# 23 Setting Syntactic Parameters

JANET DEAN FODOR

## 1 Learnability Concerns

The study of language learnability is concerned with the "logical problem of language acquisition" (Baker and McCarthy 1981). This is the problem of how it is possible in principle to acquire a language, under various assumptions about the learning mechanism and the information provided by the environment. Some studies are very abstract (e.g. Gold 1967). Others approach more closely the properties of natural languages and human psychology, and the nature of a normal child's exposure to language (e.g. Pinker 1984). Realistic models are of the most interest, but are thin on the ground at this still early stage of the discipline.

Given that we are all living proof that natural language learning is possible, what questions could arise about learnability in principle? Chomsky (1965 and elsewhere) drew attention to "the poverty of the stimulus," the fact that the environment provides less information than the eventual adult grammar contains. This has been a key argument for the existence of innate linguistic knowledge, which must apparently substitute for the missing environmental information. The stimulus for language learning is impoverished in a number of ways. The sentences children hear (or digest) are typically simpler than those they will produce and understand as adults. Negative data, concerning what is ungrammatical in the target language, are largely absent (Marcus 1993). A child might hear *Mice often eat cheese* but no one bothers to mention that **Mice eat often cheese* is unacceptable. The language sample may include ungrammatical and incomplete sentences, idioms and exceptions, all of which could invite learners to posit overgenerating grammars. A parent's elliptical imperative *No pushing, please* should not be taken as a general model for imperative formation permitting also **Much giving me cookies, please*. Ambiguous sentences, if wrongly structured by the learner, could also lead to incorrect grammars. In English *The mouse saw the cat* means the mouse did the seeing, but a learner who mistook it to mean that the cat saw the mouse could conclude

that English allows object-verb-subject word order. Learnability theory has concerned itself with each of these problems in turn.

Learning complex structures from simple input was the focus of the degree *n* research of the 1970s. The sentences of an adult language are unbounded in length and degree of embedding. Can they be projected from a subset of the language which is limited to *n* degrees of clausal embedding? Working within the Standard Theory of transformational grammar (Chomsky 1965), Hamburger, Wexler and Culicover (reported in Wexler and Culicover 1980) demonstrated degree 2 learning, given certain universal constraints of independent linguistic interest. Following this heroic work, and taking advantage of even more stringent constraints within Government Binding (GB) theory (Chomsky 1981, 1986a), Lightfoot (1989) argued that *n* can be reduced to zero "plus a little bit": the most that needs to be observed of an embedded clause is its complementizer and the subject, with all else following by general principles. (For discussion, see the commentaries following Lightfoot's paper.)

The lack of systematic negative evidence took over as the issue of main concern in the 1980s, with its moral that learning must be conservative. Changes made to the learner's developing grammar should obey the Subset Principle (SP): the language generated by the newly hypothesized grammar should be no larger than necessary to accommodate the learner's input. (This idea is evident in Gold 1967, and articulated by Angluin 1980, Berwick 1985, and others.) SP imposes a default which learners must apply, for safety reasons, when the evidence is not decisive. For example, a learner who has so far heard subjects only before verbs should assume that subjects *must* precede verbs, even though there are languages (like Italian) in which the subject may either precede or follow the verb. The standard argument for the claim that SP is necessary for successful learning is that a wrong grammar which generates a proper superset of the target language cannot be recognized as wrong without negative evidence. Examples were presented to show that the problem is real: natural languages do stand in subset/superset relations one to another (Manzini and Wexler 1987). However, it emerged that SP is too strong a remedy to match the behavior of human learners. Counter-examples to conservative learning were documented: in some domains (though by no means all) children do posit a superset of the set of well-formed adult sentences, and later retreat to the correct language (Hyams and Sigurjónsdóttir 1990).

Possible retreat mechanisms were outlined which do not require explicit negative information. A negative fact might be innately linked to a positive one and ride piggy-back on it (e.g. ungrammaticality of null subjects linked to presence of an overt expletive; Hyams 1986). The non-occurrence of a construction in an expected context might be taken as evidence of its non-existence (Chomsky 1981: ch. 1). The existence of a competing construction with the same meaning could also signal ungrammaticality, if learners rely on a preemption mechanism such as the Uniqueness Principle (Clark 1987).

SP is also too strong to allow learners to formulate valid generalizations captured by adult grammars. Under its influence, input sentences which manifest

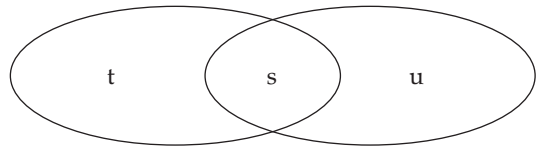


**Figure 23.1** A non-subset cross-grammar ambiguity

a broad syntactic generalization would be absorbed piecemeal into the learner's grammar as a collection of individual constructional idioms. A truly conservative SP learner would be required to posit hordes of idioms and would never attain the simple generalization. (For discussion see Fodor 1994.) To avoid this misprediction, it might be assumed that learners will jettison SP in order to keep their grammars simple. But then, how is it that restricted language phenomena are transmitted from generation to generation without being overgeneralized on the way? Alternatively, it may be that learners are somehow innately equipped to know which examples they should acquire individually, and which they can and should generalize even though a more conservative grammar is possible. This would be a more elaborate variant of the traditional SP, possibly reflecting the different generalizing tendencies of different components of the grammar: morphology, lexical subcategorization, non-lexical syntax.

Many issues relating to the lack of negative data and the avoidance of (or retreat from) overgeneration have still not been fully resolved. (For instance, SP as it is usually construed is too weak as well as too strong, since two learning steps each in accord with SP can result in a superset of the target language; see Clark 1988.) But interest shifted in the 1990s to another inadequacy of learners' input: the fact that some sentence types can be generated by more than one grammar. This problem of cross-grammar ambiguity will be the focus of the remainder of this chapter, so I need not detail it here. It may be useful, though, to note how the study of cross-grammar ambiguity relates to the more familiar SP studies.

Cross-grammar ambiguity occurs when a sentence construction (setting aside here its particular lexical content) is compatible with more than one possible grammar, so that a learner cannot tell from encounter with the sentence which grammar licensed it. A subset situation is just one case of this, in fact the extreme case: *every* sentence belonging to the subset language (i.e., to both languages) of a subset/superset pair is an instance of cross-grammar ambiguity. This kind of ambiguity is dangerous. As we have noted, a superset guess, if incorrect, would never be eliminated by unambiguous evidence requiring the subset language. SP is the safe way of resolving this kind of ambiguity. By contrast, it was widely held that no harm could result from guessing freely in non-subset situations where an ambiguous sentence (*s* in figure 23.1) is

shared by intersecting languages, each containing at least one sentence (*t*, *u*) not contained in the other.

If the correct grammar is guessed, all is well. If the wrong grammar is guessed, it can later be switched to the right one when an unambiguous input such as *t* is encountered. There is no simple general principle (like SP) that a wise learner could apply to resolve this kind of ambiguity, so it is just as well that no real harm can come from guessing in this situation. However, this is a case where the logical problem of language acquisition may underestimate the psychological problem of language acquisition. Though the correct grammar may be achieved eventually, the learner will have a wrong grammar (which both over- and undergenerates) in the meantime. And how long the meantime is depends on how much overlap there is between the target language and its competitors, and how often unambiguous sentences occur in the input sample. Moreover, the target is likely to overlap in a number of different respects with a number of different competing languages, so that many wrong hypotheses in need of eventual repair must be juggled at once.

The realization that non-subset grammar ambiguity is a serious practical problem has been slowly filtering into learnability studies, propelled by the insight of Clark (1992a and elsewhere) and the demonstration by Gibson and Wexler (1994). Clark has emphasized the extent of this kind of ambiguity: it is characteristic of the principles-and-parameters (P&P) theory of language that parameter values interact in complex ways in the derivations of sentences. The penalties for failing to unravel these interactions correctly may be severe. Clark showed that "harmless" temporary errors can feed further errors. This is because an incorrect grammar hypothesis may affect how subsequent inputs are structured by the learner, so that an unambiguous sentence which ought to correct the error may look like evidence for some other grammar instead. Gibson and Wexler showed that even where unambiguous evidence is available for correcting a wrong grammar guess, the learner may have gone so far astray that recovery of the correct grammar is not possible, given certain limits on how much a grammar may be changed on a single learning trial. As will be discussed below, this problem is compounded by the long-standing assumption of a simple learning mechanism which is capable of using input evidence only to disqualify wrong grammars, not as a constructive guide toward the correct grammar. If human learners were designed like this, they would have to resort to guessing a grammar even where there is enough information to make guessing unnecessary.

The agenda for learnability research must therefore include an investigation of how extensive cross-grammar ambiguity is, and how human learners manage to outsmart it. Are there effective strategies which limit the amount of misdirection due to ambiguous input? Can learners differentiate ambiguous from unambiguous input (or subset ambiguities from non-subset ambiguities) and apply strategies relevant to each? Is it inevitable that strategies which limit the randomness of guesses in ambiguous situations will render some grammars unlearnable, as in Gibson and Wexler's simulation?

These questions largely postdate the theoretical shift from rule based to principle based grammars, which created a watershed in learnability research by enabling parameter setting as the primary means for grammar acquisition. So cross-grammar ambiguity problems have mostly been formulated in terms of ambiguity of the triggers for parameter setting. Linguists and acquisition researchers have attempted to identify unambiguous triggers for all parameters postulated as part of Universal Grammar (UG). Until the seriousness of the ambiguity problem came to light, there was considerable optimism that the parametric model had left most learnability problems in the past. Previously, it had had to be supposed that children must devise rules and constraints to capture generalizations about the sentences they hear. But with the P&P theory, language learning appeared instead to be just a simple quiz. Does the target language permit or not permit phonologically null subjects? Do heads of constituents precede or follow their complements? Do interrogative phrases move (overtly) to clause initial position or remain in situ? As we will see, it matters how many such questions there are for learners to answer. It is standardly assumed that there are fewer parameters than there are possible rules in a rule based framework; otherwise, it would be less obvious that the amount of learning to be done is reduced in a parametric framework. A goal of linguistic research has been to consolidate facts and posit as few parameters as possible consistent with crosslanguage variation. It might turn out that there are 20 parameters or 30 or 100 and more. Only continued linguistic research will tell. On one recent estimate (Cinque 1999) there would be at least 32 parameters controlling the landing site for verb movement, perhaps multiplied by the number of possible verb forms (finite/infinitive/past participle, etc.). But I will make the working assumption here that there are exactly 20 binary syntactic parameters. This modest estimate is quite sufficient to raise all the questions of interest about how parameter setting could succeed in face of input ambiguity.

## 2 Exponential Facts of Life

### *2.1 Exponential reduction*

The most welcome aspect of parameter theory for learnability research is the economy of descriptive means relative to the wealth of languages described. How many natural languages are there? Clearly a very great many, even if we set aside all differences in phonology, morphology, and the lexicon, and think only of syntactic structure. In what follows I will take a language to be an infinite set of structural descriptions of sentences, I will assume that each language in this sense is defined by a unique grammar, and I will focus on syntactic structure, using "grammar" as shorthand for "syntactic component of a grammar." If the number of languages is $L$, the minimum number of binary parameters there could be is $n$, where $n$ is the smallest integer such that $2^n \geq L$.

For any plausible value for *L*, *n* is very much smaller than *L*. This is why it is important if learners can indeed distinguish which of *L* languages they are hearing by answering just *n* simple questions about it.

Because the relationship is exponential, the bigger L is, the greater the reduction the parameterization brings. If there are a thousand languages, *n* can be as low as 10, a reduction factor of 100. For a million languages, *n* need be only 20, a reduction factor of 50,000. For a billion languages, *n* is 30, so L is reduced by a factor greater than 30 million. In fact, the numbers are not quite this favorable unless the *n*-dimensional parametric space has no holes in it; that is, unless all parametric distinctions are fully orthogonal to all others, and L = exactly $2^n$. But natural languages do not fully exploit the parameter space. Some parameters are inapplicable to some languages, due to incompatibility with their other parameter values or lexicon. For example, a parameter distinguishing single from multiple overt *wh*-movement is not applicable to a language whose other parameter values exclude overt *wh*-movement. A nonconfigurational language with free word order, like Warlpiri or Mohawk, will have no need of values for the standard word order parameters. For such languages there is probably no answer to the question whether the object (always *pro*, bound by the lexical phrase in adjunct position) precedes or follows the verb (see Baker 1996). Non-relevance of some parameters to some languages is of interest in learnability research, but can be largely ignored here until section 6.

It will be taken for granted here that the program of capturing natural language differences by a set of binary choices is descriptively successful and, more strongly, that it truly reflects the nature of UG. If so, then it seems that all that a child has to do to acquire any one of a million languages is to sit and listen for 20 sentences, each of which will reveal the value of one parameter. Over the first three years a child hears very approximately 2,500,000 sentences, or more than 2,000 per day, though this does not distinguish between those the child digests and those she or he merely overhears (Geoffrey Pullum, personal communication, based on statistics from Hart and Risley 1995). Since every normal child succeeds in acquiring, more or less exactly, the language to which she or he is exposed, we know that somewhere among the first five (or six or seven) million sentences a child hears there is sufficient information to determine, in conjunction with the information in UG, the correct set of parameter values for the target language. The child's only task is to extract that information from the sentences.

If there were a complaint to be raised against the parameter setting model it might be that it trivializes language acquisition. If learning is as easy as that, how could it take so long? There are some plausible answers to this. Factors such as processing limitations and the need for lexical learning would slow down an inherently efficient syntax learning device. But these need not detain us, because the real puzzle is not why real-life parameter setting is not quite as easy as this "20 questions" metaphor might suggest, but why the 20 questions mode of learning is so difficult. It is so difficult that nearly two decades after Chomsky proposed it, computational linguists and psycholinguists are still

struggling to implement it in a way that is consonant with the resources of a normal child.

### *2.2 Exponential re-explosion*

What all discussions overlooked, when Chomsky gave us this elegantly simple concept, was that answering a single parametric question might be as laborious (though in a different way) as hypothesizing and testing a rule was in previous learning models. It was all too easy to take it for granted that each of the small finite number of questions could be answered with a small finite amount of effort. But in fact, on perfectly plausible assumptions, reviewed below, the workload per question can be exponentially related to the number of questions there are. That is: though the exponential reduction from $L$ languages to $n$ parameters still holds, there is an opposite and almost equal exponential explosion from the number of parameters to the number of learning steps to set them, so that the latter is on the order of $L$ or worse (Clark 1994). If so, the learner might just as well check out each grammar, one by one, against the input; nothing has been gained by the parameterization.

The belated recognition of this fact is what is now driving research on ways to implement parameter setting, in the hope of finding one that is relatively immune to the problems of scale that exponential complexity creates. Some ideas are discussed below. But first let us consider how compelling the evidence is that in the case of natural language, a learner cannot simply extract 20 bits of information from the language sample at a modest constant cost per item.

## 3 Parametric Ambiguity

### *3.1 Ambiguity and workload*

To study the cost-per-parameter problem we need a measure of the learner's workload. As a rough measure that provides a common ground across otherwise different models, let us identify the workload with the number of input sentences that must be processed by the learner before learning is complete (that is, by the time the learner has settled permanently on a grammar identical, or sufficiently similar, to the target). The more sentences consumed, the slower and more laborious the acquisition process. Some learning systems may put in more work on each input sentence than others do. But if there is a practical limit on how much work a child could do per sentence before moving onto the next one in a discourse, the measure of sentences consumed is not unuseful even for such systems.

A simple argument leads to the alarming conclusion that to set one parameter could cost the learner thousands or millions of input sentences. The argument rests on the fact that the learner's language sample is a set of word

strings, while syntactic parameters determine sentence *structure*. Because a string may be compatible with more than one structure, the input can be indeterminate with respect to the structural properties that the learner must have access to for parameter setting. It seems plausible that the difficulty of setting parameters is a function of how structurally indeterminate (on average) an input word string is; that is, how many distinct structural descriptions it could have. The learner's task is to identify the structural description it has in the target language. The more others it could be assigned, the greater the opportunity for errors of parameter setting; or alternatively, the greater the effort required to avoid errors.

The number of structural descriptions an input sentence could have is in the worst case a function of how many grammars there are; that is, it is bounded by $L$, not $n$. Each grammar might, in principle, assign a sentence a different derivation. So if there are 20 parameters, there could be a million or so different structural descriptions for any target language sentence, each corresponding to a different array of parameter settings. Of course this is not the least bit likely in reality. But it is important to recognize that this is the trend, the direction in which the numbers will drift in the worst case.

Consider a simple example: a sequence consisting of just a verb followed by its subject. This sequence does not have a million derivations. It is not licensed by all natural language grammars, but it is licensed by many. In some cases the parametrically relevant structure is the same. There are clusters of grammars which differ with respect to other parameters but which are alike with respect to the parameters relevant to licensing this sentence type (for example, grammars that differ only with respect to object position or the acceptability of headless relative clauses, etc.). But also, there are grammars which license a VS sentence under different parametric descriptions. For simplicity here we may bend the language facts a little and suppose there are just three relevant parameters: one that controls postposing of a subject to follow the verb, as in Italian (Burzio 1986); one that controls raising (fronting) of the verb to the Inflection position, while the subject remains in its underlying position within VP, as in Irish and Welsh (Koopman and Sportiche 1991); and one that controls raising of the verb to the Complementizer position, as in German yes/no questions, where the subject remains lower as Specifier of the Inflection phrase (Taraldsen 1986). With another stretch of the imagination we may suppose that the three parameter values which give VS order are not mutually exclusive: two or more of them may be at work in the same language (as indeed appears to be the case in Bantu languages like Shona and Swahili; see Demuth and Harford in press). In that case there would be seven ways to obtain a VS string, even if the underlying order is SV (Kayne 1994): by the parameter setting for subject postposing, by the setting for verb to I, by the setting for verb to C, by any pair of these in the same grammar, or by all three together; only the negative value for all three in the same grammar would fail to license VS order. The number of potential analyses here is not quite $2^3$, but it is bounded by $2^3$, not by 3. Thus, the parametric indeterminacy of any target

sentence can rise *exponentially* with the number of parameters in the language domain.

Can this estimate of the degree of potential parametric ambiguity be resisted? If not, we are heading breakneck toward the conclusion that setting even one parameter can be exceedingly costly. We must re-examine the premises from which the estimate was derived. A central one is that learners' input consists only of strings, not of syntactic structures. In fact this is too severe. Let us re-consider it, along with other standard assumptions about the nature of the input for learning. These are simplistic and too extreme, but they facilitate formal work on learnability problems. They include the following:

i Learners consult only one sentence at a time (and have no memory for prior sentences), and they do not have access to negative evidence.

ii For each language there is only one correct grammar, and the sample a learner receives is compatible only with that one grammar; that is, the input suffices to determine the target parameter settings. (Bertolo et al. 1997 and Fodor in press discuss special cases.) As part of this I will assume here, though it is unrealistic, that all sentences in the learner's sample are well formed in the target language.

iii A learner sets syntactic parameters only on the basis of sentences all of whose lexical items are known. (For discussion of problems see Stabler 1998, Fodor in press.)

iv With more bearing on the ambiguity issue, we may follow Gibson and Wexler (1994) and others in taking the input to be something more than word strings though less than full structural descriptions. It is commonly assumed that the words have been lexically categorized into nouns, verbs, determiners, etc., and that the learner knows the grammatical roles of constituents; for example, that an English learner knows that in *The cat saw the mouse, the cat* is the subject of *saw*, and *the mouse* is its object. Modifying a stronger assumption by Wexler and Culicover (1980), I will assume that a child can determine part or all of the meaning of a sentence from the verbal or non-verbal context and will not accept a syntactic analysis which contradicts that meaning.

v Learners also use the prosodic contour to constrain the syntactic analysis of the word string (Morgan 1986). Recent work on infant perception of sentence prosody makes an excellent case for this as a practical possibility (see Nespor et al. 1996, papers in Morgan and Demuth 1996, and references there). Though prosodic phrasing does not faithfully reflect all aspects of syntactic phrasing, sensitivity to prosody implies that input strings are at least partially hierarchically structured.

On the basis of (iv) and (v) the extreme estimate of potential parametric ambiguity can be toned down. Though this will not be emphasized in the discussion below, it seems likely that semantic and prosodic information can significantly shrink the structural indeterminacy of input strings and thereby

facilitate syntactic learning. (This is different, however, from the more dramatic claim implied by Mazuka 1996 and Nespor et al. 1996 that syntactic parameters may be prosodically triggered. This strikes me as less plausible, and the existing empirical evidence does not favor one hypothesis over the other.) On the other hand, the literature contains some examples of ambiguity for which neither prosody nor meaning offers significant assistance. Clark (1988) noted that an accusative subject of an infinitival complement is compatible with either Exceptional Case Marking (ECM) or structural assignment of case in infinitives. Gibson and Wexler (1994) observed that SVO word order is compatible with either the positive value of the Verb Second (V2) parameter (as in German *Die Mäuse sahen die Katze*) or the negative value (as in English *The mice saw the cat*). In both cases the meaning and prosodic contour for the competing analyses can be essentially indistinguishable. (V2 constructions permit but do not require a prosodic break before the verb.)

It seems fair to conclude, then, that the problem of parametric ambiguity does not rest solely on the simplistic assumption that learners hear only unstructured word strings. An ambiguity problem remains, even with a more inclusive concept of learners' input.

## *3.2 Younger learners work harder?*

If a worst-case exponential relation between the number of parameters and the extent of parametric ambiguity is not assailable, the only point at which the exponential workload argument might be deflected is the postulate that parametric ambiguity *must* complicate the answering of parametric questions. That assumption also seems indisputable, but our goal must be to find a way around it if there is one. In section 4 I will argue that there is no general formula for escaping the impact of ambiguity. It depends on the particular parametric decoding procedure that a learning model employs. It also depends on how cooperative the language facts are: the structural characteristics of languages could be such as to minimize parametric ambiguity in the kinds of sentence that children typically learn from even if the rest of a language were highly ambiguous. However, first we should take a deeper look at the extent of the problem. Two points need to be made. One adds to the ambiguity load; the other can decrease it.

A plausible assumption, which will be important below, is that every syntactic parameter that contributes to the licensing of a word string does so via its effect on the structural description of the string. This is the case, for example, for the ±verb second ambiguity of SVO strings, where *The mice saw the cat* is –V2, while *Die Mäuse sahen die Katze* is +V2. On standard (though not undisputed) assumptions, the +V2 analysis has the verb in C and the subject as its Specifier, while on the –V2 analysis the verb is in some head position lower than C. In other words: parametric ambiguity is associated with structural ambiguity. We may conjecture that this is always so (see Fodor 1998a for discussion). Still, parametric ambiguity is distinct from structural ambiguity

relative to a single grammar, such as in English *Flying planes can be dangerous*. A learner *qua* learner does not care about within-language ambiguity. As long as the right language has been hypothesized, that is sufficient, whether the particular analysis that was intended by the speaker was retrieved or not. Nevertheless, cross-grammar and within-grammar ambiguity can be difficult to tell apart, particularly when one is a child and does not yet know what the target grammar is. So within-language ambiguity may interfere with parameter setting. The extent of this problem is not known. Without at all underestimating its potentially damaging effects, I must set it aside here.

Second: what matters for learning is not how structurally ambiguous a string is relative to all possible grammars, but how ambiguous it is relative to grammars that the learner *has not yet excluded* as incorrect for the target language. I will call the former *gross* ambiguity and the latter *net* ambiguity. Gross ambiguity is a fact about the sentences in the learner's language sample, in relation to the domain of possible languages. (For convenience we may assume here that gross ambiguity is evenly distributed across the sentences of the sample, though in fact there is likely to be some variability; see section 6.) Net ambiguity, on the other hand, is a fact about the learner's state of knowledge as well as the language sample. It represents the uncertainty still to be eliminated before learning is complete. If learners set parameters decisively, and discard for ever the disconfirmed values, then net ambiguity will decline across the course of learning as more and more parameters are set. If the same sentence is encountered by a child at two years and again at four years, its net ambiguity will not be the same on the two occasions because the child's grammar will have advanced in the meantime. What determines learning effort is presumably net ambiguity; that is, how many structural descriptions an input sentence could have *for all the learner now knows*. If this is right, it leads to the important conclusion that a beginning learner must work harder to set a parameter than a more advanced learner would to set that same parameter.

The net ambiguity of a word string (i.e., the number of distinct structural analyses it has on the basis of grammars not yet excluded by the learner) can in principle be as high as $2^{(n-p)}$, where $n$ is the number of parameters in the domain (all relevant to licensing the target language; see section 2.1) and $p$ of them have so far been set (correctly) by the learner. In a domain of 20 parameters, the net ambiguity of an input could be as high as 1,048,576 (the total number of possible grammars) at the outset of learning. By the time all but one of the parameters have been set, net ambiguity would be at most 2. Thus, the first parameter setting event faces a potential net ambiguity up to half a million times higher than the potential net ambiguity for the last one (regardless of which parameters the learner happens to set first and last). The curve is the familiar exponential decline: for setting the second parameter the maximum degree of ambiguity would be half of that for setting the first; for setting the third it would be half of that; and so on. If the total number of parameters to be set is greater, the disparity between first and last is greater still. If there were 50 relevant parameters, the multiplier would be $2^{49}$, which is up in the

trillions. If some parameters are irrelevant to the target language, the disparity is less; for 15 relevant parameters the maximum ambiguity for setting the first is only 16,384 times higher than for setting the last. It is clear, though, that for any plausible number of parameters to be set, parameter setting difficulty is far from uniform across the course of learning if it depends on the degree of net parametric ambiguity of sentences.

The cheerful way to put this is: the task gets easier and easier as time goes on. The more parameters you set, the easier it becomes to set more of them; the more you know, the faster you learn. The disturbing side of it is: the learning task is a great deal more onerous at first than it is later on. The less you know the harder it is to learn. However small the cost of setting the last parameter may be, that cost magnified a thousand- or a million-fold for the setting of the first parameter is bound to add up to something unmanageable.

In summary: if the learner's workload is a function of parametric ambiguity, then the exponential reduction of $L$ grammars down to $n$ binary parameters re-explodes into estimates on the order of $L$ for the cost of setting each parameter, at least at early stages of learning when few parameters have been set. This means that the greatest burden of learning is concentrated at a time when learners presumably have the fewest resources and are in need of the greatest assistance from the input. Once parameter setting is underway it may proceed efficiently, but early ambiguity is potentially so extreme that it is hard to see how learners ever get started. Thus, parameter setting is not a feasible means of language acquisition unless we can free it somehow from sensitivity to parametric ambiguity.

## 4 Parametric Decoding

Learning a language is as easy – or as difficult – as answering 20 questions. How easy that is depends on whether parametric ambiguity is the major determinant of how much work it takes to find out the answers. If parametric ambiguity is what paces parameter setting, then it is a mystery how learners manage to set their first parameter. I will consider this mystery from the psychocomputational modeler's point of view rather than the empirical study of children point of view. Later, I will consider briefly how well the two fit together.

The goal is to create a blueprint for a learning system that can extract from natural language sentences the information necessary to set 20 syntactic parameters, consuming only a reasonable number of input sentences, spread in a reasonable way over the course of learning. To do this we must find either (i) another factor which favors early learners and offsets their disadvantage with respect to ambiguity, or (ii) a method for parameter setting that is relatively insensitive to high degrees of ambiguity at any stage. I will proceed along path (ii) here. For mathematical convenience I will for the most part be treating parameters as anonymous entities with no particular linguistic content. Each

is as likely to be expressed by a sentence as any other is; each is as likely to be expressed ambiguously as any other is. In fact, there can be considerable variability in these respects. To allow for this in a more interesting and realistic model entails looking at the character of natural language sentences and the relation between sentences and the grammars that license them. I take this up in section 6.

## *4.1 Decoding ambiguous input*

Ideally, when the learning system encounters a novel kind of sentence (ambiguous or otherwise) it would know exactly which parameter values entered into the derivation by which that sentence was licensed in the grammar of its utterer. Parametric ambiguity makes this impossible, of course; the child is still trying to discern how the local adults are licensing their sentences. But at least the learner would benefit from knowing which parameter value combinations *could* have licensed that sentence. Establishing this is what I call *parametric decoding*. For example, on hearing an SVO string, the decoding device would inform the learner that it could have been derived with parameter values –V2, complement final, Specifier initial (as in English), or with +V2, complement initial, Specifier initial (as in German), or with +V2, complement final, Specifier initial (as in Swedish), or with +V2 and Specifier final, and so on. (This parameterization follows Gibson and Wexler 1994, who assumed, unlike Kayne 1994, that underlying word order differs across languages.) Encountering a parametrically unambiguous string, the decoder would report a unique parametric profile; for example, for a sequence of indirect object, finite auxiliary, subject, direct object, and thematic verb (*Den Mäusen habe ich den Käse gegeben* "I gave the cheese to the mice"), it would report the single combination: +V2, complement initial, Specifier initial (if indeed this word order is parametrically unambiguous; we may pretend that it is for now).

Once the parametric properties of an input have been decoded, the learner could follow a strategy of adopting any parameter values that are common to all analyses, in confidence that these values must be in the target grammar (since they are values without which the sentence could not have been generated). For this ideal operating system, learning would be complete as soon as each target parameter value had been unambiguously realized (*expressed*) in the learner's input. Thus, efficient parameter setting relies on efficient parametric decoding. In fact, the efficiency of decoding is seriously threatened by ambiguity.

There is a widespread view that each parameter value is associated with a "cue" which can be identified in a sentence and which then "triggers" the adoption of that parameter value. With this in mind, I examine below (section 4.2) some proposals in the literature. They divide roughly into the optimistic ones, which believe it can be done though they do not actually say how, and the pessimistic ones, which have decided that it is hopeless. The latter assume that learners do not even try to "read" parameter values accurately from

sentences but operate by trial and error instead. Anyone who has read the computational literature, and been puzzled by the lack of resemblance between recent learning models and the classical picture of setting parameters by triggering, should note that this is not for lack of interest or enthusiasm for the idea of triggering. It is due largely to the difficulty of modeling the parametric decoding process that triggering presupposes. Decoding problems have had a profound effect on the directions that learnability theory has taken. In section 4.3 I describe a novel decoding method which preserves the essence of classical parameter triggering. Though somewhat unconventional, it is accurate and efficient for parametrically unambiguous input. How it can best be adapted for ambiguous input is the topic of section 5.

At best, the decoder merely presents the options, if an input sentence is parametrically ambiguous. Deciding between them is the task of the learning component. The learner might wait for a sentence which unambiguously expresses the parameter in question. But the wait may be a long one. It is even possible that *no* sentence expresses the target value unambiguously, even though it is unambiguous with respect to the language as a whole. As a simple example: in Gibson and Wexler's three-parameter domain, +V2 must be correct if some target sentences could be derived by +V2 or by −V2 with underlying SVO, and the rest could be derived by +V2 or by −V2 with underlying SOV. Thus it seems that a learner must somehow triangulate from the multiple parametric combinations for each individual sentence, to find the unique one that is common to all the sentences in the sample. This is how a linguist might go about the task. But a child is not a little linguist, and from a child's perspective there is a two-fold problem with this approach. A real-life learner cannot afford to wait to hear the whole sample before having any grammar to use for comprehension and production. And the chore of making all the cross-comparisons between large sets of parameter value combinations would be enormous. So, although this is the *logic* of the answer to parametric ambiguity, we must hope there are other ways for learners to actually go about finding the common denominator across sentences. A goal of computational psycholinguistics is to devise a way that is effective, is not too labor intensive, and is incremental, able to make progress one sentence at a time.

This is where decoding and ambiguity interact. The decoding system is the gateway through which information about sentences reaches the parameter setting system. The learner's range of options for dealing with ambiguity is limited by what the decoder can deliver. The major issue turns out to be whether or not the work of identifying a set of parameter values that can license a sentence is so effortful that it cannot reasonably be done more than once per sentence. This is important because if multiple decoding is *not* feasible, then a learner cannot know whether a sentence is parametrically ambiguous. The only way to deal with ambiguity then would be by ignoring it. The decoder would deliver just one set of parameter values for an ambiguous sentence. The learner would have no choice but to accept the information as if it were derived from an unambiguous sentence and would adopt those values, quite possibly

incorrectly. If multiple decoding is feasible but only up to some limit, learners would be little better off, because incomplete decoding would still not distinguish reliably between ambiguous and unambiguous sentences. Only if it is able to check out *all* possible ways of licensing a string could the learner tell reliably (in the worst case) whether more than one way exists.

Thus, nothing short of total decoding of all ways of licensing a sentence would have to be feasible if learners are to sort ambiguous from unambiguous input accurately enough never to set a parameter on the basis of an ambiguous string. However, with ambiguity levels as high as they are in natural language (even if a million grammars per sentence is a vast exaggeration), full decoding is not a serious possibility. In order to abbreviate the discussion below, I impose here and now a blanket ban on any decoding scheme which presumes that a child analyzes (parses) and reanalyzes the same utterance more than a dozen times, or that a child conducts a single parallel parse in which more than a dozen analyses are computed simultaneously. Decoding on such a scale is not psychologically realistic. Yet as we have seen, decoding on any lesser scale provides the parameter setting device with insufficient information to do its job of triangulation accurately.

In summary: parametric ambiguity puts a tremendously heavy strain on the decoding system. If the decoding system cannot rise to the challenge, then the learner's task of finding the unique set of parameter values for the whole language, which is already substantial, is further hampered by uncertainty about the range of candidate values for individual sentences. High-precision parameter setting is then not possible.

## *4.2 Decoding methods*

There was a time when we thought we knew how learners decode the parametric signatures of sentences. According to the familiar metaphor, attributed by Chomsky (1986a) to James Higginbotham, parameter setting is effected by automatic flipping of parameter switches by relevant "trigger" sentences. This neat idea holds a special place in the history of the P&P model and it is a shame to have to relinquish it, but it has been tried and found wanting – Computational linguists have turned in recent years to very different mechanisms. Figure 23.2 maps some approaches that have been devised so far. I will describe how they work, and in section 5 I will consider how they respond to heavy doses of parametric ambiguity.

Method (a) is automatic switch flipping. For it to work, each switch must be equipped with a property detector responsive to the trigger property (or properties) for that parameter; that is, the particular properties of sentences which reveal the parameter's value. Since all 20 (or 40, if parameters are not pre-set to default values) detectors check the string at once, it is reasonable to suppose that they do not process it very deeply. If they did, this model would violate the ban (section 4.1) on excessive parallel processing and would be disqualified on that ground. The trigger properties must therefore be superficial and

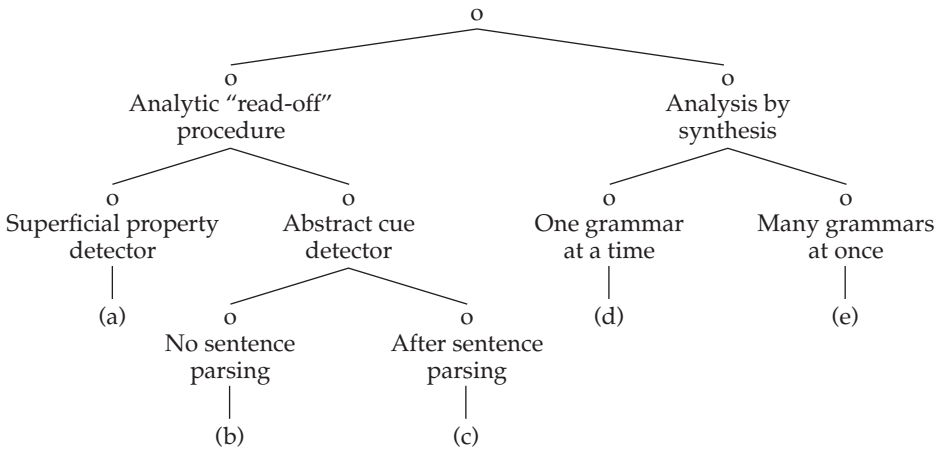


**Figure 23.2** Parametric decoding methods

easily recognizable. And that is simply not true for many natural language parameters. Perhaps it is for some. Perhaps multiple overt *wh*-movement within the same clause is identifiable in a surface word string (though probably even that is not totally reliable: imagine an SOV sentence with two *wh*-arguments). However, in many cases the relevant facts are non-surface facts (e.g. the origins rather than the landing sites of movement) which are likely to be less accessible. Underlying word order is very often obscured by movement operations, and one movement operation may be masked by a later one. So even when derivational operations do not create ambiguity, there may still be no easily perceptible surface sign by which to detect the presence of a particular parameter value (Clark 1994).

Methods (b) and (c) are truer to the facts of natural language since they allow for deeper, more abstract trigger properties, as in the "cue-based theory" of Lightfoot (1997). Lightfoot's account differs from instant triggering in two ways: its cues (equivalent to trigger properties) are abstract "elements of I-language"; and the metaphor of flipping switches gives way to that of the learner "scanning sentences" for the cues. An example of what learners must watch out for is the configuration $_{\text{SpecCP}}$[XP], which Lightfoot proposes as the cue for the positive value of the V2 parameter. Though linguistically more authentic, this approach fails procedurally in just the same way as instant triggering does. Few details are offered in the literature, but in figure 23.2 I have distinguished two possible implementations of abstract cue search. On version (b) without prior sentence parsing, the learner would have to identify the abstract cue structures from unstructured word strings. This will not work. There is no obvious way for a learner (not knowing the right grammar) to identify an XP (say, a DP) in an unstructured word string. And even if it could, it surely could not establish that this phrase is in SpecCP position, rather than in underlying subject position, or adjoined to IP by scrambling, and so forth.

With partially structured input strings as envisaged in (iv) of section 3.1, the hopelessness of this task would be diminished, but not to the point at which it would be a reliable basis for learning.

On version (c), the input is first fully parsed in order to uncover its more abstract derivational properties on the basis of which I-language cues *could* be identified. But this is quite unrealistic because it requires multiple parsings of the sentence. For instance: for purposes of recognizing the I-language cue for the +V2 parameter value, the parser must assign to the string the analysis it would have if it were licensed by the +V2 value (plus appropriate values of other parameters). Only then will the cue $_{\text{SpecCP}}$[XP] be present for the "scanner" to find.

In other words, "scanning a sentence" for the cue for parameter value $P_i$ (v) entails parsing the sentence with parameter value $P_i$ (v). But a sentence cannot normally be parsed with just one parameter value, and since the learner does not yet know what the target parameter values are, it would have to parse the sentence with $P_i$ (v) together with many combinations of values of the other parameters until it found one that succeeded for the sentence – even an unambiguous sentence. For ambiguity detection, it would (in the worst case) have to try out $P_i$ (v) with *all* combinations of the other parameters that have not yet been set. Thus on version (c) of the cue based approach, the learner's workload explodes just as anticipated in section 3. Whether the multiple parses are conducted in parallel or sequentially, they clearly disqualify abstract cue search under the excessive processing criterion. Hence, the recognition of abstract cues is either approximate or infeasible. Reliable decoding is impossible in the absence of an effective way of spotting the cues that are present; see section 5.1 below.

The best-known representative of approach (d) is the Triggering Learning Algorithm (TLA) of Gibson and Wexler (1994). As in the second (postparsing) implementation of abstract cue search, this learner tries out grammars on input strings without knowing in advance which will work. But rather than seeking pre-defined cues for particular parameters, it takes success in parsing the sentence as the mark of whether a grammar is right for the target, or at least as a sign that it shares parameter values with the right grammar. For each input the TLA tries just one new grammar, so it satisfies the processing feasibility criterion. But given that there are a million or more grammars that could need checking, this is inevitably a slow method. In figure 23.2 this approach is filed under analysis by synthesis (ABS) methods, because the TLA does not start by observing the sentence and trying to compute the right parameter values from it; instead, it first picks a combination of parameter values and only then tries them out to see whether or not they are compatible with the sentence. It is well known that ABS methods can be very wasteful of resources if undirected. The chance of hitting on the right answer out of the blue is slim (see Fodor et al. 1974: ch. 6). To increase the efficiency of the TLA, Gibson and Wexler gave it some direction: its guesses as to which grammar to try next are influenced by feedback on the success or failure of previous guesses. Specifically,

its guesses are limited to grammars which differ by no more than one parameter value from the grammar with which it most recently succeeded in parsing an input sentence. So if a grammar is successful for a while before it fails on some new input, many of its parameter values will be preserved in subsequent hypotheses. This strategy of staying in one neighborhood among the class of possible grammars, and gradually moving toward the target, is designed to save the learner from having to try out every possible grammar in the domain.

However, we have noted that parsing a sentence calls for a whole grammarful of parameter values, not just one. And this means that the TLA's positive and negative feedback, provided by success or failure in parsing input, applies to whole grammars; it cannot be attuned very closely to the correctness of individual target parameter values. Suppose, for example, that the learner has tried out a grammar, has found that it fails to parse an input sentence, and then tries parsing that sentence again after flipping parameter P7, which was previously set at the wrong value. P7 is now correctly set, so this is progress and ideally would be rewarded to encourage retention of the new value. However, it is very likely (especially early on in the course of learning) that parameters other than P7 are set wrong, and that the grammar with the correct value for P7 will fail to parse the input sentence for that reason. Hence the tentative shift to the correct value for P7 will be negatively reinforced; the learner is discouraged from making the change. (In this circumstance the TLA reverts to whatever grammar it had previously hypothesized, even though unsuccessful.)

In short: decoding is largely a hit or miss affair for the TLA due to its ABS approach, its evaluation of whole grammars, and the very ragged feedback provided by parsing success or failure. Because finding even one grammar that parses a given input is such a matter of chance, finding more than one per sentence is out of the question, so there is no possibility of ambiguity detection. The TLA therefore disregards ambiguity. It accepts any grammar it finds that works for a sentence, without checking whether others would have too. As a guessing system, it pays the price of parametric ambiguity in errors. And correcting its errors requires repeating the travails of decoding.

Method (e) in figure 23.2 represents learning systems which resemble the TLA in that they try out grammars on sentences and use parsing success as reinforcement, but which work faster by testing batches of many grammars at a time on a single input sentence. The best-known example of this is the genetic algorithm of Clark (1992a) and Clark and Roberts (1993). It records how successfully each grammar tested on a sentence can parse it, it stores the success scores of all the grammars, and it "breeds" the more successful grammars, mingling their parameter values to create a new pool of even better candidates for a next round of testing. Genetic algorithms have been a focus of recent interest for machine learning applications. But massive multiple grammar testing on each input clearly does not meet the feasibility criterion for human language processing. (Nyberg 1992 blends storage of parse success rates, as in a genetic algorithm, with TLA-like search through the grammar space.)

There is also method (f), represented by the Structural Triggers Learner (STL) of Fodor (1995, 1998a), not shown in figure 23.2 because it cuts across the tidy classification. It combines elements of the other approaches with one new twist in parametric decoding. It takes the structural cues of (b) and (c), but instead of *looking* for them in sentences, it *parses* with them as in (d) and (e), checking all parameter values simultaneously as in (a). As I will show, this can give highly efficient parameter decoding for unambiguous sentences and reliable ambiguity detection.

## *4.3 The Structural Triggers decoder*

To explain the STL, let us start from cue search. This assumes that each parameter value is associated with some detectable structural property, an aspect of tree structure. I have proposed in earlier work (Fodor 1995, 1998a) that a parameter value can be *identified* with its structural cue, which I call a *treelet* or *structural trigger*, and which I take to be the deepest manifestation of the parameter value, the source of all its effects on the derivations of sentences that it contributes to. Exactly what that structural essence is may depend on the linguistic theory that is assumed. In the original P&P theory there was no very clear theory of possible parameters, and not all proposed instances took the form of a choice between tree fragments. (For discussion see Fodor 1998d.) In the Minimalist Program (Chomsky 1995b), parameter values are identified with the formal features of functional heads, which control derivational operations. For example, where the structural trigger for +V2 in Lightfoot's model is an XP in SpecCP position, in a Minimalist framework it might be a strong Specifier feature on the C head, which will attract an XP to check it. For –V2 the corresponding feature would be weak. Formal features are (very small) tree fragments, and are not themselves derivable from any deeper fact about the language. So these featural parameter values meet the needs of the STL.

The merging of the roles of parameter value and trigger (cue) into one entity (a treelet) in the STL model is of theoretical interest but is not essential to the success of the STL's parameter decoding system. All that would be lost without it is some conceptual elegance and a modicum of representational economy. More important for learnability is the fact that the decoding method works even if the structural property that defines the parameter in UG is not directly discernible in the learner's input. As long as it leaves its imprint on derivations, however non-transparently, the STL will find it.

This is how the STL works. The treelets constitute an innate (UG-supplied) lexicon of parameter values. Every natural language grammar contains some subset of these treelets, which combine with universal grammar principles and a language-specific lexicon of morphemes and words to license the sentences of the language. The learner's task is to adopt from the universal treelet lexicon the treelets that are correct for the target language. The STL adopts a treelet into its hypothesized grammar just in case it is necessary for parsing an input sentence. Encountering a sentence, the parsing component tries to parse

it employing the learning component's currently hypothesized grammar. If the parse fails at some point, the parsing component is then permitted to draw on any of the innate treelets that have not yet been adopted, adding them temporarily into the learner's working grammar. At least one of the treelets in the innate parametric lexicon must be capable of unblocking the parse (unless the failure is purely lexical, which will not be considered here). If only one treelet does so, it is evidently necessary for licensing the target language and so it is added into the learner's grammar. From then on, it can be used to produce new sentences, and to parse incoming ones.

Note that a similar procedure could work if a parameter value were identified with something less concrete than a tree fragment, such as a phrase structure rule, or possibly some sort of abstract statement from which the legitimacy of such a treelet would follow; differences of this sort do not matter. The one crucial requirement is that the parsing mechanism, when it finds itself unable to continue the parse, should be able to identify efficiently any parameter value(s) capable of supplying the missing part of the parse tree so that forward parsing can proceed. We know that the parser can do this very efficiently in general, when the existing grammar suffices for the sentence. It rummages through its collection of tree-building devices to find what is needed to connect each word of the sentence into the parse tree. The STL merely extends this to include the small number of additional tree-building aids that constitute the parameter values.

The STL is representationally economical. If there are *n* parameters, then as few as *n* small tree fragments or features need to be innately represented to characterize the set of UG-defined parameter values/triggers. For 20 parameters, the innate treelet lexicon would contain 40 entries if a parameter is a choice between two treelets (e.g. a strong or a weak feature), either one of which may be adopted into the grammar of a particular natural language. There would be only 20 items in the parametric lexicon if a parameter is a choice between adopting a certain treelet (a strong feature) or not adopting it. I will not adjudicate between these two possibilities here. The STL is also procedurally economical, since it tries out all possible parameter value combinations in a single serial parse (see discussion below), using a grammar which is a perfectly normal natural language grammar such as the human sentence parser works with all the time, except only that it contains more of the UG-provided treelets than (adult) natural language grammars normally do. It is not even necessary for the learner to go through the two-step process of trying to parse an input with the currently hypothesized grammar, failing, and then parsing again using the extra treelets. The STL can just as well apply the treelet-augmented grammar right away to every sentence it encounters, as long as the treelets already adopted are given priority over others whenever there is a choice of which to employ. The extra richness of the augmented grammar may be expected to elevate the incidence of temporary ambiguity and consequent garden-pathing for the parser, but this is strictly limited to those points in sentences for which the learner's currently hypothesized grammar is

inadequate and learning must occur. Elsewhere, parsing complexity remains within normal adult bounds. For example, the child is assumed to parse sentences in order to comprehend them, as adults do; and for that purpose the child's parser computes just one syntactic analysis for each sentence, as is widely assumed to be the case in adult parsing.

A serious workload explosion would result if the parser were required to compute *every* analysis of a parametrically ambiguous sentence. But this is clearly ruled out by the feasibility criterion. I assume the most the parser can be asked to do is to note when an ambiguity point arises in the course of analyzing a sentence. For example, it should flag the fact that an incoming noun might be attached into the sentence structure as either a subject or an object; or that a subject is attachable as Specifier of CP or of IP or of VP; or that a PP might be attached as daughter to VP or into an NP as a modifier of the noun. The existence of a choice point in the parse is a sign that the word string is structurally ambiguous. If the choice lies between two (or more) aspects of the learner's current grammar, it is not a parametric ambiguity. We can assume it is resolved in the usual fashion, by Minimal Attachment and/or whatever other parsing strategies are active in children; see Trueswell et al. (in press). Within-language ambiguity is excluded from further consideration here. If the choice is between the current grammar and an as yet unadopted UG treelet, it will be resolved in favor of the former, since the STL model assumes, as do all "error-driven" models, that the current grammar should not be changed as long as it continues to be compatible with the input. Or the parser's choice may be between two (or more) novel treelets, neither of which is part of the current grammar. In that case the parser may opt for one analysis rather than the other in order to assign meaning to the sentence. But the learning device – if it wants to avoid risks – must not set any parameters on the basis of the analysis the parser has chosen. In general: whenever the parser picks one route to follow and does not compute through the alternative analyses, a conservative learner will want to be notified, so that it can refrain from setting any further parameters on the basis of that sentence. No treelet utilized at or after the ambiguity point can be guaranteed correct, because the sentence may have some other structural analysis, employing other parametric treelets, that the parser does not know about.

Unlike other decoding systems that have been proposed, the STL can also reliably detect that there is no parametric ambiguity in some input sentence. Where that is so, parameter setting (adoption of a new treelet) can proceed safely. The outcome will be correct, and the learning system will know that it is correct, so that other decisions can be based on it. The new and interesting issue raised by this decoding system is what a learner should do when the parser detects that parametric ambiguity *is* (or may be) present and alerts the learning system. Should it stop learning immediately, to avoid danger of errors? That is what the earliest STL model did. It embodied the belief of many theoretical linguists that every (non-default) parameter value must have a unique trigger that is readily accessible to learners. If that is so, the wisest

strategy is just patience and precision. But if the necessary unambiguous triggers do not exist, or are not guaranteed to come by frequently enough to expedite learning, then perhaps it would be more efficient overall to be less patient and less precise. Which strategy is optimal for natural languages? And which is what children do? These questions are opened for debate in sections 5.2 and 5.3.

# 5 Consequences of Ambiguity

We can now consider the decision strategies a learner might employ to choose which grammar to shift to when its current grammar has just failed on an input sentence. The case of interest is where the input sentence does not decide the matter because it is parametrically ambiguous – or may be, for all the learner knows. And the central question is: what effect does the degree of parametric ambiguity have on the effectiveness of different decision strategies? In particular: is there a learning model that is relatively immune to the high level of ambiguity in natural language, as children appear to be?

## *5.1 Without ambiguity detection: errors*

As noted in section 4.2, learning models that meet the realistic processing load criterion are generally unable to detect parametric ambiguity, because parametric decoding other than by method (f) is such a struggle. In models other than the STL, therefore, if the decoding system can find any way to license a sentence, the learner must settle for it as if it were the only way. This is equivalent to guessing which of the possible parametric analyses of a sentence is correct. The effect of ambiguity is obvious: the greater the ambiguity, the greater the pool of candidates, so the less constrained the guess. As a result, the trajectory through the domain of possible grammars which should bring the learner's guesses closer and closer to the target is not so well directed. The feedback from parsing success is not systematically related to the learner's parametric choices. In section 4.2 we observed that a move in the right direction may fail to be positively reinforced because some other aspect of the grammar is still incorrect. Once ambiguity is added in, the opposite also occurs: the learner may be positively reinforced by parsing success when it sets a parameter to the wrong value. Hence, in an ambiguous domain, time and effort can be wasted pursuing trails that lead nowhere.

The damage done by false feedback might even be permanent. It is an open question whether an ambiguity-blind system like the TLA could be led into superset errors which are uncorrectable. This is an obvious danger in any non-deterministic model which makes errors and hopes to be able to correct them later. (I use the term *deterministic* here in the sense made familiar in parsing theory by Marcus 1980, to denote unrevisable, or "indelible," computations. In non-deterministic learning, a parameter that has been set one way could later be reset to its opposite value. See discussion by Clahsen 1990.)

Even if errors are not permanent, they can be costly. They increase the learner's total workload by requiring parameters to be reset possibly many times en route to the target grammar. This cost of making errors is beginning to be quantified (in terms of additional inputs needed before convergence on the target grammar); see Berwick and Niyogi (1996) and Sakas and Fodor (in press). Since errors are unavoidable for a non-deterministic learning procedure, the costs of error correction need to be minimized. Such a learner is therefore best paired with a highly efficient mechanism for decoding and (re)setting parameters. For this reason, the non-deterministic response to ambiguity can be evaluated most favorably if it is implemented not in the TLA framework but in combination with the treelets decoding method of the STL, which has a more constructive system for finding a candidate grammar able to parse a given sentence. A non-deterministic version of the STL is outlined in section 5.3.

For cue search systems such as Lightfoot's the consequences of ambiguity are harder to assess, because no effective method of cue recognition is specified. In a recent presentation Lightfoot (1998) adopts some aspects of the STL model, such as the identification of cues and parameter values (though, oddly, without adjusting the proposed cues to reflect the true content of the parameter values; see section 4.3 above), but he does not take advantage of the treelet parsing method for parametrically decoding the input. An input sentence receives a single parse, often incomplete, possibly incorrect; how it is assigned is unclear. "As a child understands an utterance, even partially, she has some kind of mental representation of the utterance. These are partial parses" and they are scanned for I-language cues (Lightfoot 1998: 4). This is not very informative, though it is reminiscent of other learning systems that construct parse trees by guesswork tempered by UG principles, such as Fodor (1989) and Clark (1996). In any case, it is clear from Lightfoot's description that the cue search system is not intended to be error free. In fact it is capable of errors even when correct information is available, since it does not respond to a cue until it has encountered it with some fairly substantial frequency. This is why language change occurs when the frequency of occurrence of a cue declines for any reason (e.g. loss of English verb-to-I movement by the eighteenth century, following the rise of periphrastic *do* and other changes that reduced the number of constructions in which the verb was visibly raised over another element such as the subject or negation).

Thus the cue based model makes guesses, as the TLA does. So it makes errors, at least some of which it subsequently corrects. However, the frequency sensitivity explanation for historical change suggests that, unlike the TLA, this is not a one-trial-learning device. Rather than resetting a parameter on the strength of one conflicting input sentence, this learner may be designed to collect up the weight of evidence for and against each parameter value, and to adopt a value decisively only when the evidence in favor of it exceeds some threshold. This would be similar to proposals made by Kapur (1994) and Valian (1990). To what extent such a system would make overt errors in

production and perception during its period of indecisiveness concerning each parameter would depend on its strategy. It might employ on every occasion whichever value of the parameter was temporarily ahead; or it might employ both values, with probabilities in proportion to their relative standing; and so on. From the learner's point of view, the effect of increased ambiguity would be to spin out the adjudication process between the alternative values of a parameter, and postpone the time at which any values could be eliminated from consideration.

If this is what is intended by way of ambiguity management for the cue based model, it too can be implemented in a manner not unlike the non-deterministic variant of the STL to be described in section 5.3. As was observed in section 4, the concept of I-language structural cues is highly compatible with the notion of structural triggers, or treelets, on which the STL decoding system relies. So the cue based model could select any of a range of STL-type systems as its implementation, to supply the missing machinery for creating parse trees for novel input sentences. The non-deterministic STL described in section 5.3 is probably most in keeping with Lightfoot's theory.

## 5.2 *With ambiguity detection: delay*

The treelet-based parametric decoder of the STL model can determine that an input sentence is parametrically ambiguous though it cannot reasonably compute more than one analysis for each sentence. In fact the STL overestimates parametric ambiguity, since the parser will flag an ambiguity point when it encounters mere within-language structural ambiguities, or temporary ambiguities which are resolved later in the sentence, neither of which would in fact derail parameter setting. But though it sometimes overreacts, it never misses a parametric ambiguity. (Almost never; see discussion in Fodor 1998c.)

A learning system capable of detecting ambiguity has two choices for dealing with it. The learner can be conservative and refrain from setting parameters on the basis of any part of a sentence that is within the scope of an ambiguity. Or it can take risks by guessing which of the competing analyses is the correct one. The first strategy is suitable for a deterministic learner, and the second for a non-deterministic one, in the sense defined above. A conservative, deterministic version of the STL makes only correct decisions. As I will show, it pays the price of ambiguity in the time it must wait for unambiguous inputs to learn from (if indeed they exist). A non-deterministic version of the STL goes much faster but makes some wrong decisions; it pays for ambiguity in errors and the need for error correction, as the TLA and other guessing systems do. As noted in section 4, the earliest STL models were strictly conservative (Fodor 1995, 1998a, and the "weak STL" of Sakas and Fodor in press). But that is only possible if the input is cooperative; it may be expecting too much of the quality of input that children really receive. A version of the STL that does not always wait for perfect input might be more successful and provide a better match for human learners. We need to know, and one good

way to find out is to compare the two variants to see how resilient they are to attack by parametric ambiguity.

From now on I will refer to the conservative version of the STL as the D-STL (for deterministic STL). The D-STL sets parameters (adopts new treelets) online, when needed to enable the parser to analyze an input sentence, but it ceases parameter setting as soon as the parser detects any ambiguity point in a sentence – a point of net ambiguity, resolved neither by the input nor by the grammar acquired so far. Parsing continues past this point for comprehension purposes, but the learning system discards the remainder of the sentence as a basis for parameter setting. It may thereby waste some reliable information, since, as noted, it sometimes perceives parametric ambiguity where none is present. But its discard policy is at least more rational than that of other models. The TLA, for example, discards many inputs – including unambiguous ones – due to decoding failures, but sets parameters on the basis of ambiguous and unambiguous inputs alike. The D-STL discards all ambiguous and some unambiguous inputs and learns from the unambiguous remainder.

The D-STL's discard rate is necessarily higher the greater the net parametric ambiguity of the target language sample. And when input is discarded, nothing is learned from it. Therefore, in a language domain which is highly ambiguous the D-STL sets parameters less frequently than when ambiguity is low; it consumes more input sentences for each parameter it sets. At the same time, the D-STL shows the progressive disambiguation effect noted in section 3.2: learning speed picks up over the course of learning as more and more parameter values are pinned down. Establishing a parameter value means that a target treelet has been adopted into the learner's grammar, where it becomes a source of certainty rather than uncertainty for the parser. On standard assumptions, to adopt one value of a parameter is to reject its other value (e.g. adopting complement initial for VP amounts to rejecting complement final for VP). So adopting a treelet has the effect of shrinking the collection of treelets waiting in the wings to be called on if the current grammar fails. And that increases the probability that the parse is rescuable by only one treelet, which would allow that treelet then to be adopted. Thus each parameter that is set makes it easier to set the next one.

Nevertheless, we have discovered from working on this model that for the D-STL, or any other conservative learning device, the delays between usable inputs can be very long indeed. The reason was touched on at the end of section 4.3: a conservative learner must discard not only parts of sentences it knows to be parametrically ambiguous (net ambiguous), but also parts of sentences which it has not fully tracked and so does not know are *not* parametrically ambiguous. This covers a lot of ground, since for a serial parsing device any part of a sentence to the right of an ambiguity is less than fully monitored. What this adds up to is that the D-STL discards input for purposes of setting one parameter, say P7, not merely if it is ambiguous with respect to P7 but also if it is ambiguous (net ambiguous) with respect to any other parameter(s). For instance, until it had determined whether the target language

is +V2 or −V2, it could set no parameters controlling phenomena in the VP (e.g. indirect objects), because they would be masked by the ±V2 ambiguity to their left. Absurd as this might seem, it is a consequence of the extreme cautiousness that is necessary in a deterministic system. (Note that I assume here no meta-knowledge on the part of the learner about which parameters could or could not interact with each other in derivations.) Thus, the only usable data for parameter setting by the D-STL are sentences or parts of sentences in which, to the left of any ambiguity (within-grammar or cross-grammar ambiguity), a target treelet not yet in the learner's grammar is expressed unambiguously; that is, the sentence cannot be parsed without it.

Probabilities can be assigned to the factors relevant to speed of learning (e.g. the probability that the currently hypothesized grammar will fail on an input, the probability that the failure point at which the parse crashes precedes any ambiguity in the sentence, the probability that a unique UG treelet is able to rescue the parse), and the mathematics can be worked through to give the probability of usable data at various degrees of overall parametric ambiguity. (See Sakas and Fodor in press for mathematical considerations.) From this, the average wait between parameter setting events can be calculated. It is, unfortunately, very high indeed for anything approaching realistic degrees of ambiguity: millions of sentences, in some cases, between parameter setting events. Moreover, the average wait is less than the maximum wait, which is what counts if we are to ensure that *every* learner attains the target in a reasonable time. There are simplifying assumptions entering into these computations which are most likely too stringent for real life. Also, our calculations so far have not included the ameliorating effects of progressive disambiguation, by which the average delay between usable inputs decreases rapidly as learning proceeds. So these are by no means final estimates. But these early results stand at least as a warning that waiting for unambiguous input to learn from can be a costly strategy.

To summarize: D-STL acquisition comes close to the ideal of setting parameters accurately, once and for all, in response to unambiguous triggers supplied by the environment. But accuracy and speed do not go together. Even with its efficient treelet decoding procedure, the feasibility restriction to serial parsing entails that some unambiguous parametric information in input sentences is masked by ambiguities and is not accessible to the learning routine. Hence in the presence of parametric ambiguity, D-STL learning is accurate but very slow. It is especially slow at the beginning; ambiguity hits early learning hardest.

Is this compatible with the facts of human language learning? To the extent that children make syntactic mistakes, these would have to be attributed to faulty input, lack of lexical knowledge, semantic confusions, processing slips, etc. But this is not unrealistic. It has repeatedly been noted that children make remarkably few syntactic (as opposed to lexical or morphological) errors. This speaks in favor of conservative learning and against learning algorithms which engage in random guessing. (See, however, Bowerman 1990 on some

early word order errors, e.g. *Comes feet under here*.) The slow rate of parameter setting is a legitimate concern, but is not necessarily fatal. After all, if a child has just 20 parameters to set by the age of 5, learning need go no faster than one parameter every couple of months. Even if there are 50 parameters, or 100, the time for each one is still measured in weeks, not days or minutes. This contrasts with the rate of lexical learning, which has been estimated at one word every two waking hours during the pre-school years (Pinker 1994, citing work by Nagy and Anderson).

On the other hand, the extreme effect of ambiguity on the setting of the first few parameters does seem hard to reconcile with human performance. The prediction is that – at least relative to a constant flow of information from the environment – the earliest-set parameters are set orders of magnitude more slowly than later parameters. No empirical surveys have been done to establish how many parameters children have set correctly at what ages, and whether this accelerates. Studies of particular syntactic phenomena do occasionally reveal a lag between the time at which relevant evidence appears to be available to children and the time at which they have demonstrably mastered the facts. These cases are often attributed to late maturation of some UG contribution to the construction (e.g. the maturation of A-chains, Borer and Wexler 1987; see also Wexler 1999).

Perhaps such cases should be re-examined from the perspective of conservative learning. It seems unlikely, but might these laggardly phenomena be particularly susceptible to masking by other ambiguities in the same sentence? More commonly, acquisition research reveals that children know more of the syntax of their language than they normally make use of. Constructions can be elicited which occur rarely if at all in the child's spontaneous production at that stage (e.g. a relative clause in a purpose clause at 3 years 5 months: *Jabba, please come over to point to the one that's asleep*; *wh*-extraction from a subordinate clause at 3 years 11 months: *Squeaky, what do you think that is?*; see Crain et al. 1987, Crain and Thornton 1991). And comprehension experiments with babies not yet producing any word combinations confirm that they already know some basic facts of their target language such as the surface order of subject, object, and verb (Hirsh-Pasek and Golinkoff 1991). Tentatively, then, I conclude that children exhibit no great delay in getting the early parameters set. (See also the Very Early Parameter Setting generalization of Wexler 1998.)

There is an interesting possible explanation for this which might save the D-STL. Perhaps natural languages are particularly kind to the conservative strategy for parameter setting, allowing it to be successful on sentences typical of children's early input even though for other sentences it would indeed be slow. This could reconcile the theoretical predictions of conservativism with the achievements of human learners. We know what linguistic properties would be of assistance. The beginnings of sentences are the most important: an unresolved ambiguity late in a sentence will mask less than if it were to occur at the beginning. Also, for equivalent overall amounts of ambiguity, it would be better for a sentence to have more ambiguity points with fewer treelet

competitors at each, rather than fewer ambiguities with more competitors, since then the setting of one or two parameters has a good chance of eliminating an ambiguity entirely, thereby opening up later parts of sentences for further learning. Suppose these helpful properties were characteristic of relatively simple sentences such as infants comprehend and on which they presumably rely for learning, sentences such as *Where's the kitty?* or *Mommy will read you a story*. This could break the early learning bottleneck despite massive parametric ambiguity in the language as a whole. To the contrary, however, a look at natural languages makes it all too clear that early learners are faced with multiple ambiguities starting from the very first word of a sentence (see section 6). The natural language domain does not help out the D-STL; it makes deterministic early learning as difficult as it possibly could be.

## *5.3 With ambiguity detection: no waiting*

The STL has a choice of strategies for responding to input ambiguity: it can exploit its ambiguity detection ability or not do so. If it does not, it is in the same boat as the TLA and other grammar guessing systems that are unable to detect ambiguity because they try out only one grammar at a time. Disregarding ambiguity is the approach of the non-deterministic variant of the STL, also known, for reasons that will be clear, as the *Parse Naturally* STL (PN-STL; Fodor 1998c).

Its parser is still a serial parser, and it does exactly what a normal (adult) human parser would normally do: it computes its favorite analysis of an input word string, based on Minimal Attachment and other innate preference principles. (Minimal Attachment says that an input word should be attached into the parse tree using the fewest possible new nodes; Fodor 1998b defends the innateness of both the parsing mechanism and its preferences.) Unlike the deterministic STL, it does not record ambiguity points. The only difference from adult sentence processing is that at a point of parse failure, the learner has access to the innate lexicon of parametric treelets. These are temporarily folded into the grammar acquired so far, as described above, so the parser's use of them is governed by Minimal Attachment and so forth, just as for other elements of the grammar. Having computed its preferred analysis, the parser reports it to the learning component without comment on ambiguity. The learner treats the parse tree as if it were correct, and adopts any new parametric treelets it contains. If the analysis *is* correct, all is well. But if a sentence is globally parametrically ambiguous, the parser's preferred analysis may differ from the target analysis, so the parameter values adopted may be incorrect. The higher the net ambiguity rate the greater the chance that this is so.

In short: the PN-STL lets the parsing strategies make decisions about how to resolve parametric ambiguities, and since these decisions cannot always be right, the PN-STL makes mistakes in setting parameters. For instance, the Minimal Chain Principle (De Vincenzi 1991) is another important parsing principle, which favors non-movement analyses over movement analyses.

This parsing preference will cause the learner to adopt –V2 rather than +V2 where there is a choice. (Gibson and Wexler 1994 also propose that learners prefer non-movement analyses, though not for parsing reasons.) This is correct for SVO strings in English but not in German. For German the V2 parameter will need resetting when a sentence is encountered for which no –V2 analysis is possible. The acquisition of +V2 in German is still not well understood, but there are some reasons for believing that the verb does not move to the C projection until Case morphology is acquired and the evidence of Accusative-initial sentences outweighs the avoidance of movement (Weissenborn 1990).

It seemed self-evident in the early days of working with the STL that its ability to more or less effortlessly detect and avoid ambiguity was a great asset not to be wasted. The PN-STL does waste this gift, but it has some good features to recommend it nonetheless. The mistakes it makes are not random or stupid ones. Its disambiguation choices are systematic, so they help to explain the uniformity of language acquisition by all normal children. They reflect the human parser's natural tendencies, which may increase the chance that the selected analysis is the one the speaker intended. The human parser is apparently a least-effort device, preferring to build structures that are as simple as possible, so even if the parser's analysis is wrong, at least the processing load is not excessive. Error correction is needed but is relatively fast given that the PN-STL does not suffer from decoding delays. Also, the PN-STL is well able to benefit from useful-but-not-quite-reliable hints about sentence structure from prosody and semantics just as adult parsing does (within bounds of modularity). These non-syntactic sources of information cannot conveniently be used by an ABS system like the TLA, which first picks a new grammar to try and only then inspects the properties of the input sentence. Nor can they be exploited by a deterministic system that cannot afford to take chances on partial cues. And a system that uses prosody to set some syntactic parameters directly could employ only a small proportion of the ubiquitous structure-sensitive prosodic patterns in natural language sentences. To extract the most benefit from prosodic and semantic cues, they should be used by a non-deterministic learner not insisting on total accuracy, where they can affect parameter setting indirectly by contributing to selection of the most likely tree structure for novel sentence constructions. Above all, the Parse Naturally system brings relief for early learners. Parametric questions are answered (albeit tentatively) as soon as they arise, so learners rapidly gain a substantial working grammar which can be used for comprehension and production until such time as the target parameter settings are stabilized.

These merits must be weighed against two disadvantages that beset all non-deterministic learning. When errors occur they can create misleading contexts for setting other parameters, thus generating even more errors (Clark 1988). This tendency is presumably exacerbated by high levels of parametric ambiguity. Also, an error-prone non-deterministic learner can never afford to

dismiss the parameter values that are the competing partners of the ones it has adopted, because it may need to revert to them later. Eliminating disconfirmed values was the source of the progressive speed-up in learning rate over time discussed in section 3.2. It appears that the PN-STL sacrifices this acceleration in return for the speed it gains at the start. The effect is to flatten out the learning curve across the whole course of learning, evening up the workload – perhaps not a bad trade-off. And arguably, the PN-STL does not suffer too greatly in efficiency by its retention of unpromising-looking parameter values just in case they are needed later. The PN-STL is relatively unaffected by the number of grammars in the pool of candidates, because it considers only the most highly ranked one. Unlike other models, it selects among just those that can parse the current sentence, and attends only to the parser's preferred candidate at each choice point. Furthermore, the PN-STL can be given the capability of keeping a running tab on the success rates of all the UG treelets depending on how often they come to the rescue of a blocked parse, as suggested in section 5.1 for the cue based learner. By this means, even though it never really eliminates any treelets, the PN-STL would gain much the same advantage as if it definitively adopted some and dismissed others. Its strategy would be to give priority to the most successful treelets. The more these are used the stronger they will get, so they will streak further ahead of the others and will be, in effect, the only ones in play – except if input is subsequently encountered which offers no choice but to boost a previously low-ranked treelet. Thus, rapid narrowing of the field of likely candidates may be compatible with revision capability in case of unexpected turns in the data.

An empirical prediction of this model is that the phenomena in each language that are the hardest to learn will include those for which there exists an incorrect parametric alternative which the parsing system strongly prefers. This is not the same as claiming that all structurally complex constructions are challenging for learners. Difficulty is predicted just where the evidential support from the learner's input for the correct treelet would have to fight against the parser's disinclination on-line to assign the correct structural analysis rather than some other. Studies of adult sentence processing show that the human parsing mechanism does sometimes fail to compute a correct analysis that is highly dispreferred. To the extent that parsing preferences are innate and already active in children, we should observe slow spots in learning correlated with the known dislikes of the human parser, as in the case of the Minimal Chain preference mentioned above. More work is needed to generate exact predictions for particular languages. There is also much work to be done in establishing the error curves for PN-STL learning under various conditions of parametric ambiguity, expression rates, and so forth, to see how accuracy varies over the course of learning. For modeling the complex dynamics of error correction systems, mathematical methods are less practical, so computer simulation will be needed to evaluate the PN-STL and compare it with the performance of the deterministic STL.

## *5.4 Assessment*

There is a family of possible STL learners, all using the innate lexicon of treelets to decode the parametric information in input sentences. Here I have utilized the general STL format to compare two different varieties incorporating some design choices proposed in other models. The purpose of the comparison is to gain insight into how the human language learning mechanism is designed, by assessing the strengths and weakness of the models, and comparing their performance with that of human learners. The D-STL emphasizes the accuracy aimed for in early switch-setting models. The PN-STL takes chances and relearns where necessary, like the TLA. They have in common their efficient decoding procedure, and the STL emphasis on sentence structure as the mediator between input word strings and grammars; parameters are concerned with aspects of that structure. The theoretical linguistic concept of a structural cue or trigger for each parameter value is preserved and is integrated into a psycholinguistic account of sentence parsing which spans adults and children, and which takes on much of the burden of the learning process.

There are more STL varieties imaginable than these two, which contrast maximally in their handling of ambiguity. A possible intermediate system would flag ambiguity points like the D-STL, but in case of ambiguity would set parameters anyway like the PN-STL. This would give the initial fast progress of the Parse Naturally approach, but the learner's confidence in a treelet could be scaled to whether it was adopted on the basis of ambiguous or unambiguous evidence. Another design that might recommend itself is the D-STL equipped with default parameter values for the child to use in language production and comprehension while waiting for decisive evidence of target values. Defaults, however, must be employed with great caution in a deterministic system, since they can engender errors (even the otherwise beneficial Subset Principle default; see Fodor 1998c). Perhaps other STL variants will emerge that are superior to these.

Formal evaluation of these models has barely begun. We know too little still about their performance characteristics for there to be a final judgment yet. The discussion so far has suggested the following rough and ready evaluation. The deterministic approach which became a practical possibility with the advent of STL decoding is the only way to achieve fully accurate parameter setting. Computationally this accuracy is essentially cost-free, but in terms of learning rate it is not, especially at early stages. The PN-STL comes closer to delivering constant rate parameter setting across the timespan of learning, despite the enormous range of uncertainty levels that children face at different stages. But the parameter setting errors of the PN-STL seem not to do justice to real children, who tend with few exceptions not to use syntactic constructions they do not know how to use correctly. Also, at present it is an open question whether the errors of the PN-STL, like those of Gibson and Wexler's TLA, may lead it into territory from which it may never retreat.

How can we advance on these approximate assessments? Computational linguistic research can continue to spell out the efficiency characteristics and convergence rates of each approach. It falls to psycholinguistics to determine which accords best with empirical data on actual parameter setting progress by children. More extensive experimental data on children's sentence processing may also be informative. One other possible source of information is the linguistic facts themselves. By comparing the properties of natural languages with how they might have been in a make-believe world better designed for learners, we may get a sharper estimate of how much of a challenge acquisition really is, and hence how robust the human learning mechanism must be. I try out this line of thought in section 6.

# 6 Patterns of Parametric Ambiguity in Natural Language

Is it possible to hold on to the ideal picture of rapid error free parameter setting for natural language? So far we have seen that total accuracy is probably incompatible with speed. Moreover, it appears that natural languages are designed to magnify this incompatibility. The distribution of parametric ambiguity in natural language is far more damaging for learners than it need be. It is possible, though still unclear at present, that developments in syntactic theory might ameliorate this situation. If not, deterministic learning is probably not practicable; some guesswork must be resorted to to get the job done.

## *6.1 String-to-structure alignment*

It would have been more convenient for children if natural language parameters were all concerned with surface facts, and if every parameter value expressed by a sentence were expressed independently of the others and unambiguously. (Of course, it would be more convenient if there were no parameters at all. See Pinker and Bloom 1990 and commentaries for speculation on why human evolution did not go further and provide us with a fully formed innately specified language.) Instead, the way of natural language is to let P&P values and lexical items intermingle in a derivation so that at the surface there is no separable piece of the word string attributable to each piece of the grammar involved in the derivation. The relation between word strings and their parametric generators is thus opaque at best. And because different derivations may converge on the same word string it is also often ambiguous. A large part of the problem is that sentences have abstract structures far richer than the lexical items that realize them audibly: the non-terminal-to-terminal node ratio is high. This is particularly so for syntactic analyses since Pollock (1989), in which what were once represented as features of lexical projections now appear as functional heads with projections of their own. As Bertolo et al. (1997) have pointed out, this can create ambiguities concerning the position of

a verb among the stack of inflectional heads representing tense, agreement, aspect, and so forth. A′-movement and its traces also contribute to the disproportion of inaudible to audible elements.

Neither non-terminal nodes nor empty categories would be a problem for learners if everything about them were innate. In fact, although their existence and distribution are regulated in part by innate principles, they are also to some extent parameterized. Even if the array of functional heads in the extended verbal projection is universal and totally predictable for learners with access to UG (e.g. Cinque 1999), it is still necessary for a learner to discover how that structure aligns with the words of target language sentences. For instance, in *The mouse squeaks* is *squeaks* in V or $AGR_o$ or T or $AGR_s$ or C? Is *the mouse* in the Specifier of V or of T or $AGR_s$ or C? Bertolo et al. contemplate a language such that no surface facts fully determine the answers to these questions. In that case a seriously conservative learner such as the D-STL would wait forever for unambiguous input to set the verb movement parameters. The learner could never parse verbs at all for lack of knowing where they should be parked in relation to the innately prescribed non-terminal nodes in the parse tree. Even in a language where the answers are determinate, it may take quite a lot of evidence from input sentences to establish them; and a conservative learner can build no parse trees until that evidence has been encountered. Note that this is a situation in which neither prosodic nor semantic cues are of any assistance. Verb movement is a formal operation which has no effect on phonological phrasing or meaning.

The Minimalist Program suggests that morphological cues might be useful, since overt movement is driven by strong features on functional heads and there is some tendency for strong features to be overtly realized morphologically. For example, rich inflection is often cited as a predisposing factor for verb movement to I. If the correlation between overt morphological realization and strong syntactic features were exact, learners would be able to read off from the verb's morphology (once acquired) not only that the verb must have moved, but exactly which functional head it moved to within the extended verbal projection. Unfortunately it appears that this relationship is not reliable enough for learners to trust. The Minimalism based learning system of Wu (1994) treats morphological strength and syntactic strength as independent parameters (though see Pollock 1997 for a new way of relating them). The only evidence that learners can rely on, it seems, is the positional markers that linguists rely on in motivating analyses with split Infl: the adverbs which may intervene among the verbal head positions, an overtly realized negative head, and so forth. For example, Pollock (1989) used the contrast between *John often kisses Mary*/**John kisses often Mary* and **Jean souvent embrasse Marie*/*Jean embrasse souvent Marie* to argue that the finite verb moves to a higher functional head in French than in English. Adverbs and negation are audible items, and they can fix the locations of movable entities such as the verb and its arguments, within the (inaudible) extended verbal projection.

The problem with these indicators of string-to-structure alignment is that adverbs and negation are optional in sentences. So learners will receive only occasional doses of positional information. Many of the sentences children hear are not structurally disambiguated by such elements. Lack of positional markers is especially true of very simple sentences like *Mickey squeaks*, which presumably constitute the intake (the processible input) of beginning learners, who are most in need of disambiguation assistance. To make things worse, the unresolved ambiguities of verb and argument position that young children are exposed to occur often at the very beginnings of sentences (e.g. the structural position of an initial noun is multiply indeterminate). This, we have noted, is the worst possible location for an ambiguity because it will block the acquisition of any other parametric facts the sentence may contain (section 5.2). Furthermore, these ambiguities are systematic; they are not a matter of accidental overlap of word strings, which a child might be unlucky enough to encounter once but which would not recur. It is not just *Mickey squeaks* which is structurally indeterminate for learners, but *all* sentences that contain just a verb with some complements.

Thus, natural language design is extremely cruel to children: (i) natural languages have multiple positions capable of hosting the same lexical category (e.g. verb); (ii) children are not free to choose which position it should be in since there is a right and wrong answer for each language; yet (iii) UG does little to ensure that the target position is recognizable in basic sentences in which the item appears. This is especially punishing for a conservative learning strategy which demands certainty before it takes action, and for which one ambiguity can block the learning of other facts. As the extreme case of the paralysis noted by Bertolo et al. (1997), a truly conservative learner might have no grammar at all for verb placement until the verb has been observed in relation to every one of the positional landmarks that UG provides. Cinque (1999) lists 32 classes of adverb that would need to be observed, as well as multiple positions for negation.

In summary: natural languages abound in ambiguities of the worst kind for a deterministic learner: systematic ambiguities which occur early in sentences and early in a learner's career, and which are highly frequent but resolved infrequently. An artificial domain of languages in which simple sentences facilitate parameter setting can easily be created. But natural language design seems to do all it can to exacerbate the early learning problem. These observations appear to put the precision-loving deterministic STL at a disadvantage relative to the happy-go-lucky PN-STL. The former backs away from the ambiguities the primary linguistic data throw at it. The latter muddles on through until some fixed points of information finally begin to arrive; and if some never do, it has a grammar anyway.

However, the distribution of parametric ambiguity in natural languages depends not only on the language facts but also on their proper theoretical interpretation. Before we abandon forever the goal of high-precision triggering

of syntactic parameters, it is appropriate to consider what difference it would make to learning if the language facts were differently analyzed.

## *6.2 The problem of short sentences*

The problems circle around the properties of short sentences. Young children produce short sentences, and they show signs of comprehending short sentences better than long ones, by and large, so we assume that they learn from short sentences. Chomsky (1988: 70) wrote: "Notice that the value of the [headedness] parameter is easily learned from short simple sentences. To set the value of the parameter for Spanish, for example, it suffices to observe three-word sentences such as (3)," where (3) is *Juan habla inglés* "Juan speaks English." But this is wishful thinking. An SVO sentence does not suffice to establish head–complement order. Consider *Johann spricht Englisch*.

Though they may be easier for production and comprehension, short sentences are not necessarily simpler for acquisition than long ones are. All depends on what they leave out. A short sentence simplifies early learning by not presenting embedded questions or adverbial clauses or long distance extraction. The parameters peculiar to those constructions do not have to be set yet, and they do not create ambiguities that get in the way of setting other parameters. But a short sentence complicates learning if it leaves out the items that resolve parametric ambiguities or show the scaffolding into which the overt items fit. Ideally, the earliest sentences children attend to would be composed of items which are not themselves parameterized, and which will help to clarify the parameter settings needed for other items to come. Here, as we have seen, natural languages win no design prizes. (I am indebted to Anne Christophe and Norbert Hornstein, personal communication, for insisting on this point.)

The shortness of sentences affects different parameters to different degrees. For obvious reasons it is easier for a short sentence to reveal the positive value of the null subject parameter than the positive value of the V2 parameter. Though the particularities of different parameters cannot really be set aside, some general effects of sentence length can still be discerned. The shortness of a sentence imposes an inherent limit on how informative it can be for learners. Parametric information is carried by the overt items in a sentence: the categories of the words and the discernible relationships among them such as precedence or agreement (and also, sometimes, by what is missing). If a sentence needs more parameter values for its derivation than can be signaled by the words it contains, parametric ambiguity results. So short sentences are liable to overflow their parametric banks, so to speak, unless they are derived using only a small proportion of the full set of parameter values that generates the whole language. But this condition is hard to satisfy if a sentence consisting of just one verb has a structure with the full array of inflectional heads, all needing to be specified for weak or strong features controlling movement.

Conclusion: It could be that the best way to facilitate early learning would be to ensure that UG permits short sentences to have simple, relatively

parameter-free derivations. Ambiguity overload can be kept at bay if parametric questions do not pile up unanswered, waiting until more "advanced" input is heard and absorbed.

## *6.3 How much help from UG?*

We have observed that even if large chunks of language structure are innately programmed into human brains, a learner may be quite unable to tell how that structure should align with the words heard. This is parallel to the situation for syntactic categories: the categories Noun and Verb are surely innate but children still must learn which of the words they hear belong to which categories, and that is not a trivial task. The idea that innately prescribed structure is cost-free for language learners is common in linguistic research and it seems eminently plausible. It is also welcome, because it means there is no reason not to assume the innateness of many aspects of language structure. A structural configuration needed for one language can be assumed to occur, inaudibly, in other languages too, as long as there is no specific evidence to the contrary. This brings linguistic theory closer to being able to claim that all natural languages have essentially the same derivational structures – except only that not all the same parts of the universal structure are spelled out in every language.

Unfortunately, the argument that what is innate is *ipso facto* effortless for learners is not valid. It is clear now that even if the structural scaffolding of sentences is everywhere fixed and the same, any particular sentence may be highly ambiguous with respect to how its words are attached to that scaffolding. For UG to be truly helpful, it should supply innate sentence structures *and* fix their relation to surface word strings – or at least constrain that relation tightly enough that learners can rapidly fill in the rest. As long as there is substantial crosslinguistic variation with respect to how innately defined structure is overtly lexicalized, there will be ambiguities of string-to-structure alignment that may be very onerous for learners to resolve. Every bit of universal structure, even if it is not "used" in some language, can make that language harder to learn.

But UG can assist learners in two other ways. First and most obviously: whatever the facts are that adults know, a learner has a lighter time of it the more of them it knows innately. If simple sentences have rich structure, better that it should be innate than not. More interestingly: UG principles control how much invisible structure learners have to assign to word strings. Thereby they control the balance between parametric ambiguity, which is troublesome, and parametric irrelevance, which can be helpful as a way of postponing difficult questions until the input is rich enough to provide the answers. The question, then, is whether UG could permit simple sentences to have parametrically simple and unambiguous derivations. There is more than one way this could be achieved. Which if any is correct is a linguistic issue. I sketch two broad approaches here.

i Adopt a theory of UG which matches the structural ambiguity of overt elements to their time of occurrence in learners' intake: the earliest items should be the most determinate. This would reduce the string-to-structure alignment problem by improving the balance in early language samples between the elements that need to be structurally located and the elements that can help to locate them. In earlier forms of transformational grammar before the split-Infl hypothesis, and in other linguistic frameworks such as LFG or HPSG, the verb is the fixed element in a clause, and adjuncts are positioned relative to it rather than vice versa. For target grammars of this kind, beginning learners could build correct trees for subjects and verbs right away, based on *Mickey squeaks* and other simple sentences in their conversational milieu. There would be minimal structural ambiguity to contend with at the outset, and optional elements could be added in later on.

ii Retaining the split-Infl concept, adopt a theory which entails that each sentence has only as much structure (in Infl and elsewhere) as is needed to derive its own properties (e.g. surface word order). (See, for example, Giorgi and Pianesi 1997. For an important form of argument against this idea, see Cinque 1999: ch. 6.) There would be no parametric ambiguity due to ambiguities of hidden structure (though "genuine" parametric ambiguities would still occur, such as between exceptional case marking and structural case assignment in infinitives, in the example noted by Clark 1988). Thus, a child's simplest structural hypothesis about an input sentence could be correct for that sentence even if additional parameters must be set to derive more complex sentences later.

   Note that this goes beyond the suggestion that each language employs (for all its sentences) only as many functional projections as are needed to acount for the phenomena of that language (e.g. Fukui 1986, Bobaljik and Thráinsson 1998; see also discussion in Haegeman 1997 and other references there). It also differs from the proposal of Radford (1990) that functional categories are omitted from the sentence structures computed by early learners because the ability to represent them does not mature until about 2 years. It might be combined with the assumption that functional heads can be featurally underspecified, so that a learner could acknowledge that some projection must be present to provide a landing site for movement, without yet knowing what the content of the head is. This would seem promising from a learning point of view, since it allows that a child could always build just the minimal warranted structure, not committing to any more of its details than are certain.

Other proposals in a similar spirit involve defaults. Assuming that the maximal structure is present in every sentence derivation, there might be movement defaults which keep the verb and its arguments low in the structure until specific evidence of overt movement to a higher position is encountered: features controlling movement would be weak until proven strong. Or,

assuming that languages differ with respect to the inventory of functional projections they employ, it would be natural for defaults to exclude any particular functional head until the input proves it necessary for reasons of morphology or movement. These alternatives may be more palatable theoretically than varying the richness of structure sentence by sentence, but they will demand some departure from full determinism of the learning procedure, since default parameter values are technically "errors" in languages which have the non-default value, so other learning decisions made on the basis of them cannot be guaranteed correct.

Whichever approach turns out to be right, it is clear that the assessment of current learning models is very much in the hands of theoretical linguistics. Linguists proposing syntactic parameters have often specified input triggers which could set those parameters. This is important. Unfortunately, as learning theory has begun to model the time course of parameter setting, we find that it has become more difficult to propose a realistic collection of triggers that will allow all parameters to be set accurately in a reasonable amount of time. For learnability theory there is therefore a great deal hanging on the outcome of current linguistic research on the richness of sentence structures, and particularly on recent reconsiderations of the linguistic evidence for and against the hypothesis that all sentences in all languages have identical hierarchies of functional projections providing potential landing sites for parameterized movement.

## NOTE

* Parts of this chapter appeared in the essay "Twenty questions" written for the celebration of Noam Chomsky's 70th birthday, available at the MIT Press website (http://mitpress.mit.edu/celebration). I am grateful to Mark Baltin and Chris Collins for their ideas and advice on how to adapt it to its role in the *Handbook*. Much of its content stems from discussions with several colleagues, particularly Stefano Bertolo, Erich Groat, William Sakas, and Virginia Teller.