the input text. If it were not expressed, the fitness metric would be unable to distinguish between those hypotheses with a *1* in the *n*th position and those with a *0* in the *n*th position. Without parameter expression, the learner has no stake in setting the *n*th parameter either way. A performance metric like μ_1 (weak convergence), for example, will not distinguish between a learning system $\mathcal{L}$ that has the hypothesis $(x_1, \ldots, x_{n-1}, 1, x_{n+1}, \ldots x_m)$ and a learning system $\mathcal{L}'$ that has the hypothesis $(x_1, \ldots, x_{n-1}, 0, x_{n+1}, \ldots x_m)$. Both systems will behave identically on the input text. Furthermore, the fitness metric will be unable to distinguish between $(x_1, \ldots, x_{n-1}, 1, x_{n+1}, \ldots, x_m)$ and $(x_1, \ldots, x_{n-1}, 0, x_{n+1}, \ldots, x_m)$ because, all else being equal their behavior is the same. The former hypothesis has no reproductive advantage over the latter, and so the value *1* in the *n*th position will not come to dominate within the population. Those hypotheses with *0* in that position will be as likely to pass on this setting as will the correct hypothesis. By the criterion of strong convergence, the learner will be unable to converge in this context except by chance.[24] In other words, a learnable system of parameters $\mathcal{P}$ must have the following property:

(66) *Parameter Expressability*
For all parameters x_i in a parameterized system $\mathcal{P}$ and for each possible value, v_j, of x_i, there must exist a datum, d_k, in the input text such that a syntactic analysis τ of d_k expresses v_j.

Parameter Expressability requires that the learner be able to detect the effects of a parameter setting in the input text. However, we do not require that any given datum express a parameter setting unambiguously. As has been observed, each datum may be amenable to a variety of analyses relative to a system of parameters and, thus, express parameters ambiguously. It must be the case, however, that the full range of parameter settings in the target sequence be expressed in the input. Furthermore, the target parameter settings must be expressed with sufficient frequency that the learner can focus on them.

The preceding consideration suggests a stronger constraint. In order for the learner to converge to the value *1* in the *n*th position of the parameter string, this setting must receive expression above some minimal frequency, $f_{(1, n)}$, in the input text, s. Let $f_{(v, m)}(s_i)$ denote the probability of expression of the *m*th parameter value *v* in the input text s_i. Let $\Phi_{(v, m)}(p_a)$

[24]Parameter expression within this framework is the correlate of *detectable error* within the framework developed in Wexler and Culicover (1980).

denote the minimal probability of expression necessary for the learner to converge on the value v of the mth parameter in the target sequence of parameter settings p_a. A learnability proof for a parameterized system $\mathcal{S}$ must demonstrate the following:[25]

(67) *Frequency of Parameter Expression*
Given an input text s_i, a target parameter sequence p_a, and a learning system $\mathcal{L}$, $\lim_{T \to \infty} \bar{\mu}_0(\mathcal{L}, T) = 1$ if, for all parameter values v in positions m in the target p_a, $f_{(v, m)}(s_i) \geq \Phi_{(v, m)}(p_a)$.

In other words, the learner can strongly converge to the target given an input text, s_i just in case the expression of each target parameter setting in s_i exceeds some minimal threshold value. It must be the case that the learner can resolve conflicting possibilities that arise given the ambiguous expression of parameter settings in the input text. The target settings must be expressed with a frequency that exceeds the expression of conflicting possibilities. In this way, the learner will tend to assign an overall higher fitness to hypotheses that contain the target settings, thus resolving potential conflicts that arise between the correct values and their alternatives.

In order to guarantee that the threshold value, $\Phi_{(v, m)}(p_a)$, is exceeded for all parameters in the target—given an input text consisting of short, grammatical sentences—we must guarantee that parameters can be expressed in simple structures that the learner is likely to encounter in the input text. In particular, suppose that C is a measure of syntactic complexity that, when given a syntactic structure g, returns an integer that corresponds to the complexity of a syntactic structure, g, where $C(g_i) < C(g_k)$ implies that g_i is simpler than g_k. For the purposes of the present discussion, we may assume that the function C is simply a count of the nodes in a tree, g, plus the lengths of the syntactic chains in g. The following must be established.[26]

(68) *Boundedness of Parameter Expression.*
For all parameter values v_i in a parameterized system $\mathcal{P}$, there exits a syntactic structure τ_j that expresses v_i where the complexity $C(\tau_j)$ is less than or equal to some constant, U.

The intent of the Boundedness of Parameter Expression (BPE) is to ensure that each target value, v, plays a part in the analysis of a simple

[25]The Frequency of Parameter Expression constraint is related to Wexler and Culicover's (1980) Property 1 that "errors occur sufficiently often."

[26]The following corresponds to the Boundedness of Minimal Degree of Error (BDE) in Wexler and Culicover (1980).

structure that is likely to occur with frequency greater than $\Phi_{(v,\ m)}$, the threshold value, in the input text s_i. For the moment, we put aside the question of the precise formulation of the complexity measure C and the exact value of U. We might speculate, following Rizzi (1989), that U essentially picks out the domain of government of a head, in which case the BPE places an extremely strong constraint on the formulation of parameters.[27] The intuition here is that a value, v_i, for a parameter, x_n, which parameter could only be expressed in a structure whose complexity is greater than U, would be unlikely to be expressed with a frequency greater than the threshold, $\Phi_{(i,\ n)}$. The learner would then have little chance of setting x_n to v_i, thus preventing $\bar{\mu}_0$ from approaching 1 in the limit.

The learning system described here tests the input stream for the expression of parameters by exploiting the notion of relative fitness. When a parameter value is expressed by a given datum, those hypotheses that bear the relevant settings will receive a higher fitness and will be reproductively favored. Those that do not bear the proper settings will tend not to reproduce and will stand a higher chance of being pruned from the population of active hypotheses. Because the fitness metric judges relative fitness, and not absolute correctness, of hypotheses, the learner can exploit partially correct hypotheses, using the basic genetic operators to create new hypotheses from subparts of successful hypotheses. The genetic operators tend not to perpetuate past failures because those hypotheses with incorrect settings tend to have a poor chance of engaging in reproduction but a high chance of being removed from the population. Thus, relative fitness and the genetic operators interact in such a way as to maximize $\bar{\mu}_0$. Furthermore, the learner will tend to be opportunistic, because he or she will tend to maximize $\bar{\mu}_0$ in as few trials as possible.

4.3 The Endgame Problem

The tendency to maximize system performance can cut both ways (Berwick, 1991; Clark, 1991). On the one hand, the learner will tend to exploit a past observed best hypothesis, h_i, in order to maximize performance. On the other hand, there may be another as yet untested hypothesis, h_j, whose performance will outstrip h_i. Nevertheless, there is an increasing chance as the learner approaches the target that h_j will degrade system performance. In order to maximize μ at any given step the learner may tend to rest with a near optimal hypothesis instead of risking a drop in performance in order

[27]Chomsky's (1986) notion of *barrier* (defined in terms of θ-government and L-marking) plus Spec-Head Agreement could be taken as placing an upper limit on the syntactic domain that could be mentioned by a parameter, for example.

to gain new information.[28] In particular, there is some probability $p(h_i)$ that h_i is the best hypothesis, and there is a probability $p(h_j)$ that h_j is the best hypothesis. Notice that the two probabilities may overlap, although as the overall system performance increases, $p(h_i) - p(h_j)$ will also tend to increase. This leads the learner to prefer to remain with h_i rather than risk testing h_j. Nevertheless, because $p(h_j) > 0$, there is still a measurable chance that the learner could win by testing h_j. Thus, the system's greediness relative to maximizing performance can actually delay convergence significantly.

The differential reproduction rate of fit hypotheses causes certain parameter settings to dominate within a population. As a result, as the learner approaches an optimal hypothesis located near the target, population diversity tends to diminish. This in turn reduces the learner's ability to create new hypotheses via the genetic operators. If all the hypotheses contain *1* in the third position in the parameter string, then the only way to create a hypothesis with a *0* in the third position is by random mutation. It is possible, then, that the learner's best hypothesis differs from the target by only one or two parameter settings that are not represented in the population. Converging on the target will then rely on a random walk (via the mutation operator) around the optimal hypothesis, a technique that does not differ significantly from a brute enumerative search.

The situation is complicated by the fact that the incorrectly set parameters may be expressed only relatively infrequently in the input text. That is, the frequency, $f_{(i,\ n)}$, for the value v_i of the incorrectly set parameter x_n may be barely above the threshold, $\Phi_{(i,\ n)}$ necessary to set the parameter. In order to set the parameter correctly, the learner must have a hypothesis containing the correct setting available when that parameter is expressed by some input datum. Otherwise, no particular reproductive advantage will be given to the hypothesis containing the correct setting, and the random walk will continue. The two conditions of availability in the population and parameter expression must be satisfied contemporaneously if the learner is to converge quickly. As this situation is frequently encountered toward the end of the learning cycle, when the learner has found a near optimal hypothesis, we refer to it as the *endgame problem.*

(69) *The Endgame Problem*
Given the presence of a near optimal hypothesis that dominates

[28]This is essentially the two-armed bandit problem. Holland (1975) and Goldberg (1989) both provided extensive discussions of this problem with respect to GAs. It should be kept in mind, however, that the two-armed bandit problem holds generally and is not specific to GAs; as both Holland and Goldberg noted, the implicit parallelism of GAs tends to ameliorate the problem.

> the population, how can the learner guide his or her search for the target in a way that both maintains system performance and avoids a random walk around the optimum?

It might seem that the solution lies simply in maintaining variation within the population. This condition alone will not suffice to guarantee rapid convergence. The biases introduced by the higher reproductive rate of fitter hypotheses can slow the rate at which the correct setting is introduced into the optimal hypothesis string. In particular, if the correct setting is only represented in hypothesis strings with otherwise poor performance, whereas the more fit strings lack this setting, the probability of mating between a poor hypothesis and a more fit hypothesis is sufficiently low as to slow the learner. In order to guarantee rapid convergence, it must be the case that population diversity is maintained and that new hypotheses are constructed frequently by virtue of mating dissimilar pairs. This will force the system to search for new information. However, because the majority of these newly created hypotheses will be less fit than the learner's optimum, the average performance of the learning system will be degraded. Thus, the trade-off between system performance and gaining new information remains.

One method for dealing with the endgame problem would be to elaborate the representation of the hypotheses in such a way as to "cache" variation within the hypothesis string itself. To this point, the hypothesis strings have been haploidal (i.e., a single string). Following Holland (1975), we might choose a diploidal representation (a pair of strings) and define a dominance relation for parameter values at each position in the string. For example, suppose that UG contains four binary parameters. We now allow each position in the hypothesis string to consist of a pair of values, v_1 and v_2. For example, a possible hypothesis string would be:

(70) ((1 1) (0 1) (0 0) (1 0))

In order to interpret (70) as a sequence of parameter settings, we must define a dominance relation for each position. For the sake of simplicity, I assume that if, at any position, $v_1 = v_2$, then the parameter is interpreted as set to v_1. Otherwise, if they disagree, we define which value is dominant by giving a table of dominance relations (relative order of the two values is irrelevant), as shown in Figure 7. Thus, if the first position contains a pair $v_1 \neq v_2$, then the value is taken as *0*. Given this, we can interpret (70) as the hypothesis string:

(71) 1 1 0 0

Dominance Table	
p_1	0 1 ⟶ 0
p_2	0 1 ⟶ 1
p_3	0 1 ⟶ 1
p_4	0 1 ⟶ 0

FIGURE 7 Table of dominance relations.

Notice, now, that the diploidal representation plus dominance could allow a certain amount of variation to be hidden within representations. Despite this variation, a single near optimal phenotype could be expressed by a population containing genotypic variation. By defining the genetic operators accordingly, recessive settings could reemerge and be tested, potentially, this could push the learner off of a false optimum near the end of the learning cycle and speed convergence. It should be noted, however, that this model remains to be tested via a working simulation.[29]

5. INTERPRETING THE MODEL

The model presented in the preceding sections is a general computational framework for studying the learnability properties of parameterized systems of grammar. We should be careful to distinguish between models of learnability and models of language acquisition. The former demonstrate *in principle* learnability; in other words, an algorithm is given that, upon exposure to a text, will converge on a target contained in a well-defined set of grammars. Although the algorithm may use strictly bounded computational resources and may converge in a plausible amount of time, it does not follow that it provides a plausible theory of language acquisition. Instead, a demonstration of in principle learnability is a first step toward a full-blown theory of language acquisition. A theory of grammar for which there is no tractable learning algorithm is unlikely to play an important role in a theory of language acquisition. It does not follow, though, that a theory of grammar accompanied by a computationally tractable theory of learnability has solved the puzzle of language acquisition.

A true theory of language acquisition must address the following issues, among many others:

[29]The simulation described in Clark (1990b) uses the simple haploidal representation assumed throughout most of this article.

(72) a. A characterization of the developmental stages of language acquisition (Brown, 1973).
b. A characterization of the errors children make when acquiring their first language.
c. An explanation as to why certain types of logically possible errors do not occur.
d. A set of mechanisms that are biologically plausible in the sense that they could, in principle, characterize computations performed by the human nervous system.

These issues are, no doubt, related by basic mechanisms that are tied to the structure of the human language ability. To the degree that we can derive the properties found at the various developmental stages, we have gone a long way toward understanding the real errors that children make and, perhaps, why they fail to make some obvious mistakes. Furthermore, these properties may follow from natural constraints that are linked to underlying biological mechanisms. Notice though that a theory of in principle learnability could fail on any of the issues listed in (72) but still succeed in establishing that a class of grammars has the learnability property. A true theory of language acquisition cannot demonstrably fail on the stated problems and still count as a successful theory of acquisition. I maintain, despite the differences in success criteria, that the theory of learnability must inform the theory of acquisition and that the theory of acquisition must be founded on a theory of learnability. The computational question of in principle learnability is prior to a detailed formal theory of language acquisition in the sense that only a grammatical theory with the learnability property is a plausible candidate for the theory of acquisition. Therefore, it is fair to ask how far the theory of learnability can be pushed with respect to the issues in (72).

5.1 An Example of Developmental Sequencing

A detailed interpretation of the learnability model would presuppose a specific system of parameters and a careful analysis both of child performance and adult input to children. Such an analysis would be beyond the scope of this article. The methodology, however, is clear, and some general comments, speculative though they may be, can be made. First, the GA model, like many others, assumes that the continuity hypothesis (Gleitman & Wanner, 1982; Pinker, 1979, 1984; White, 1982) is correct. The continuity hypothesis assumes that the learner always hypothesizes possible grammars. I have assumed, throughout, that the hypothesis space for the learner is exhausted by the set of possible parameter settings, each of which constitutes a possible target. Note that this type of model could violate

continuity if the learner were permitted to hypothesize combinations of parameter settings that could not count as a possible target. For example, as noted, it is possible that natural languages that have syntactic movement must have at least one bounding node for subjacency; potentially, the learner could go through a stage where no bounding nodes exist. If so, such a stage would not be a possible target.

Second, like other models, the present framework has factored out lexical acquisition as, to some degree, independent and prior to syntactic parameter setting.[30] Notice, now, that when the learner encounters an exotic lexical item, the fitness metric will not be able to distinguish between competing hypotheses on the basis of violations caused due to the presence of the unknown element. This is because the performance of all hypotheses will be uniformly degraded. Because fitness measures differences in performance, the learner will be unable to distinguish between any hypotheses if performance is uniform. For a possible application of this property of fitness, see Hyams (1986), Pierce (1989), and Stromswold (1990) on the acquisition of modals in English. Here, the learner is unable to place modals in a coherent grammatical class. The present framework predicts, then, that no parameter setting should take place until the learner is able to form a coherent class of modals that could then feed the syntactic analyses. Uniform degradation of the hypotheses due to the presence of the as yet unanalyzed modals and auxiliaries might, then, account for the delay in the acquisition of certain word order phenomena without appeal to other acquisition machinery. This point is expanded upon later, but see Hyams (1986), Pierce (1989), Stromswold (1990), and the references cited in these works for discussion of the relevant phenomena.

The GA model, though, involves a number of rather subtle interactions that may provide some additional insight to the issues in (72). As we saw in the previous section, the current model relies on the frequency of parameter expression in the input text to drive the learner to converge on the correct sequence of parameter setting. It does not, however, predict that the learner will simply duplicate the frequencies of constructions encountered in the input. There is a natural tension that exists between the various terms in the fitness metric. As was demonstrated earlier, the learner is driven to that sequence of parameter settings that minimize grammatical violations (the Violation Theorem), the size of the language generated (the Superset Theorem), and the size of the representations generated (the Elegance Theorem). These three conditions need not work together in perfect harmony to drive the learner smoothly toward the target they may, in fact, pull the learner in different directions. For example, the metric may, for a

[30]For an analogous assumption, within a rather different learnability framework, see Wexler and Culicover (1980).

time, prefer a grammar that minimizes the size of representations at the cost of generating a greater number of grammatical violations.

Let us consider, briefly, one case where the various terms in the fitness metric are at odds with each other. As has been noted by many researchers (Clahsen, 1990; Miller, 1979; Mills, 1985; Roeper, 1973), children learning German tend, at an early age, to produce verb-final utterances (examples (73a) and (73b) cited in Mills, 1985, from the Miller corpus; examples (73c) and (73d) from Clahsen, 1990).

(73) a. teddy sofa fahren
teddy moped drive
'Teddy drives the moped.'
b. teddy holen
teddy fetch
'Fetch Teddy.'
c. nur pier Julia neid
only paper Julia cut
'Julia is only allowed to cut paper.'
d. ich schaufel haben
I shovel have
'I have {the/a} shovel.'

This is despite the fact that German has obligatory V2 in root clauses. Thus, although German is underlyingly SOV, many utterances will have the order SVO(V) where the tensed verb is moved to second position and untensed and participial forms are left in final position. Why are German children not misled by the root V2 structure of clauses into hypothesizing that their language is SVO?

Let us assume that V2 order in root clauses is the result of movement of the inflected verb to C^0 and obligatory movement of some constituent, *XP*, to the specifier position of *CP*.[31] This implies the presence, in root clauses, of an obligatory chain that connects the verb in C^0 to its base position as well as a chain linking an element in the specifier of *CP* with its base position, as shown schematically in (74).

(74) $[_{CP}\ XP_i[\overline{C}[_{C^0}\ V_j]\ [_{TP}\ \ldots\ t_i\ \ldots\ t_j]]]$

Notice that the movements shown in (74) will add to the complexity measure adopted here with respect to the fitness metric due to the presence

[31]This is in keeping with much recent work in generative syntax (see Belletti, 1990; Pollock, 1989; Rizzi, 1991; Rizzi & Roberts, 1989, among many, many others). For articles more specifically related to V2 in German, see Haider (1986), the articles collected in Toman (1984), and Haider and Prinzhorn (1986).

of relatively long chains. That is, all things being equal, the fitness metric will penalize a hypothesis that forces movements of the type shown in (74), a result that follows immediately from the Elegance Theorem. Unless this penalty is outweighed by some other factor (e.g., grammatical violations), the fitness metric will tend to disprefer a hypothesis that forces movement and will select a hypothesis that allows elements to remain *in situ*.

As observed in Mills (1985) the use of modals and auxiliaries is common in adult speech to children. In these cases, the thematic verb will be in its canonical position. Furthermore, one form of the imperative involves verb-final order as well as the (dialectal) use of the auxiliary *tun* ('do'). These are illustrated in (75a), (75b), and (75c), respectively (from Mills, 1985).

(75) a. willst du ein Haus bauen?
want (Modal) you a house build
'Would you like to build a house?'

b. jetzt aufstehen!
now up-stand
'Now stand up!'

c. wir tun hier Bilder malen
we AUX here pictures painted
'We painted pictures here.'

The frequency of such utterances is apparently fairly high. Indeed, these utterances, which express the verb final parameters, must occur with a frequency above the threshold for parameter setting. By the Frequency of Parameter Expression property, the frequency of verb-final constructions is greater than Φ, the base frequency for parameter setting. Thus, the child has evidence that *VP* is head-final. The fitness metric will thus prefer hypotheses where the word-order parameters have been fixed for verb final *VP* as a consequence of the Violation Theorem.

Having established the basic word order as OV, however, the learner faces an apparent dilemma. Due to the structure of V2 phenomena, the hypothesis that is able to analyze verb second structures is also one that is penalized for having more complex representations (the Elegance Theorem). If the learner were presented solely with the thematic verb in second position, the fitness metric would drive him or her quickly toward setting the V2 parameters, due to the number of grammatical violations that would otherwise occur. But the examples in (75) show that such is not the case. The learner is often presented with the thematic verb in final position due to the presence of auxiliaries and modals. Thus, the frequency of thematic verbs in V2 position must be less than Φ, the base frequency for setting the V2 parameter.

Mills (1985) pointed out that auxiliaries and modals are still rare in child speech at this stage, presumably because the child has not yet fixed a coherent grammatical class for them. Thus, as pointed out earlier, all hypotheses will be penalized on this type of input until such time as the learner acquires modals and auxiliaries as a grammatical class. Such an across-the-board degradation of performance will be discounted by the fitness metric, because it cannot be used to decide between hypotheses. Thus, the factor that would give a competitive edge to a hypothesis with the V2 parameters set correctly will be disregarded by the fitness metric. The V2 settings will provide no extra fitness to hypotheses and, furthermore, will detract from their fitness on sentences involving modals auxiliaries, or the type of imperative in (75b). The very data that provide support for the verb-final settings subvert the correct settings for V2 because (a) they generate across-the-board violations that are then disregarded by fitness, and (b) the V2 settings increase the complexity of representations.

In summary, the interaction among the Violation Theorem, the Elegance Theorem, and the Frequency of Parameter Expression implies that the fitness metric will prefer hypotheses that have set the word-order parameters to verb final and that do not have movement of the tensed verb to C^0. On the whole, then, we predict that children should produce root verb-final utterances. Notice that we also predict that V2 positioning of the thematic verb should be frequent enough in the input stream to keep the V2 settings in the population of hypotheses. Thus, children should produce occasional V2 orders. Finally, when the child has acquired the modals and auxiliaries as a true class, the Violation Theorem predicts that the fitness metric will correctly distinguish V2 from non-V2 hypotheses. The latter will consistently involve a grammatical violation that the former will not. In other words, the across-the-board violations triggered by the presence of modals and auxiliaries will disappear, removing a confounding factor for the learner. Thus, after acquisition of the modals and auxiliaries, the fitness metric will guarantee that the V2 settings will dominate in the population, and the child should correctly begin producing consistent root V2 utterances.[32]

This analysis is, of course, only suggestive. It cannot be taken as confirmed in advance of a detailed model of the parameters involved and of a careful analysis of caretaker speech to children. Many other features of child language may be correlated with the acquisition of V2 (Clahsen,

[32]See Hyams (1986) for evidence that this is indeed correct. Similar analyses should be applicable to the acquisition of auxiliary verbs in English (Hyams, 1986; Pierce, 1989) and to the late appearance of subject auxiliary inversion (de Villiers & de Villiers, 1985; Stromswold, 1990). Indeed, one might extend the analysis to acquisition of VSO ordering, which also tends to be delayed (Ochs, 1985).

1990). I have included it here for two reasons. First, it illustrates how one might go about mapping a formal learnability theory onto a theory of language acquisition. Second, it demonstrates that, although the GA theory consists of extremely simple learning mechanisms, it can nonetheless be used to study developmental sequencing. The interplay of the various components of the theory (frequency of expression, violations, elegance, etc.) can themselves conspire to derive at least some acquisition sequences without appeal to special devices such as deduction or maturation (Borer & Wexler, 1987). Although it may well be that such devices must be appealed to in a complete theory of language acquisition, it is interesting to observe how far one can go without them.

5.2 A Biological Interpretation

The image of a population of parsing devices competing and mating in the rough-and-tumble world of linguistic input is amusing but may seem somewhat remote from the realities of language acquisition. I put forth a proposal in this section that, in fact, a perfectly sensible interpretation of the model exists. In the absence of a full model, however, the remarks here must be taken as speculative, and I leave the development of such a model for future research. Nevertheless, some preliminary comments can be made.

As I argued earlier, the learning model here can be adapted to actual parsing models. The hypothesis strings can be taken as encoding either a grammar or a parsing device. We could, for example, design an algorithm that interprets each string somewhat in the way in which a universal Turing machine can interpret indices as programs. Earlier, I used the analogy of a computer accompanied by a set of circuit boards that could be snapped into the appropriate slots to modify the computer's behavior. The GA could search the alternatives relative to the task at hand and find the optimal configuration of boards for the computer. Given a parsing model that clearly reflects the linguistic modules in the parsing algorithm, such as that described by Berwick (1987), Fong (1990), or Gibson (1991), for example, the technique of tying the learning algorithm with the parser would be, first, to parameterize the various modules by isolating those parts of the program where variation could occur. Parameter values would then be alternative subprograms that could be snapped into place to create a running parser. Each bit position in a hypothesis string would then be indexed to a file containing a subprogram and instructions as to where to place the subprogram in the main body of the code. To interpret a hypothesis string, the learner would move through the hypothesis string, locate each file, and copy its contents into the main parsing program. The program could then be compiled and run against the input text, the learner's behavior evaluated by the fitness metric, and the related hypothesis string subjected to the

genetic operators. In this way, a relatively compact program could represent a large number of parsing devices and learn from a positive-only input text.

Imagine, now, that the parsing device is represented by a network of circuits. For example, some circuits would propose $\overline{X}$ representations of the input, others would construct A- and $\overline{A}$-chains, and still others would work as detectors to verify that conditions like the Case Filter and the θ-Criterion were obeyed. A sentence would be input to the network, activating certain circuits that would begin transmitting messages throughout the network of circuits until some well-defined pattern of activation is established that would serve as a representation of the input sentence. Some of the detectors, $(\delta_1, \delta_2, \ldots, \delta_n)$, might have alternative designs that would, for example, test for an abstract Case assignment to the right or to the left of a head, X^0. For example, $\delta_1(0)$ might be a detector that responds to a relation of θ-role assignment to the left, whereas $\delta_1(1)$ would respond to θ-role assignment to the right. We can imagine that, at the start, these alternative circuits are connected to the machine in parallel, where each alternative is associated with a probability that it will be activated on any given parse. Alternatively, the circuits are themselves connected together with a set of weights.

The problem is to find the best way of distributing the probabilities among the various detectors and circuits so as to maximize the machine's ability to represent the input stream. The various patterns of circuits and detectors could be represented as a binary string. The machine represented by this binary string could be tested against a stream of input sentences, and the optimal distribution of probabilities could be found using the fitness metric plus the genetic operators, as described earlier. An algorithm for adjusting the probabilities would, of course, have to be specified.[33] In addition, some method of interpreting the representations generated by such a network in a way that has the same properties as phrase markers (for purposes of the elegance factor in the fitness metric) would have to be developed. Nevertheless, the circuit interpretation of the hypothesis strings does not alter the essential mechanisms for learning discussed here; the basic change lies in the interpretation of the hypothesis strings.

The view, implicit in the preceding, that the nervous system can be modeled by a set of circuits is, of course, neither new nor original. McCulloch and Pitts (1943/1988) established a calculus for describing neural mechanisms in terms of circuits. Hebb's (1949) *cell assemblies* are another variant, and certainly connectionism and its variants continue in

[33]For example, a weight could be incremented by some factor Δ, if it was activated by a parser of above average performance and decremented otherwise. See Belew et al. (1990), Caudell and Dolan (1989), Cavicchio (1970), and Holland (1975), for some work relating neural nets with genetic algorithms.

this tradition.[34] The view that natural selection takes place over groups of neurons is an important element in Edelman's (1987, 1989) theory of neuronal group selection (the TNGS). We can interpret the present theory in terms of the TNGS by supposing that parameter values are represented by neuronal groups. These competing neuronal groups would be part of some larger, fixed brain structure. In fairness, I should note that Edelman placed great emphasis on the variability present in neural circuitry that arises from stochastic biochemical processes acting on the organism during embryological development. It is an open question, then, whether the random processes that act on a developing organism are compatible with the sort of fixed structures presupposed by a P&P model.

The preceding comments are, of course, far from giving an articulated theory of parsing and learnability. This article is intended to give the formal foundations of parameter setting and learnability in a context that relies on well-understood mathematical and biological notions (particularly, population genetics). Recent developments in cognitive science have been marked by both increasing emphasis on biological modeling and a decline in interest in traditional artificial intelligence techniques, especially with the rise of connectionism. The increased emphasis on quasi-biological modeling has brought with it an attack on "classical" symbol-processing models (but see the articles collected in Pinker & Mehler, 1988, for a potent counterattack; particularly, Fodor & Pylyshyn, 1988, and Pinker & Prince, 1988). Whereas traditional information-processing models have been far removed from biology, much recent work has lost some of the advantages of the information processing paradigm (see, in particular, the critique of connectionist models of verb learning in Pinker & Prince, 1988). The work I have discussed here represents a third way between the classical information-processing paradigm and the biological paradigm. Although the model is explicitly biological in focus, it is tied to a model of symbol processing in such a way as to reconcile the physical with the abstract.

ACKNOWLEDGMENTS

Support for this research came from a grant from the Fondation Ernst et Lucie Schmidheiny and from grant #11-25362.88 from the Fonds national suisse pour la recherche scientifique.

[34]The classic work on artificial neural nets is Nilsson (1965/1990). There is an enormous literature on connectionism. Some relevant titles I consulted were Ackley, Hinton, and Sejnowski (1988), Feldman and Ballard (1988), Hinton (1990), and Waltz and Pollack (1988). The massive Rumelhart and McClelland (1986) and the equally massive McClelland and Rumelhart (1986) are probably the basic works in the area. As is clear from my comments, I do not adopt the hypothesis that neural nets can replace symbol processing. The latter forms a necessary foundation for theorizing. However, much can be learned by attempting to relate symbol-processing theories to neural circuitry.

I acknowledge help and comments from Robert Berwick, Ami Dykman, Luigi Rizzi, Ian Roberts, Tali Siloni, Eric Wehrli, and an anonymous reviewer for *Language Acquisition*. The errors are uniquely my own.

REFERENCES

Ackley, D., Hinton, G., & Sejnowski, T. (1988). A learning algorithm for Boltzmann machines. In D. Waltz & J. Feldman (Eds.), *Connectionist models and their implications: Readings from cognitive science* (pp. 285–307). Norwood, NJ: Ablex.

Anderson, S. R. (1986). The typology of anaphoric dependencies: Icelandic (and other) reflexives. In L. Hellan & K. K. Christensen (Eds.), *Topics in Scandinavian syntax* (pp. 65–88). Dordrecht, The Netherlands: Reidel.

Belew, R., McInerney, J., & Schraudolph, N. (1990). *Evolving networks: Using the genetic algorithm with connectionist learning* (CSE Technical Report No. CS90-17). San Diego: University of California.

Belletti, A. (1990). *Generalized verb movement: Aspects of verb syntax*. Torino, Italy: Rosenberg and Sellier.

Berwick, R. (1985). *The acquisition of syntactic knowledge*. Cambridge, MA: MIT Press.

Berwick, R. (1987). *Principle-based parsing* (Technical Report No. 972). Cambridge, MA: MIT Artificial Intelligence Laboratory.

Berwick, R. (1991, March). *Parsing and language acquisition: From rules to parameters*. Paper presented at the American Association for Artificial Intelligence Spring Symposium on Machine Learning, Natural Language and Ontology, Stanford, CA.

Booker, L. B., Goldberg, D. E., & Holland, J. H. (1990). Classifier systems and genetic algorithms. In J. Carbonell (Ed.), *Machine learning: Paradigms and methods*. Cambridge, MA: MIT Press.

Borer, H., & Wexler, K. (1987). The maturation of syntax. In T. Roeper & E. Williams (Eds.), *Parameter setting* (pp. 123–172). Dordrecht, The Netherlands: Reidel.

Brown, R. (1973). *A first language: The early stages*. Cambridge, MA: Harvard University Press.

Carbonell, J. (Ed.). (1990). *Machine learning: Paradigms and methods*. Cambridge, MA: MIT Press.

Caudell, T., & Dolan, C. (1989). Parametric connectivity: Training of constrained networks using genetic algorithms. In J. Schaffer (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms* (pp. 370–374). San Mateo, CA: Morgan Kaufmann.

Cavicchio, D. J. (1970). *Adaptive search using simulated evolution*. Unpublished doctoral dissertation, University of Michigan, Ann Arbor.

Chien, Y.-C., & Wexler, K. (1990). Children's knowledge of locality conditions in binding as evidence for the modularity of syntax and pragmatics. *Language Acquisition, 1*, 225–295.

Chomsky, N. (1965). *Aspects of the theory of syntax*. Cambridge, MA: MIT Press.

Chomsky, N. (1977). On *wh*-movement. In P. Culicover, T. Wasow, & A. Akmajian (Eds.), *Formal syntax* (pp. 71–132). New York: Academic.

Chomsky, N. (1981a). *Lectures on government and binding*. Dordrecht, The Netherlands: Foris.

Chomsky, N. (1981b). Principles and parameters in syntactic theory. In N. Hornstein & D. Lightfoot (Eds.), *Explanation in linguistics: The logical problem of language acquisition* (pp. 32–75). London: Longman.

Chomsky, N. (1985). *Knowledge of language*. New York: Praeger.

Chomsky, N. (1986). *Barriers.* Cambridge, MA: MIT Press.

Chomsky, N. (1989). Some notes on economy of derivations and representations. *MIT Working Papers in Linguistics, 10,* 1–31.

Chung, S., & McCloskey, J. (1987). Government, barriers, and small clauses in Modern Irish. *Linguistic Inquiry, 18,* 173–237.

Clahsen, H. (1990, May). *Constraints on parameter setting: A grammatical analysis of some acquisition stages in German child language.* Paper presented at the University of Geneva.

Clark, R. (1990a, October). *Causality, natural selection and parameter setting.* Paper presented at the 15th Annual Boston University Conference on Language Development, Boston.

Clark, R. (1990b). *Papers on learnability and natural selection* (Tech. Rep. in Formal and Computational Linguistics, No. 1). Geneva: Université de Genève, Département de Linguistique.

Clark, R. (1991, March). *A computational model of parameter setting.* Paper presented at the American Association for Artificial Intelligence Spring Symposium on Machine Learning, Natural Language and Ontology, Stanford, CA.

Clark, R., & Roberts, I. (in press). A computational model of language learning and language change. *Linguistic Inquiry.*

Darwin, C. (1859). *On the origin of species.* London: John Murray.

Dawkins, R. (1983). *The extended phenotype: The long reach of the gene.* Oxford: Oxford University Press.

Dawkins, R. (1986). *The blind watchmaker.* London: Penguin.

de Villiers, J., & de Villiers, P. (1985). The acquisition of English. In D. Slobin (Ed.), *The crosslinguistic study of language acquisition: Vol. 1. The data* (pp. 27–139). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Dresher, B. E., & Kaye, J. (1990). A computational learning model for metrical phonology. *Cognition, 34,* 137–195.

Edelman, G. (1987). *Neural Darwinism: A biological theory of neuronal group selection.* New York: Basic.

Edelman, G. (1989). *The remembered present: A biological theory of consciousness.* New York: Basic.

Falconer, D. (1989). *Introduction to quantitative genetics.* Essex: Longman Scientific & Technical.

Feldman, J., & Ballard, D. (1988). Connectionist models and their properties. In D. Waltz & J. Feldman (Eds.), *Connectionist models and their implications: Readings from cognitive science* (pp. 13–627). Norwood, NJ: Ablex.

Finer, D. (1987). Comments on Solan. In T. Roeper & E. Williams (Eds.), *Parameter setting* (pp. 211–219). Dordrecht, The Netherlands: Reidel.

Fodor, J. A., & Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: A critical analysis. In S. Pinker & J. Mehler (Eds.), *Connections and symbols* (pp. 3–71). Cambridge, MA: MIT Press.

Fong, S. (1990). *The computational implementation of principle-based parsers* (MIT Parsing Volume, 1988–1989). Cambridge, MA: Massachusetts Institute of Technology, Center for Cognitive Science.

Gibson, E. (1991). *A computational theory of human linguistic processing: Memory limitations and processing breakdown.* Unpublished doctoral dissertation, Carnegie Mellon University, Pittsburgh.

Gleitman, L., & Wanner, E. (1982). Language acquisition: The state of the art. In L. Gleitman & E. Wanner (Eds.), *Language acquisition: The state of the art* (pp. 3–48). Cambridge: Cambridge University Press.

Goldberg, D. (1989). *Genetic algorithms in search, optimization, and machine learning.* Reading, MA: Addison-Wesley.

Haider, H. (1986). V-Second in German. In H. Haider & M. Prinzhorn (Eds.), *Verb second phenomena in Germanic languages* (pp. 47–75). Dordrecht, The Netherlands: Foris.

Haider, H., & Prinzhorn, M. (Eds.). (1986). *Verb second phenomena in Germanic languages.* Dordrecht, The Netherlands: Foris.

Haldane, J. B. S. (1990). *The causes of evolution.* Princeton, NJ: Princeton University Press.

Hebb, D. O. (1949). *The organization of behavior.* New York: Wiley.

Hinton, G. (1990). Connectionist learning procedures. In J. Carbonell (Ed.), *Machine learning: Paradigms and methods* (pp. 185–234). Cambridge, MA: MIT Press.

Holland, J. (1975). *Adaptation in natural and artificial systems.* Ann Arbor: University of Michigan Press.

Hyams, N. (1986). *Language acquisition and the theory of parameters.* Dordrecht, The Netherlands: Reidel.

Kazman, R. (1991). *The induction of the lexicon and the early stages of grammar.* Unpublished doctoral dissertation, Carnegie Mellon University, Pittsburgh.

Lightfoot, D. (1989). The child's trigger experience: Degree-0 learnability. *Behavioral and Brain Sciences, 12*(2), 321–375.

McClelland, J., & Rumelhart, D. (1986). *Parallel distributed processing: Explorations in the microstructure of cognition: Vol. 2. Applications.* Cambridge, MA: MIT Press.

McCulloch, W., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. Reprinted in W. McCulloch (1988), *Embodiments of mind* (pp. 19–39). Cambridge, MA: MIT Press.

Mehler, J., & Dupoux, E. (1990). *Naître humain* [To be born human]. Paris: Editions Odile Jacob.

Michalsky, R. S., Carbonell, J., & Mitchell, T. (Eds.). (1983). *Machine learning: An artificial intelligence approach.* San Mateo, CA: Morgan Kaufmann.

Michalsky, R. S., Carbonell, J., & Mitchell, T. (Eds.). (1986). *Machine learning: An artificial intelligence approach, Volume II.* San Mateo, CA: Morgan Kaufmann.

Miller, M. (1979). *The logic of language development in early childhood.* New York: Springer-Verlag.

Mills, A. E. (1985). The acquisition of German. In D. Slobin (Ed.), *The crosslinguistic study of language acquisition: Vol. 1. The data* (pp. 141–254). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Morgan, J. (1986). *From simple input to complex grammar.* Cambridge, MA: MIT Press.

Newport, E. (1976). Motherese: The speech of mothers to young children. In N. Castellan, D. Pisoni, & G. Potts (Eds.), *Cognitive theory* (Vol. 2, pp. 367–371). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Nilsson, N. (1990). *The mathematical foundations of learning machines.* San Mateo, CA: Morgan Kaufmann. (Original work published 1965)

Nyberg, E. (1991). *A non-deterministic, success-driven model of parameter setting in language acquisition.* Unpublished doctoral dissertation, Carnegie Mellon University, Pittsburgh.

Ochs, E. (1985). Variation and error: A sociolinguistic approach to language acquisition in Samoa. In D. Slobin (Ed.), *The crosslinguistic study of language acquisition: Vol. 1. The data* (pp. 783–838). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Osherson, D., Stob, M., & Weinstein, S. (1986). *Systems that learn: An introduction to learning theory for cognitive and computer scientists.* Cambridge, MA: MIT Press.

Pica, P. (1984). Subject, tense and truth: Towards a modular approach to binding. In J. Gueron, H. Obenauer, & J.-Y. Pollock (Eds.), *Grammatical representation* (pp. 259–291). Dordrecht, The Netherlands: Foris.

Pierce, A. (1989). *On the emergence of syntax: A crosslinguistic study.* Unpublished doctoral dissertation, Massachusetts Institute of Technology, Cambridge.

Pinker, S. (1979). Formal models of language learning. *Cognition, 1,* 217–283.

Pinker, S. (1984). *Language learnability and language development.* Cambridge, MA: Harvard University Press.

Pinker, S., & Mehler, J. (Eds.). (1988). *Connections and symbols.* Cambridge, MA: MIT Press.

Pinker, S., & Prince, A. (1988). On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. In S. Pinker & J. Mehler (Eds.), *Connections and symbols* (pp. 73–193). Cambridge, MA: MIT Press.

Pollock, J.-Y. (1989). Verb movement, universal grammar and the structure of IP. *Linguistic Inquiry, 20,* 365–424.

Rizzi, L. (1982). *Issues in Italian syntax.* Dordrecht, The Netherlands: Foris.

Rizzi, L. (1989). On the format for parameters. *Behavioral and Brain Sciences, 12*(2), 355–356.

Rizzi, L. (1991). *Residual V2 and the WH-criterion* (Tech. Rep. in Formal and Computational Linguistics, No. 2). Geneva: Université de Genève, Département de Linguistique.

Rizzi, L., & Roberts, I. (1989). Complex inversion in French. *Probus, 1,* 1–30.

Roeper, T. (1973). Theoretical implications of word order, topicalization and inflections in German child language. In C. Ferguson & D. Slobin (Eds.), *Studies of child language development* (pp. 541–554). New York: Holt, Rinehart & Winston.

Roeper, T., & Nishigauchi, T. (1987). Deductive parameters and the growth of empty categories. In T. Roeper & E. Williams (Eds.), *Parameter setting* (pp. 91–121). Dordrecht, The Netherlands: Reidel.

Rumelhart, D., & McClelland, J. (1986). *Parallel distributed processing: Explorations in the microstructure of cognition: Vol. 1. Foundations.* Cambridge, MA: MIT Press.

Schaffer, J. D. (Ed.). (1989). *Proceedings of the Third International Conference on Genetic Algorithms.* San Mateo, CA: Morgan Kaufmann.

Schraudolph, N., & Belew, R. (1990). *Dynamic parameter encoding for genetic algorithms* (CSE Tech. Rep. No. CS90-175). San Diego: University of California.

Simon, H. (1982). *Models of bounded rationality: Vol. 2. Behavioral economics and business organization.* Cambridge, MA: MIT Press.

Spiess, E. (1989). *Genes in populations* (2nd ed.). New York: Wiley.

Stromswold, K. (1990). *Learnability and the acquisition of auxiliaries.* Unpublished doctoral dissertation, Massachusetts Institute of Technology Cambridge.

Toman, J. (1984). *Studies in German grammar.* Dordrecht, The Netherlands: Foris.

Travis, L. (1984). *Parameters and the effects of word order change.* Unpublished doctoral dissertation, Massachusetts Institute of Technology, Cambridge.

Waltz, D., & Pollack, J. (1988). Massively parallel parsing: A strongly interactive model of natural language interpretation. In D. Waltz & J. Feldman (Eds.), *Connectionist models and their implications: Readings from cognitive science* (pp. 181–204). Norwood, NJ: Ablex.

Wexler, K., & Culicover, P. (1980). *Formal principles of language acquisition.* Cambridge, MA: MIT Press.

Wexler, K., & Manzini, R. (1987). Parameters and learnability in binding theory. In T. Roeper & E. Williams (Eds.), *Parameter setting* (pp. 41–76). Dordrecht, The Netherlands: Reidel.

White, L. (1982). *Grammatical theory and language acquisition.* Dordrecht, The Netherlands: Reidel.

Yang, D. W. (1983). The extended binding theory of anaphors. *Language Research, 19(2).*