

# A Model of Early Syntactic Development

Pat Langley

The Robotics Institute

Carnegie-Mellon University

Pittsburgh, Pennsylvania 15213 USA

## ABSTRACT

AMBER is a model of first language acquisition that improves its performance through a process of error recovery. The model is implemented as an adaptive production system that introduces new condition-action rules on the basis of experience. AMBER starts with the ability to say only one word at a time, but adds rules for ordering goals and producing grammatical morphemes, based on comparisons between predicted and observed sentences. The morpheme rules may be overly general and lead to errors of commission; such errors evoke a discrimination process, producing more conservative rules with additional conditions. The system's performance improves gradually, since rules must be relearned many times before they are used. AMBER's learning mechanisms account for some of the major developments observed in children's early speech.

## 1. Introduction

In this paper, I present a model that attempts to explain the regularities in children's early syntactic development. The model is called AMBER, an acronym for Acquisition Model Based on Error Recovery. As its name implies, AMBER learns language by comparing its own utterances to those of adults and attempting to correct any errors. The model is implemented as an adaptive production system - a formalism well-suited to modeling the incremental nature of human learning. AMBER focuses on issues such as the omission of content words, the occurrence of telegraphic speech, and the order in which function words are mastered. Before considering AMBER in detail, I will first review some major features of child language, and discuss some earlier models of these phenomena.

Children do not learn language in an all-or-none fashion. They begin their linguistic careers uttering one word at a time, and slowly evolve through a number of stages, each containing more adult-like speech than the one before. Around the age of one year, the child begins to produce words in isolation, and continues this strategy for some months. At approximately 18 months, the child begins to combine words into meaningful sequences. In order-based languages such as English, the child usually follows the adult order. Initially only pairs of words are produced, but these are followed by three-word and later by four-word utterances. The simple sentences occurring in this stage consist almost entirely of content words, while grammatical morphemes such as tense endings and prepositions are largely absent.

During the period from about 24 to 40 months, the child masters the grammatical morphemes which were absent during the previous stage. These "function words" are learned gradually; the time between the initial production of a morpheme and its mastery may be as long as 16 months. Brown (1973) has examined the order in which 14 English morphemes are

acquired, finding the *order* of acquisition to be remarkably consistent across children. In addition, those morphemes with simpler meanings and involved in fewer transformations are learned earlier than more complex ones. These findings place some strong constraints on the learning mechanisms one postulates for morpheme acquisition.

Now that we have reviewed some of the major aspects of child language, let us consider the earlier attempts at modeling these phenomena. Computer programs that learn language can be usefully divided into two groups: those which take advantage of semantic feedback, and those which do not. In general, the early work concerned itself with learning grammars in the absence of information about the meaning of sentences. Examples of this approach can be found in Solomonoff (1959), Feldman (1969) and Horning (1969). Since children almost certainly have semantic information available to them, I will not focus on their research here. However, much of the early work is interesting in its own right, and some excellent systems along these lines have recently been produced by Berwick (1980) and Wolff (1980).

In the late 1960's, some researchers began to incorporate semantic information into their language learning systems. The majority of the resulting programs showed little concern with the observed phenomena, including Siklossy's ZBIE (1972), Klein's AUTOLING (1973), Hedrick's production system model (1976), Anderson's LAS (1977), and Sembagamoorthy's PLAS (1979). These systems failed as models of human language acquisition in two major areas. First, they learned language in an all-or-none manner, and much too rapidly to provide useful models of child language. Second, these systems employed conservative learning strategies in the hope of avoiding errors. In contrast, children themselves make many errors in their early constructions, but eventually recover from them.

However, a few researchers have attempted to construct plausible models of the child's learning process. For example, Kelley (1967) has described an "hypothesis testing" model that learned successively more complex phrase structure grammars for parsing simple sentences. As new syntactic classes became available, the program rejected its current grammar in favor of a more accurate one. Thus, the model moved from a stage in which individual words were viewed as "things" to the more sophisticated view that "subjects" precede "actions". One drawback of the model was that it could not learn new categories on its own initiative; instead, the author was forced to introduce them manually.

Reeker (1976) has described PST, another theory of early syntactic development. This model assumed that children have limited short term memories, so that they store only portions of an adult sample sentence. The model compared this reduced sentence to an internally generated utterance, and differences

between the two were noted. Six types of differences were recognized (missing prefixes, missing suffixes, missing infixes, substitutions, extra words, and transpositions), and each led to an associated alteration of the grammar. PST accounted for children's omission of content words and the gradual increase in utterance length. The limited memory hypothesis also explained the telegraphic nature of early speech, though Reeker did not address the issue of function word acquisition. Overgeneralizations did occur in PST, but the model could revise its grammar upon their discovery, so as to avoid similar errors in the future. PST also helped account for the incremental nature of language acquisition, since differences were addressed one at a time and the grammar changed only slowly.

Selfridge (1981) has described CHILD, another program that attempted to explain some of the basic phenomena of first language acquisition. This system began by learning the meanings of words in terms of a conceptual dependency representation. Word meanings were initially overly specific, but were generalized as more examples were encountered. As more words were learned and their definitions became less restrictive, the length of CHILD's utterances increased. CHILD differed from other models of language learning by incorporating a non-linguistic component. This enabled the system to correctly respond to adult sentences such as *Put the ball in the box*, and led to the appearance that the system understood language before it could produce it. Of course, this strategy sometimes led to errors in comprehension. Coupled with the disapproval of a tutor, such errors were one of the major spurs to the learning of word orders. Syntactic knowledge was stored with the meanings of words, so that the acquisition of syntax necessarily occurred after the acquisition of individual words.

Although these systems fare much better as psychological models than other language learning programs, they have some important limitations. We have seen that Kelley's system required syntactic classes to be introduced by hand, making his explanation less than satisfactory. Selfridge's CHILD was much more robust than Kelley's program, and was unique in modeling children's use of nonlinguistic cues for understanding. However, CHILD's explanation for the omission of content words – that those words are not yet known – was implausible, since children often omit words that they have used in previous utterances. Reeker's PST explained this phenomenon through a limited memory hypothesis, which is consistent with our knowledge of children's memory skills. Still, PST included no model of the process through which memory improved; in order to simulate the acquisition of longer constructions, Reeker would have had to increase the system's memory size by hand. Both CHILD and PST learned relatively slowly, and made mistakes of the general type observed with children. Both systems addressed the issue of error recovery, starting off as abominable language users, but getting progressively better with time. This is a promising approach that I attempt to develop it in its extreme form in the following pages.

## 2. An Overview of AMBER

Although Reeker's PST and Selfridge's CHILD address the transition from one-word to multi-word utterances, we have seen that problems exist with both accounts. Neither of these programs focus on the acquisition of function words, their explanations of content word omissions leave something to be desired, and though they learn more slowly than other systems,

they still learn more rapidly than children. In response to these limitations, the goals of the current research are:

- Account for the omission of content words, and the eventual recovery from such omissions.
- Account for the omission of function words, and the order in which these morphemes are mastered.
- Account for the gradual nature of both these linguistic developments.

In this section I provide an overview of AMBER, a model that provides one set of answers to these questions. Since more is known about children's utterances than their ability to understand the utterances of others, AMBER models the learning of generation strategies, rather than strategies for understanding language.

Selfridge's and Reeker's models differ from other language learning systems in their concern with the problem of recovering from errors. The current research extends this idea even further, since all of AMBER's learning strategies operate through a process of error recovery.<sup>1</sup> The model is presented with three pieces of information: a legal sentence, an event to be described, and a main goal or topic of the sentence. An event is represented as a semantic network, using relations like agent, action, object, size, color, and type. The specification of one of the nodes as the main topic allows the system to restate the network as a tree structure, and it is from this tree that AMBER generates a sentence. If this sentence is identical to the sample sentence, no learning is required. If a disagreement between the two sentences is found, AMBER modifies its set of rules in an attempt to avoid similar errors in the future, and the system moves on to the next example.

AMBER's performance system is stated as a set of condition-action rules or productions that operate upon the goal tree to produce utterances. Although the model starts with the *potential* for producing (unordered) telegraphic sentences, it can initially generate only one word at a time. To see why this occurs, we must consider the three productions that make up AMBER's initial performance system. The first rule (*the start rule*) is responsible for establishing subgoals; it may be paraphrased<sup>2</sup> as:

START  
If you want to describe node1,  
and node2 is in relation to node1,  
then describe node2.

Matching first against the main goal node, this rule selects one of the nodes below it in the tree and creates a subgoal to describe that node. This rule continues to establish lower level goals until

<sup>1</sup>In spirit, AMBER is very similar to Reeker's model, though they differ in many details. Historically, PST had no impact on the development of AMBER. The initial plans for AMBER arose from discussions with John R. Anderson in the fall of 1979, while I did not become aware of Reeker's work until the fall of 1980.

<sup>2</sup>For the sake of clarity, I will be presenting only English paraphrases of the actual PRISM productions. All variables are italicized; these may match against any symbol, but all occurrences of a variable must match to the same element.

a terminal node is reached. At this point, a second production (the *speak* rule) is matched; this rule may be stated:

**SPEAK**

If you want to describe a concept,  
and word is the word for concept,  
then say word and note that concept  
has been described.

This production retrieves the word for the concept AMBER wants to describe, actually says this word, and marks the terminal goal as satisfied. Once this has been done, the third and final performance production becomes true. This rule matches whenever a subgoal has been satisfied, and attempts to mark the supergoal as satisfied; it may be paraphrased as:

**STOP**

If you want to describe node1,  
and node2 is in relation to node1,  
and node2 has already been described,  
then note that node1 has been described.

Since the *stop* rule is stronger<sup>3</sup> than the *start* rule (which would like to create another subgoal), it moves back up the tree, marking each of the active goals as satisfied (including the main goal). As a result, AMBER believes it has successfully described an event after it has uttered only a single word. Thus, although the model starts with the potential for producing multi-word utterances, it must learn additional rules (and make them stronger than the *stop* rule) before it can generate multiple content words in the correct order.

In general, AMBER learns by comparing adult sentences to the sentences it would produce in the same situations. These predictions reveal two types of mistakes – errors of *omission* and errors of *commission*. These errors are detected by additional *learning* productions that are responsible for creating new performance rules. Thus, AMBER is an example of what Waterman (1975) has called an *adaptive production system*, which modifies its own behavior by inserting new condition-action rules. Below I discuss AMBER's response to errors of omission, since these are the first to occur and thus lead to the system's first steps beyond the one-word stage. I consider the omission of content words first, and then the omission of grammatical morphemes. Finally, I discuss the importance of errors of commission in discovering conditions on the production of morphemes.

### 3. Learning Preferences and Orders

AMBER's initial self-modifications result from the failure to predict content words. Given its initial ability to say one word at a time, the system can make two types of content word omissions – it can fail to predict a word *before* a correctly predicted one, or it can omit a word *after* a correctly predicted one. Rather different rules are created in each case. For example, imagine that Daddy is bouncing a ball, and suppose that AMBER predicted only the word "ball", while hearing the sentence "Daddy is bounce ing the ball". In this case, one of the system's learning rules would note the omitted content word

<sup>3</sup>The notion of strength plays an important role in AMBER's explanation of language learning. When a new rule is created, it is given a low initial strength, but this is increased whenever that rule is relearned. And since stronger productions are preferred to their weaker competitors, rules that have been learned many times determine behavior.

"Daddy" before the content word "ball", and an agent production would be created:

**AGENT**

If you want to describe event1,  
and agent1 is the agent of event1,  
then describe agent1.

Although I do not have the space to describe the responsible learning rule in detail, I can say that it matches against situations in which one content word is omitted before another, and that it always constructs new productions with the same form as the *agent* rule described above. In this case, it would also create a similar rule for describing actions, based on the omitted "bounce". Note that these new productions do not give AMBER the ability to say more than one word at a time. They merely increase the likelihood that the program will describe the agent or action of an event instead of the object.

However, as AMBER begins to prefer agents to actions and actions to objects, the probability of the second type of error (omitting a word after a correctly predicted one) increases. For example, suppose that Daddy is again bouncing a ball, and the system says "Daddy" while it hears "Daddy is bounce ing the ball". In this case, a slightly different production is created that is responsible for ordering the creation of goals. Since the agent relation was described but the object was omitted, an agent-object rule is constructed:

**AGENT-OBJECT**

If you want to describe event1,  
and agent1 is the agent of event1,  
and you have described agent1,  
and object1 is the object of event1,  
then describe object1.

Together with the agent rule shown above, this production lets AMBER produce utterances such as "Daddy ball". Thus, the model provides a simple explanation of why children omit some content words in their early multi-word utterances. Such rules must be constructed many times before they become strong enough to have an effect, but eventually they let the system produce telegraphic sentences containing all relevant content words in the standard order and lacking only grammatical morphemes.

### 4. Learning Suffixes and Prefixes

Once AMBER begins to correctly predict content words, it can learn rules for saying grammatical morphemes as well. As with content words, such rules are created when the system hears a morpheme but fails to predict it in that position. For example, suppose the program hears the sentence "Daddy \* is bounce ing \* the ball",<sup>4</sup> but predicts only "Daddy bounce ball". In this case, the following rule is generated:

**ING-1**

If you have described action1,  
and action1 is the action of event1,  
then say ING.

Once it has gained sufficient strength, this rule will say the morpheme "ing" after any action word. As stated, the production is overly general and will lead to errors of commission. I consider AMBER's response to such errors in the following section.

<sup>4</sup>Asterisks represent pauses in the adult sentence. These cues are necessary for AMBER to decide that a morpheme like "is" is a prefix for "bounce" instead of a suffix for "Daddy".

The omission of prefixes leads to very similar rules. In the above example, the morpheme "is" was omitted before "bounce", leading to the creation of a prefix rule for producing the missing function word:

**IS-1**

If you want to describe action1,  
and action1 is the action of event1,  
then say IS.

Note that this rule will become true *before* an action has been described, while the rule *ing-1* can apply only *after* the goal to describe the action has been satisfied. AMBER uses such conditions to control the order in which morphemes are produced.

Figure 1 shows AMBER's mean length of utterance as a function of the number of sample sentences (taken in groups of five) seen by the program.<sup>5</sup> As one would expect, the system starts with an average of around one word per utterance, and the length slowly increases with time. AMBER moves through a two-word and then a three-word stage, until it eventually produces sentences lacking only grammatical morphemes. Finally, the morphemes are included, and adult-like sentences are produced. The incremental nature of the learning curve results from the piecemeal way in which AMBER learns rules for producing sentences, and from the system's reliance on the strengthening process.

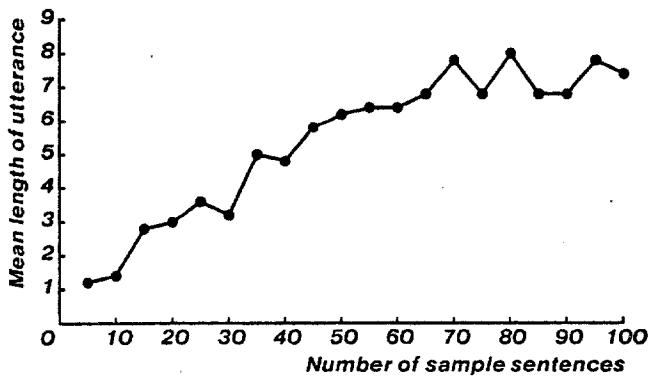


Figure 1. Mean length of AMBER's utterances.

## 5. Recovering from Errors of Commission

Errors of commission occur when AMBER predicts a morpheme that does not occur in the adult sentence. These errors result from the overly general prefix and suffix rules that we saw in the last section. In response to such errors, AMBER calls on a discrimination routine in an attempt to generate more conservative productions with additional conditions.<sup>6</sup> Earlier, I considered a rule (*is-1*) for producing "is" before the action of an event. As stated, this rule would apply in inappropriate situations as well as correct ones. For example, suppose that AMBER learned this rule in the context of the sentence "Daddy *is* bounce ing the ball". Now suppose the system later uses this rule to predict the same sentence, but that it instead hears the sentence "Daddy *was* bounce ing the ball".

<sup>5</sup>AMBER is implemented on a PDP KL-10 in PRISM (Langley and Neches, 1981), an adaptive production system language designed for modeling learning phenomena; the run summarized in Figure 1 took approximately 2 hours of CPU time.

At this point, AMBER's discrimination routine would retrieve the rule responsible for predicting "is" and lowers its strength; it would also retrieve the situation that led to the faulty application, passing this information to the discrimination routine. Comparing the earlier good case to the current bad case, the discrimination mechanism finds only one difference – in the good example, the action node was marked *present*, while no such marker occurred during the faulty application. The result is a new production that is identical to the original rule, except that an additional condition has been included:

**IS-2**

If you want to describe action1,  
and action1 is the action of event1,  
and action1 is in the present,  
then say IS.

This new condition will let the variant rule fire only when the action is marked as occurring in the present. When first created, the *is-2* production is too weak to be seriously considered. However, as it is learned again and again, it will eventually come to mask its predecessor. This transition is aided by the weakening of the faulty *is-1* rule each time it leads to an error.

Once the variant production has gained enough strength to apply, it will produce its own errors of commission. For example, suppose AMBER uses the *is-2* rule to predict "The boy *s* *is* bounce ing the ball", while the system hears "The boy *s* *are* bounce ing the ball". This time the difference is more complicated. The fact that the action had an agent in the good situation is no help, since an agent was present during the faulty firing as well. However, the agent was *singular* in the first case but not during the second. Accordingly, the discrimination mechanism creates a second variant:

**IS-3**

If you want to describe action1,  
and action1 is the action of event1,  
and action1 is in the present,  
and agent1 is the agent of event1,  
and agent1 is singular,  
then say IS.

The resulting rule contains two additional conditions, since the learning process was forced to chain through two elements to find a difference. Together, these conditions keep the production from saying the morpheme "is" unless the agent of the current action is singular in number.

Note that since the discrimination process must learn these sets of conditions separately, an important prediction results: *the more complex the conditions on a morpheme's use, the longer it will take to master*. For example, three sets of conditions are required for the "is" rule, while only a single condition is needed for the "ing" production. As a result, the former is mastered after the latter, just as found in children's speech. Table 1 presents the order of acquisition for the six classes of morpheme learned by AMBER, and the order in which the same morphemes were mastered by Brown's children. The number of sample sentences the model required before mastery are also included.

<sup>6</sup>Anderson's ALAS (1981) system uses a very similar process to recover from overly general morpheme rules. AMBER and ALAS have much in common, both having grown out of discussions between Anderson and the author. Although there is considerable overlap, ALAS generally accounts for later developments in children's speech than does AMBER.

The general trend is very similar for the children and the model, but two pairs of morphemes are switched. For AMBER, the plural construction was mastered before "ing", while in the observed data the reverse was true. However, note that AMBER mastered the progressive construction almost immediately after the plural, so this difference does not seem especially significant. Second, the model mastered the articles "the", "a", and "some" before the construction for past tense. However, Brown has argued that the notions of "definite" and "indefinite" may be more complex than they appear on the surface; thus, AMBER's representation of these concepts as single features may have oversimplified matters, making articles easier to learn than they are for the child.

Thus, the discrimination process provides an elegant explanation for the observed correlation between a morpheme's complexity and its order of acquisition. Observe that if the conditions on a morpheme's application were learned through a process of generalization such as that proposed by Winston (1970), exactly the opposite prediction would result. Since generalization operates by removing conditions which differ in successive examples, simpler rules would be finalized later than more complex ones. Langley (1982) has discussed the differences between generalization-based and discrimination-based approaches to learning in more detail.

CILDREN'S ORDER	AMBER'S ORDER	LEARNING TIME
PROGRESSIVE	PLURAL	59
PLURAL	PROGRESSIVE	63
PAST TENSE	ARTICLES	166
ARTICLES	PAST TENSE	186
THIRD PERSON	THIRD PERSON	283
AUXILIARY	AUXILIARY	306

Table 1. Order of morpheme mastery by the child and AMBER.

Some readers will have noted the careful crafting of the above examples, so that only one difference occurred in each case. This meant that the relevant conditions were obvious, and the discrimination mechanism was not forced to consider alternate corrections. In order to more closely model the environment in which children learn language, AMBER was presented with randomly generated sentence/meaning pairs. Thus, it was usually impossible to determine the correct discrimination that should be made from a single pair of good and bad situations.

AMBER's response to this situation is to create all possible discriminations, but to give each of the variants a low initial strength. Correct rules, or rules containing at least some correct conditions, are learned more often than rules containing spurious conditions. And since AMBER strengthens a production whenever it is relearned, variants with useful conditions come to be preferred over their competitors. Thus, AMBER may be viewed as carrying out a *breadth-first search* through the space of possible rules, considering many alternatives at the same time, and selecting the best of these for further attention. Only variants that exceed a certain threshold (generally those with correct conditions) lead to new errors of commission and additional variants. Eventually, this search process leads to the correct rule, even in the presence of many irrelevant features.

Figure 2 presents the learning curves for the "ing" morpheme. Since AMBER initially lacks an "ing" rule, errors of commission abound at the outset, but as this production and its variants are strengthened, such errors decrease. In contrast, errors of commission are absent at the beginning, since AMBER lacks an "ing" rule to make false predictions. As the morpheme rule becomes stronger, errors of commission grow to a peak, but they disappear as discrimination takes effect. By the time it has seen 63 sample sentences, the system has mastered the present progressive construction.

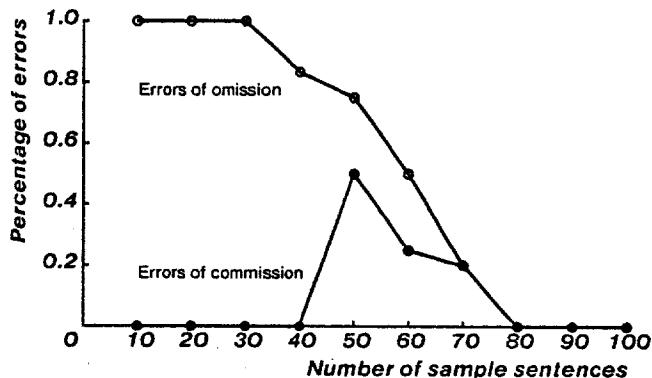


Figure 2. AMBER's learning curves for the morpheme "ing".

## 6. Directions for Future Research

In the preceding pages, we have seen that AMBER offers explanations for a number of phenomena observed in children's early speech. These include the omission of content words and morphemes, the gradual manner in which these omissions are overcome, and the order in which grammatical morphemes are mastered. As a psychological model of early syntactic development, AMBER constitutes an improvement over previous language learning programs. However, this does not mean that the model can not be improved, and in this section I outline some directions for future research efforts.

### 6.1. Simplicity and Generality

One of the criteria by which any scientific theory can be judged is *simplicity*, and this is one dimension along which AMBER could stand some improvement. In particular, some of AMBER's learning heuristics for coping with errors of omission incorporate considerable knowledge about the task of learning a language. For example, AMBER knows the *form* of the rules it will learn for ordering goals and producing morphemes. Another questionable piece of information is the distinction between major and minor meanings that lets AMBER treat content words and morphemes as completely separate entities. One might argue that the child is born with such knowledge, so that any model of language acquisition should include it as well. However, until such innateness is proven, any model that can manage without such information must be considered simpler, more elegant, and more desirable than a model that requires it to learn a language.

In contrast to these domain-specific heuristics, AMBER's strategy for dealing with errors of commission incorporates an apparently domain-independent learning mechanism – the discrimination process. This heuristic can be applied to any domain in which overly general rules lead to errors, and can be used on a variety of representations to discover the conditions under which such rules should be selected. In addition to language development, the discrimination process has been applied to concept learning (Anderson, Kline, and Beasely, 1979; Langley, 1982) and strategy acquisition (Brazdil, 1978; Langley, 1982). Langley (1982) has discussed the generality and power of discrimination-based approaches to learning in greater detail. As we shall see below, this heuristic may provide a more plausible explanation for the learning of word order. Moreover, it opens the way for dealing with some aspects of language acquisition that AMBER has so far ignored – the learning of word/concept links and the mastering of irregular constructions.

## 6.2. Learning Word Order Through Discrimination

AMBER learns the order of content words through a two-stage process, first learning to prefer some relations (like agent) over others (like action or object), and then learning the relative orders in which such relations should be described. The adaptive productions responsible for these transitions contain the actual *form* of the rules that are learned; the particular rules that result are simply instantiations of these general forms. Ideally, future versions of AMBER should draw on more general learning strategies to acquire ordering rules.

Let us consider how the discrimination mechanism might be applied to the discovery of such rules. In the existing system, the generation of "ball" without a preceding "Daddy" is viewed as an error of omission. However, it could as easily be viewed as an error of *commission* in which the goal to describe the object was prematurely satisfied. In this case, one might use discrimination to generate a variant version of the *start* rule:

```
If you want to describe node1,
and node2 is the object of node1,
and node3 is the agent of node1,
and you have described node3,
then describe node2.
```

This production is similar to the *start* rule, except that it will set up goals only to describe the object of an event, and then only if the agent has already been described. In fact, this rule is identical to the *agent-object* rule discussed in an earlier section; the important point is that it is also a special case of the *start* rule that might be learned through discrimination when the more general rule fires inappropriately. The same process could lead to variants such as the *agent* rule, which express preferences rather than order information. Rather than starting with knowledge of the forms of rules at the outset, AMBER would be able to determine their form through a more general learning heuristic.

## 6.3. Major and Minor Meanings

The current version of AMBER relies heavily on the representational distinction between major meanings and modulations of those meanings. Unfortunately, some languages express through content words what others express through grammatical morphemes. Future versions of the system should lessen this distinction by using the same representation for both types of information. In addition, the model might employ a single production for learning to produce both content words

and morphemes; thus, the program would lack the *speak* rule described earlier, but would construct specific versions of this production for particular words and morphemes. This would also remedy the existing model's inability to learn new connections between words and concepts. Although the resulting rules would probably be overly general, AMBER would be able to recover from the resulting errors by additional use of the discrimination mechanism.

The present model also makes a distinction between morphemes that act as prefixes (such as "the") and those that act as suffixes (such as "ing"). Two separate learning rules are responsible for recovering from function word omissions, and although they are very similar, the conditions under which they apply and the resulting morpheme rules are different. Presumably, if a single adaptive production for learning words and morphemes were introduced, it would take over the functions of both the prefix and suffix rules. If this approach can be successfully implemented, then the current reliance on pause information can be abandoned as well, since the pauses serve only to distinguish suffixes from prefixes.

Such a reorganization would considerably simplify the theory, but it would also lead to two complications. First, the resulting system would tend to produce utterances like "Daddy ed" or "the bounce" before it learned the correct conditions on morphemes through discrimination. (This problem is currently avoided by including information about the relation when a morpheme rule is first built, but this requires domain-specific knowledge about the language learning task.) Since children very seldom make such errors, some other mechanism must be found to explain their absence, or the model's ability to account for the observed phenomena will suffer.

Second, if pause information (and the ability to take advantage of such information) is removed, the system will sometimes decide a prefix is a suffix and vice versa. For example, AMBER might construct a rule to say "ing" before the object of an event is described, rather than after the action has been mentioned. However, such variants would have little effect on the system's overall performance, since they would be weakened if they ever led to deviant utterances, and they would tend to be learned less often than the desired rules in any case. Thus, the strengthening and weakening processes would tend to direct search through the space of rules toward the correct segmentation, even in the absence of pause information.

## 6.4. Mastering Irregular Constructions

Another of AMBER's limitations lies in its inability to learn irregular constructions such as "men" and "ate". However, by combining discrimination and the approach to learning word/concept links described above, future implementations should fare much better along this dimension. For example, consider the irregular noun "foot", which forms the plural "feet". Given a mechanism for connecting words and concepts, AMBER might initially form a rule connecting the concept \**foot* to the word "foot". After gaining sufficient strength, this rule would say "foot" whenever seeing an example of the concept \**foot*. Upon encountering an occurrence of "feet", the system would note the error of commission and call on discrimination. This would lead to a variant rule that produced "foot" only when a *single* marker was present. Also, a new rule connecting \**foot* to "feet" would be created. Eventually, this new rule would also lead to errors of commission, and a variant with a *plural* condition would come to replace it.

Dealing with the rule for producing the plural marker "s" would be somewhat more difficult. Although AMBER might initially learn to say "foot" and "feet" under the correct circumstances, it would eventually learn the general rule for saying "s" after plural agents and objects. This would lead to constructions such as "feet s", which have been observed in children's utterances. The system would have no difficulty in detecting such errors of commission, but the appropriate response is not so clear. Conceivably, AMBER could create variants of the "s" rule which stated that the concept to be described must not be \*foot. However, a similar condition would also have to be included for every situation in which irregular pluralization occurred (deer, man, cow, and so on). Similar difficulties arise with irregular constructions for the past tense.

A better solution would have AMBER construct a special rule for each irregular word, which "imagined" that the inflection had already been said. Once these productions became stronger than the "s" and "ed" rules, they would prevent the latter's application and bypass the regular constructions in these cases. Overly general constructions like "foot s" constitute a related form of error. Although AMBER would generate such mistakes before the irregular form was mastered, it would not revert to the overgeneral regular construction at a later point, as do many children. The area of irregular constructions is clearly a phenomenon that deserves more attention in the future.

## 7. Conclusions

In conclusion, AMBER provides explanations for several important phenomena observed in children's early speech. The system accounts for the one-word stage and the child's transition to the telegraphic stage. Although AMBER and children eventually learn to produce all relevant content words, both pass through a stage where some are omitted. Because it learns sets of conditions one at a time, the discrimination process explains the order in which grammatical morphemes are mastered. Finally, AMBER learns gradually enough to provide a plausible explanation of the incremental nature of first language acquisition. Thus the system constitutes a significant addition to our knowledge of syntactic development.

Of course, AMBER has a number of limitations that should be addressed in future research. Successive versions should be able to learn the connections between words and concepts, should reduce the distinction between content words and morphemes, and should be able to master irregular constructions. Moreover, they should require less knowledge of the language learning task, and rely more of domain-independent learning mechanisms such as discrimination. But despite its limitations, the current version of AMBER has proven itself quite useful in clarifying the incremental nature of language acquisition, and future models promise to further our understanding of this complex process.

## References

- Anderson, J. R. Induction of augmented transition networks. *Cognitive Science*, 1977, 1, 125-157.
- Anderson, J. R. A theory of language acquisition based on general learning principles. *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, 1981.
- Anderson, J. R., Kline, P. J., and Beasley, C. M. A general learning theory and its application to schema abstraction. In G. H. Bower (ed.), *The Psychology of Learning and Motivation*, Volume 13, 1979.
- Berwick, R. Computational analogues of constraints on grammars: A model of syntactic acquisition. *Proceedings of the 18th Annual Conference of the Association for Computational Linguistics*, 49-53, 1980.
- Brazdil, P. Experimental learning model. *Proceedings of the AISB Conference*, 1978, 46-50.
- Brown, R. *A First Language: The Early Stages*. Cambridge, Mass.: Harvard University Press, 1973.
- Feldman, J. A., Gips, J., Horning, J. J., and Reder, S. Grammatical complexity and inference. Technical Report No. CS 125, Computer Science Department, Stanford University, 1969.
- Hedrick, C. Learning production systems from examples. *Artificial Intelligence*, 1976, 7, 21-49.
- Horning, J. J. A study of grammatical inference. Technical Report No. CS 139, Computer Science Department, Stanford University, 1969.
- Kelley, K. L. Early syntactic acquisition. Rand Report P-3719, 1967.
- Klein, S. Automatic inference of semantic deep structure rules in generative semantic grammars. Technical Report No. 180, Computer Sciences Department, University of Wisconsin, 1973.
- Langley, P. A general theory of discrimination learning. To appear in Klahr, D., Langley, P., and Neches, R. T. (eds.) *Self-Modifying Production System Models of Learning and Development*, 1982.
- Langley, P. and Neches, R. T. PRISM User's Manual. Technical Report, Department of Computer Science, Carnegie-Mellon University, 1981.
- Reeker, L. H. The computational study of language acquisition. In M. Yovits and M. Rubinoff (eds.), *Advances in Computers*, Volume 15. New York: Academic Press, 1976.
- Selfridge, M. A computer model of child language acquisition. *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, 1981, 92-96.
- Sembugamoorthy, V. PLAS, a paradigmatic language acquisition system: An overview. *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, 1979, 788-790.
- Siklossy, L. Natural language learning by computer. In H. A. Simon and L. Siklossy (eds.), *Representation and Meaning: Experiments with Information Processing Systems*. Englewood Cliffs, N. J.: Prentice-Hall, 1972.
- Solomonoff, R. A new method for discovering the grammars of phrase structure languages. *Proceedings of the International Conference on Information Processing*, UNESCO, 1959.
- Waterman, D. A. Adaptive production systems. *Proceedings of the Fourth International Joint Conference on Artificial Intelligence*, 1975, 296-303.
- Winston, P. H. Learning structural descriptions from examples. MIT AI-TR-231, 1970.
- Wolff, J. G. Language acquisition and the discovery of phrase structure. *Language and Speech*, 1980, 23, 255-269.