# Dispelling Myths about Language Bootstrapping

Jeffrey Mark Siskind*
M. I. T. Artificial Intelligence Laboratory
545 Technology Square, Room NE43–800b
Cambridge MA 02139
617/253–5659
internet: Qobi@AI.MIT.EDU

**Abstract**

Two competing theories have been proposed to explain how children begin acquiring language without any prior linguistic experience. The first, semantic bootstrapping, claims that children first acquire word meanings and then use this information to drive acquisition of syntax. The second, syntactic bootstrapping, claims the inverse, that children use some syntactic knowledge in figuring out what words mean. There are difficulties with both approaches. Semantic bootstrapping on one hand, requires a referential completeness assumption, that children possess a concrete understanding of the referent of each word before assigning a lexical category to that word and before formulating syntactic generalizations around those category assignments. Syntactic bootstrapping on the other hand, requires that children be able to recover the phrase boundaries of utterances, without the use of syntax, and be able to isolate verbs prior to knowing their meaning. Proponents of both theories argue their case by claiming that in principle, language acquisition is impossible without such assumptions. This paper, an extension of work reported by Siskind (1990), attempts to refute such claims by presenting a set of principles, implemented as an algorithm, that can simultaneously acquire syntactic parameters of $\overline{\mathrm{X}}$ theory and a lexicon comprising both category and semantic information from a training corpus containing both linguistic and non-linguistic input. Before training, the algorithm does not possess a fixed grammar of the target language, nor any information, syntactic or semantic, about the words to be learned. No referential completeness assumption is made, nor does the algorithm require knowledge of the phrase structure of, or the lexical category of any word in, the linguistic input. The successful operation of the algorithm is demonstrated on training sessions both in English and in Japanese.

## 1 Introduction

This paper addresses an issue in language acquisition which has become known as the *bootstrapping problem.* While in the later stages of language acquisition, children are assisted by previously acquired linguistic knowledge, how do children begin the language acquisition task without such knowledge? In particular, how do they assign syntactic categories and semantic representations to the words they hear as part of complete utterances?

Two competing theories have been proposed for solving this problem. The first, due to Grimshaw (1979, 1981) and Pinker (1984) states roughly that children first acquire the meanings of words and then use this information to derive the syntactic constraints of their language. This theory has become known as *semantic bootstrapping.* Semantic bootstrapping assumes that children first learn the meanings of words, by some unspecified mechanism. They then apply a default mapping to assign a syntactic category to each word based on its semantic category. In particular, THINGS are mapped to nouns and EVENTS are mapped to verbs. Such a default mapping has been termed a Canonical Structure Realization. Finally, syntactic rules are formed around these abstract syntactic categories which later generalize to cases where the words are not of the appropriate semantic class but are nonetheless of the appropriate syntactic class. For example, a child hearing the utterance *Fido barked* knows that *Fido* is a dog and that *bark* is an action and thus maps *Fido* to a noun and *bark* to a verb and forms the grammar rule $S \rightarrow N\ V$ as a template to account for the utterance. Elliott and Wexler (1986) propose a variant of of semantic bootstrapping which requires only that children map THINGS to nouns. Principles of universal grammar then assist the remainder

---

[*] This research is supported in part by a Presidential Young Investigator Award to Professor Robert C. Berwick under National Science Foundation Grant DCR–85552543, by a grant from the Siemens Corporation, and by the Kapor Family Foundation. Part of this research was performed while the author was supported by an AT&T Bell Laboratories Ph.D. scholarship and while the author was visiting Xerox PARC as a research intern and as a consultant.

of the bootstrapping process. Their scheme however, requires that children be able to recover the bracketed phrase structure of the utterances they hear solely from acoustic and prosodic information.

The second theory, due to Gleitman (1990), states roughly the converse: that children use syntactic information to acquire word meanings. This theory has become known as *syntactic bootstrapping*. Like Elliott and Wexler, Gleitman also assumes that children are able to recover the phrase structure of utterances from acoustic information alone, and that they use this phrase structure to derive the subcategorization frames associated with each verb. She then proposes that key elements of a verb's meaning can be derived solely from its subcategorization frame. For example, a child hearing the utterance *John told Mary that Bill left* will deduce that the verb *told* takes an NP complement and an $\overline{\mathrm{S}}$ complement and thus is likely to be a verb of communication. Gleitman's method acquires word meanings from the utterances alone, without any reference to the non-linguistic context of the utterances. Brent (1990) has used Gleitman's method to learn components of word meaning by scanning large text corpora.

Proponents of both semantic and syntactic bootstrapping support their case primarily by arguing that *in principle* language acquisition must work their way as it is impossible to explain language acquisition without such an assumption. Grimshaw and Pinker (1990) attempt to refute Gleitman's claims by highlighting the fact that her theory does not offer a complete account of how verb meanings are acquired. They argue, that while her theory is in principle plausible, she has yet to prove that it is actually both a necessary and correct account of child language acquisition.

This paper takes a different approach. It argues that neither semantic nor syntactic bootstrapping are necessary to account for language bootstrapping. It does so by demonstrating an algorithm called DAVRA[1] which determines the syntactic category and meaning of words without relying on semantic or syntactic bootstrapping. It does not argue that semantic and syntactic bootstrapping are *wrong*, just that they are not *in principle necessary*. Likewise, there is no claim that DAVRA is a correct account of child language acquisition, simply that it is a plausible account. Determining which account is actually correct awaits further research.

DAVRA relies on a collection of syntactic and semantic principles, collectively termed Universal Grammar. Following the poverty of stimulus argument, DAVRA assumes that the language learner is innately endowed with a language faculty which incorporates the principles of Universal Grammar. These principles are summarized in Section 2. Unlike previous work (Siskind, 1990) which assumed a known fixed context free grammar prior to language acquisition, DAVRA uses instead a direct encoding of $\overline{\mathrm{X}}$ theory including a capability for parametric variation in the language to be learned. DAVRA has successfully been applied both to English (Section 4) and Japanese (Section 5) examples, learning the correct $\overline{\mathrm{X}}$ parameter settings for each. Several key points about this work deserve particular emphasis.

- A common assumption about language acquisition dating as far back as Locke (1690) is that children are presented with single word utterances, such as 'milk', in a context where it is clearly evident that 'milk' refers to milk. This is termed *referential completeness* and is a key assumption underlying semantic bootstrapping (Bloom 1990). Heath (1983, 1989 p. 338) gives evidence that in some cultures, children are rarely presented with referentially complete stimuli yet they successfully learn language. DAVRA is not limited to single word utterances and furthermore allows the learner uncertainty in associating meanings with utterances.
- The learner starts out without knowing the meaning or syntactic category of *any* words in the linguistic input. This differs from some prior work (Granger 1977) which learns the syntactic category or meaning of words appearing in utterances with but a single unknown word, using the context of the remaining known words as a filter on possible syntactic category and meaning assignments for the unknown word. Furthermore, the learner starts out without knowing the $\overline{\mathrm{X}}$ parameter settings of the language being learned. At the completion of the training session, the learner has acquired
  - a meaning for each word in the training session,
  - a syntactic category for each word in the training session and
  - the $\overline{\mathrm{X}}$ parameter settings for the syntax of the language learned.

  This is true bootstrapping from nothing more than principles of Universal Grammar.
- We do not assume that the learner has access, via prosody, to the bracketed phrase structure of the linguistic input.

The key to the success of our paradigm is cross-situational learning. A number of prior approaches to language acquisition, in particular Elliott and Wexler (1986) and Lasnik (1989), attempt to demonstrate learning from a

---

[1] DAVRA or דברא is a make-believe Aramaic word for *word*.

single utterance. We believe that in most situations, a single utterance does not offer enough constraint to uniquely determine either parameter settings or syntactic categories and meanings of words. Instead, we believe that the learner must find a lexicon and parameter settings which can simultaneously and consistently explain multiple utterances. Words that co-occur across multiple utterances are the keys which enable the learner to decipher the language acquisition puzzle. Note that this is not a form of distributional learning. In its classic form, distributional learning infers equivalence classes between word by observing two different words occurring in the same location within otherwise equivalent utterances. We make no such restriction on the form of the input nor do we require the learner to hypothesize any semantic similarity between a group of words to classify them as the same syntactic category.

## 2 A Linguistic Theory Supporting Language Acquisition

The linguistic theory incorporated into DAVRA is characterized by the following principles. We assume that the learner is innately endowed with a language faculty which operates according to these principles.

1. The learner is able to distinguish between linguistic and non-linguistic input. Normally, linguistic input is available on the auditory channel while non-linguistic input is available on the visual channel though this is not always the case. Whatever channels carry the linguistic and non-linguistic information (they may in fact be the same channel) the learner is able to separate and distinguish the linguistic from the non-linguistic information.

2. The learner is able to segment the linguistic input into sentences, to segment those sentences into words and to group different occurrences of the same word into the same equivalence class despite minor acoustic variation between occurrences.

3. The learner is equipped with a mechanism for representing meanings of individual words and entire utterances. All we require is that utterance meanings be represented by ground expressions in some calculus and that word meanings be represented by expressions in the same calculus, possibly containing variables. In this paper we arbitrarily take Jackendoff's (1983) conceptual structures as our meaning calculus. Thus the meaning of the utterance *The cup slid from John to Mary* would be $\text{GO}(\textbf{cup}, [_{\text{Path}}\ \text{FROM}(\textbf{John}), \text{TO}(\textbf{Mary})])$ and the meaning of the word *slid* would be $\text{GO}(x, [_{\text{Path}}\ y, z])$. A companion paper (Siskind, 1991) discusses the inadequacy of this representation and proposes an alternate representation.

4. The learner is exposed to utterances in a single language. Each utterance the learner hears is grammatically correct in that language and the learner is able to associate each utterance with a set of possible meanings for that utterance. One of those possible meanings must actually be the correct meaning of the utterance. The learner's innate perceptual abilities combined with her naive theories of physics and psychology allow her to postulate plausible meanings for each utterance. Siskind (1991) proposes a mechanism for how this may be done. Note that we do not require that the learner associate a *single* meaning with each utterance, rather that the learner postulate a set of plausible meanings, only one of which need be the actual meaning of the utterance. Future work will relax this constraint even further, allowing for some ungrammatical utterances or utterances for which none of the possible meanings associated with that utterance by the learner turn out to be correct.

5. The learner parses each input utterance according to the following variant of $\overline{\text{X}}$ theory:
   (a) Each nonterminal node in the parse tree has either one or two daughters.
   (b) The lexical categories are N, V, P and I.
   (c) Each lexical category X projects into the categories $\text{X}_{\text{SPEC}}$, $\overline{\text{X}}$ and XP.
   (d) Each utterance that the learner hears is of category IP.
   (e) $\text{I}_{\text{SPEC}}$ is processed as NP.
   (f) $\overline{\text{I}}$ is processed as VP. This differs somewhat from current linguistic theory and is done to simplify DAVRA. Future work will discuss modifications to DAVRA which handle $\overline{\text{I}}$ in accord with current linguistic theory.
   (g) The language the learner hears is either a *SPEC initial* language or a *SPEC final* language. If the language is SPEC initial then for every lexical category X, the language follows the rule

$$\text{XP} \rightarrow \text{X}_{\text{SPEC}}\ \overline{\text{X}}.$$

   If the language is SPEC final then for every lexical category X, the language follows the rule

$$\text{XP} \rightarrow \overline{\text{X}}\ \text{X}_{\text{SPEC}}.$$

(h) The language the learner hears is either a *head initial* language or a *head final* language. If the language is head initial then for every lexical category X (except for I) the language follows the rule

$$\overline{\mathrm{X}} \rightarrow \overline{\mathrm{X}}\ \mathrm{YP}$$

for every lexical category Y. If the language is head final then for every lexical category X (except for I) the language follows the rule

$$\overline{\mathrm{X}} \rightarrow \mathrm{YP}\ \overline{\mathrm{X}}$$

for every lexical category Y. Furthermore, irrespective of whether the language is head initial or final, the language also follows the rule

$$\overline{\mathrm{X}} \rightarrow \mathrm{X}$$

for every lexical category X (except for I).

(i) The categories $\mathrm{X}_{\mathrm{SPEC}}$ (except for $\mathrm{I}_{\mathrm{SPEC}}$) and lexical categories X (except for I) are terminal.

6. A meaning is associated with each node in the parse tree. The meanings associated with terminals nodes are word meanings from the lexicon. The meaning associated with the root node is one of the meanings postulated for the utterance. The meanings associated with nonterminal nodes are related by the following *linking rule.*
   (a) If a node has a single daughter, then the meaning of the parent and the daughter are the same.
   (b) If a node $X$ has two daughters $Y$ and $Z$, then one of the daughters is called the *template* and the other is called the *argument.* We will call $X$ the *resultant.* The resultant meaning is derived from the template meaning by renaming the variables of the argument meaning so that they are distinct from those in the template meaning and then substituting the argument meaning for all occurrences of some variable in the template meaning. Alternatively, the argument meaning may be the distinguished symbol $\perp$, in which case the resultant meaning is the same as the template meaning.
7. Nodes of category $\overline{\mathrm{X}}$ are templates while nodes of category $\mathrm{X}_{\mathrm{SPEC}}$ and XP are arguments.
8. Argument meanings must be variable-free.
9. A word cannot have a meaning which is just a variable.
10. A node of category XP cannot have $\perp$ as its meaning.
11. The following exceptions notwithstanding, any terminal can be non-overt, i.e. it may have no overt word associated with it.
    (a) The semantics of a node with no overt descendants must be $\perp$.
    (b) Nodes of category $\overline{\mathrm{X}}$ must have at least one overt descendant.
12. The learner observes a *monosemy* constraint, i.e. the learner will assign each word at most one syntactic category and one meaning. Future work will relax this constraint. Other work in language acquisition often assumes a converse constraint that each distinct possible meaning be conveyed by at most one distinct word. Note that we do *not* require such a constraint ruling out synonyms.

Note that the above principles do not account for movement. While dealing with movement adds significant complexity to this system, there does not seem to be any reason why it could not be incorporated in a fashion analogous to the techniques used in this paper. This is a fruitful area for future research.

## 3 The Algorithm

DAVRA is presented in a nondeterministic dialect of COMMON LISP known as SCREAMER (Siskind, 1991). DAVRA has been implemented and correctly processes the examples given in Sections 4 and 5. For the sake of brevity, only portions of the code are included in this section. The complete code is available from the author. Use of a nondeterministic dialect allows a straightforward and transparent encoding of the principles of Universal Grammar directly as statements in the program. While useful for pedagogical purposes, more efficient implementations are possible. Siskind (1990) discusses one such algorithm (called MAIMRA) for a linguistic theory which is similar to, though not identical to, the one presented in Section 2.

```
(proclaim '(special categories head-initial? spec-initial? lexicon))
```

```
(defun-nondeterministic fracture-words (words category meaning)
```

The essence of Davra is the routine `fracture-words`. `Fracture-words` attempts to assign a syntactic `category` and `meaning` to a list of `words`. The basic strategy is a top down one: nondeterministically split `words` into two phrases, a template and an argument; nondeterministically assign part of the resultant `meaning` to the template and part to the argument according to the linking rule (principle 6); and recursively call `fracture-words` on both the template and argument. This routine uses four pieces of information global to the language acquisition process: the lexical `categories` which project into the $\overline{\mathrm{X}}$ system, a flag which indicates whether the language is `head-initial?` or final, another flag which indicates whether the language is `spec-initial?` or final and the `lexicon`, which is a map from words to their syntactic categories and meanings.

```
  (if (and (consp category) (eq (second category) 'p) (eq meaning 'semanticless)) (fail))
```

This implements principle 10 that a node of category XP cannot have $\perp$ as its meaning.

```
  (if (and (null words) (not (eq meaning 'semanticless))) (fail))
```

This implements principle 11a that the semantics of a node with no overt descendants must be $\perp$.

```
  (cond
   ((equal category '(i spec)) (fracture-words words '(n p) meaning))
   ((equal category '(i bar)) (fracture-words words '(v p) meaning))
```

There are five cases in the `fracture-words` routine. These two cases implement principles 5e and 5f (that $\mathrm{I}_{\mathrm{SPEC}}$ is processed as NP and that $\overline{\mathrm{I}}$ is processed as VP).

```
   ((and (consp category) (eq (second category) 'bar))
    (either
      (fracture-words words (first category) meaning)
```

The third case handles phrases of type $\overline{\mathrm{X}}$. It implements principles 5h and 6a. A node of category $\overline{\mathrm{X}}$ can be handled by one of two rules:

$$\overline{\mathrm{X}} \rightarrow \mathrm{X}$$
$$\overline{\mathrm{X}} \rightarrow \overline{\mathrm{X}}\ \mathrm{YP}$$

A nondeterministic choice is made between the two by the `either` clause.

```
      (let* ((split (split words))
             (template (if head-initial? (first split) (second split)))
             (argument (if head-initial? (second split) (first split))))
        (if (null template) (fail))
        (if (null argument) (fail))
        (let ((argument-meaning (possible-argument-meaning meaning)))
          (fracture-words argument `(,(member-of categories) p) argument-meaning)
          (fracture-words
           template category (possible-template-meaning argument-meaning meaning))))))
```

This implements the second alternative for phrases of type $\overline{\mathrm{X}}$. It incorporates principles 5h, 6b, 7, 10, 11a and 11b. It nondeterministically splits the phrase into two halfs, one to become the `template`, the other to become the `argument`. By principle 5h, the choice of which half becomes the `template` and which the `argument` is determined by the `head-initial?` parameter. By principles 7 and 11b, the `template` must not be null. Furthermore, by a combination of principles 7, 10 and 11a, the `argument` must also not be null. The routines `possible-argument-meaning` and `possible-template-meaning` implement the linking process (principle 6b) in reverse. Given a resultant `meaning`, they nondeterministically return all possible template meanings and `argument-meanings` that can combine to form the resultant `meaning`. They will be described in greater detail later. Two recursive calls are made to `fracture-words`, one to fracture the `argument` as a phrase of some category YP, for any lexical category Y, and one to fracture the `template` as a phrase of category $\overline{\mathrm{X}}$.

```
  ((and (consp category) (eq (second category) 'p))
   (let* ((split (split words))
          (template (if spec-initial? (second split) (first split)))
          (argument (if spec-initial? (first split) (second split))))
     (if (null template) (fail))
     (let ((argument-meaning (possible-argument-meaning meaning)))
       (fracture-words argument `(,(first category) spec) argument-meaning)
       (fracture-words
        template
        `(,(first category) bar)
        (possible-template-meaning argument-meaning meaning)))))
```

The fourth case handles phrases of type XP. It incorporates principles 5g, 6b, 7 and 11b. Like before, it nondeterministically splits the phrase into two halfs, one to become the `template`, the other to become the `argument`. By principle 5g, the choice of which half becomes the `template` and which the `argument` is determined by the `spec-initial?` parameter. By principles 7 and 11b, the `template` must not be null. Like before, the resultant `meaning` is nondeterministically divided into a template meaning and an `argument-meaning` to implement principle 6b. Two recursive calls are made to `fracture-words`, one to fracture the `argument` as a phrase of category $X_{\mathrm{SPEC}}$ and one to fracture the `template` as a phrase of category $\overline{\mathrm{X}}$.

```
  ((or (and (consp category) (eq (second category) 'spec)) (symbolp category))
   (unless (null words)
     (unless (null (rest words)) (fail))
     (let* ((new-definition (list category (canonicalize-meaning meaning)))
            (old-definition (gethash (first words) lexicon)))
       (if old-definition
           (unless (equal new-definition old-definition) (fail))
           (locally-setf (gethash (first words) lexicon) new-definition))))))
```

The final case handles terminals. According to principle 5i, categories $X_{\mathrm{SPEC}}$ and lexical categories X are terminal. By principle 6, a lexical entry comprising a syntactic category and meaning is created. If this word already has a different lexical entry then enforce the monosemy constraint (principle 12) by failing. Principle 11 allows a terminal to be non-overt. In this case no lexical entry is added to the lexicon.

```
(defun-nondeterministic subexpression (expression)
  (if (consp expression)
      (either expression (subexpression (member-of (rest expression))))
      expression))

(defun-nondeterministic possible-argument-meaning (resultant-meaning)
  (either 'semanticless
          (let ((argument-meaning (subexpression resultant-meaning)))
            (unless (variable-free? argument-meaning) (fail))
            (if (equal argument-meaning resultant-meaning) (fail))
            argument-meaning)))
```

The function `subexpression` nondeterministically returns some subexpression of an `expression`. The function `possible-argument-meaning` implements half of an inverse form of the linking rule (principle 6b). It returns possible `argument-meaning`s that can link with an appropriate template meaning to yield the `resultant-meaning`. Such an `argument-meaning` can either be $\perp$ (denoted as `semanticless`), or some subexpression of the `resultant-meaning`. Principle 8 requires that argument meanings be variable-free so `argument-meaning`s that are not are filtered out. Furthermore, since by principle 9, a word cannot have a meaning which is just a variable, no node can have a meaning which is just a variable. If the `argument-meaning` were to be the same as the `resultant-meaning`, then the template meaning would have to be just a variable. Thus, `argument-meaning`s which are the same as the `resultant-meaning` are filtered out.

```
(defun-nondeterministic variable-substitute (subexpression expression variable)
  (cond ((equal expression subexpression) (either variable expression))
        ((consp expression)
```

```
         (cons (variable-substitute subexpression (car expression) variable)
               (variable-substitute subexpression (cdr expression) variable)))
        (t expression)))

(defun-nondeterministic possible-template-meaning (argument-meaning resultant-meaning)
  (if (eq argument-meaning 'semanticless)
      resultant-meaning
      (let ((template-meaning
             (variable-substitute
              argument-meaning
              resultant-meaning
              (make-variable (1+ (highest-variable resultant-meaning))))))
        (if (equal template-meaning resultant-meaning) (fail))
        template-meaning)))
```

The function `variable-substitute` takes an `expression` and returns a similar expression where subexpressions of that expression which are equal to `subexpression` are nondeterministically either replaced or not replaced by a `variable`. The function `possible-template-meaning` implements the other half of an inverse form of the linking rule (principle 6b). It returns possible `template-meaning`s that can link with a given `argument-meaning` to yield the `resultant-meaning`. If the `argument-meaning` is $\perp$ then the `template-meaning` is the same as the `resultant-meaning`. Otherwise, we nondeterministically substitute a new variable for occurrences of the `argument-meaning` within the `resultant-meaning`. Note that since the linking rule requires that the argument meaning be substituted for *some* variable in the template meaning, when doing the nondeterministic inverse substitution of a variable for occurrences of the `argument-meaning` in the `resultant-meaning`, we must guarantee that at least one such substitution has occurred. We must filter out a `template-meaning` that is equal to the `resultant-meaning` since a substitution has not occurred.

## 4 An English Example

Consider a scenario where the learner observes John rolling from a location near Mary to a location near Bill while hearing the utterance *John rolled.* The learner might hypothesize at least the following six potential meanings for that utterance since each of the following six events are subevents of the main event observed.

- John was near Mary (at the beginning of the main event).
- John was near Bill (at the end of the main event).
- John moved along some unspecified path.
- John moved along a path starting from a location near Mary.
- John moved along a path to a location near Bill.
- John moved along a path from a location near Mary to a location near Bill.

We presented DAVRA with a training session consisting of the nine utterances given in Figure 1. Each of the nine utterances was paired with between three and six possible meanings similar to those discussed above. These possible meanings were represented as Jackendovian conceptual structures.

Prior to the training session, DAVRA was not given any linguistic information other than the principles covered in Section 2. In particular, DAVRA was not given the $\overline{\mathrm{X}}$ parameter settings for English, nor was DAVRA given the syntactic category or meaning of any of the words appearing in the training session. From this training session alone, DAVRA produces the following lexicon as output:

| |
|---|
| $\mathrm{BE}(\mathbf{person}_1, \mathrm{AT}(\mathbf{person}_3)) \vee \mathrm{BE}(\mathbf{person}_1, \mathrm{AT}(\mathbf{person}_2))\vee$<br>$\mathrm{GO}(\mathbf{person}_1, [_{\mathrm{Path}}\ ]) \vee \mathrm{GO}(\mathbf{person}_1, \mathrm{FROM}(\mathbf{person}_3))\vee$<br>$\mathrm{GO}(\mathbf{person}_1, \mathrm{TO}(\mathbf{person}_2)) \vee \mathrm{GO}(\mathbf{person}_1, [_{\mathrm{Path}}\ \mathrm{FROM}(\mathbf{person}_3), \mathrm{TO}(\mathbf{person}_2)])$<br>*John rolled.* |
| $\mathrm{BE}(\mathbf{person}_2, \mathrm{AT}(\mathbf{person}_3)) \vee \mathrm{BE}(\mathbf{person}_2, \mathrm{AT}(\mathbf{person}_1))\vee$<br>$\mathrm{GO}(\mathbf{person}_2, [_{\mathrm{Path}}\ ]) \vee \mathrm{GO}(\mathbf{person}_2, \mathrm{FROM}(\mathbf{person}_3))\vee$<br>$\mathrm{GO}(\mathbf{person}_2, \mathrm{TO}(\mathbf{person}_1)) \vee \mathrm{GO}(\mathbf{person}_2, [_{\mathrm{Path}}\ \mathrm{FROM}(\mathbf{person}_3), \mathrm{TO}(\mathbf{person}_1)])$<br>*Mary rolled.* |
| $\mathrm{BE}(\mathbf{person}_3, \mathrm{AT}(\mathbf{person}_1)) \vee \mathrm{BE}(\mathbf{person}_3, \mathrm{AT}(\mathbf{person}_2))\vee$<br>$\mathrm{GO}(\mathbf{person}_3, [_{\mathrm{Path}}\ ]) \vee \mathrm{GO}(\mathbf{person}_3, \mathrm{FROM}(\mathbf{person}_1))\vee$<br>$\mathrm{GO}(\mathbf{person}_3, \mathrm{TO}(\mathbf{person}_2)) \vee \mathrm{GO}(\mathbf{person}_3, [_{\mathrm{Path}}\ \mathrm{FROM}(\mathbf{person}_1), \mathrm{TO}(\mathbf{person}_2)])$<br>*Bill rolled.* |
| $\mathrm{BE}(\mathbf{object}_1, \mathrm{AT}(\mathbf{person}_1)) \vee \mathrm{BE}(\mathbf{object}_1, \mathrm{AT}(\mathbf{person}_2))\vee$<br>$\mathrm{GO}(\mathbf{object}_1, [_{\mathrm{Path}}\ ]) \vee \mathrm{GO}(\mathbf{object}_1, \mathrm{FROM}(\mathbf{person}_1))\vee$<br>$\mathrm{GO}(\mathbf{object}_1, \mathrm{TO}(\mathbf{person}_2)) \vee \mathrm{GO}(\mathbf{object}_1, [_{\mathrm{Path}}\ \mathrm{FROM}(\mathbf{person}_1), \mathrm{TO}(\mathbf{person}_2)])$<br>*The cup rolled.* |
| $\mathrm{BE}(\mathbf{person}_3, \mathrm{AT}(\mathbf{person}_1)) \vee \mathrm{BE}(\mathbf{person}_3, \mathrm{AT}(\mathbf{person}_2))\vee$<br>$\mathrm{GO}(\mathbf{person}_3, [_{\mathrm{Path}}\ ]) \vee \mathrm{GO}(\mathbf{person}_3, \mathrm{FROM}(\mathbf{person}_1))\vee$<br>$\mathrm{GO}(\mathbf{person}_3, \mathrm{TO}(\mathbf{person}_2)) \vee \mathrm{GO}(\mathbf{person}_3, [_{\mathrm{Path}}\ \mathrm{FROM}(\mathbf{person}_1), \mathrm{TO}(\mathbf{person}_2)])$<br>*Bill ran to Mary.* |
| $\mathrm{BE}(\mathbf{person}_3, \mathrm{AT}(\mathbf{person}_1)) \vee \mathrm{BE}(\mathbf{person}_3, \mathrm{AT}(\mathbf{person}_2))\vee$<br>$\mathrm{GO}(\mathbf{person}_3, [_{\mathrm{Path}}\ ]) \vee \mathrm{GO}(\mathbf{person}_3, \mathrm{FROM}(\mathbf{person}_1))\vee$<br>$\mathrm{GO}(\mathbf{person}_3, \mathrm{TO}(\mathbf{person}_2)) \vee \mathrm{GO}(\mathbf{person}_3, [_{\mathrm{Path}}\ \mathrm{FROM}(\mathbf{person}_1), \mathrm{TO}(\mathbf{person}_2)])$<br>*Bill ran from John.* |
| $\mathrm{BE}(\mathbf{person}_3, \mathrm{AT}(\mathbf{person}_1)) \vee \mathrm{BE}(\mathbf{person}_3, \mathrm{AT}(\mathbf{object}_1))\vee$<br>$\mathrm{GO}(\mathbf{person}_3, [_{\mathrm{Path}}\ ]) \vee \mathrm{GO}(\mathbf{person}_3, \mathrm{FROM}(\mathbf{person}_1))\vee$<br>$\mathrm{GO}(\mathbf{person}_3, \mathrm{TO}(\mathbf{object}_1)) \vee \mathrm{GO}(\mathbf{person}_3, [_{\mathrm{Path}}\ \mathrm{FROM}(\mathbf{person}_1), \mathrm{TO}(\mathbf{object}_1)])$<br>*Bill ran to the cup.* |
| $\mathrm{BE}(\mathbf{object}_1, \mathrm{AT}(\mathbf{person}_1)) \vee \mathrm{BE}(\mathbf{object}_1, \mathrm{AT}(\mathbf{person}_2))\vee$<br>$\mathrm{GO}(\mathbf{object}_1, [_{\mathrm{Path}}\ ]) \vee \mathrm{GO}(\mathbf{object}_1, \mathrm{FROM}(\mathbf{person}_1))\vee$<br>$\mathrm{GO}(\mathbf{object}_1, \mathrm{TO}(\mathbf{person}_2)) \vee \mathrm{GO}(\mathbf{object}_1, [_{\mathrm{Path}}\ \mathrm{FROM}(\mathbf{person}_1), \mathrm{TO}(\mathbf{person}_2)])$<br>*The cup slid from John to Mary.* |
| $\mathrm{ORIENT}(\mathbf{person}_1, \mathrm{TO}(\mathbf{person}_2))\vee$<br>$\mathrm{ORIENT}(\mathbf{person}_2, \mathrm{TO}(\mathbf{person}_3))\vee$<br>$\mathrm{ORIENT}(\mathbf{person}_3, \mathrm{TO}(\mathbf{person}_1))$<br>*John faced Mary.* |

Figure 1: An English training session presented to DAVRA

| Head Initial, SPEC Initial. | | |
|---|---|---|
| *John*: | [N] | $\mathbf{person}_1$ |
| *Mary*: | [N] | $\mathbf{person}_2$ |
| *Bill*: | [N] | $\mathbf{person}_3$ |
| *cup*: | [N] | $\mathbf{object}_1$ |
| *the*: | $[\mathrm{N}_{\mathrm{SPEC}}]$ | $\bot$ |
| *rolled*: | [V] | $\mathrm{GO}(x, [_{\mathrm{Path}}\ ])$ |
| *ran*: | [V] | $\mathrm{GO}(x, y)$ |
| *slid*: | [V] | $\mathrm{GO}(x, [_{\mathrm{Path}}\ y, z])$ |
| *faced*: | [V] | $\mathrm{ORIENT}(x, \mathrm{TO}(y))$ |
| *from*: | [N,V,P] | $\mathrm{FROM}(x)$ |
| *to*: | [N,V,P] | $\mathrm{TO}(x)$ |

Note that Davra has determined on the basis of the training session that English is both head initial and SPEC initial. Additionally, Davra has converged to a single meaning for each word in the training session, without referentially complete knowledge of the meaning of any of the training utterances. Furthermore, for all but the prepositions, Davra has determined a unique syntactic category for each word. The only uncertainty remaining after processing this session is whether *from* and *to* are nouns, verbs or prepositions. It is easy to see that Davra can never uniquely determine that an English preposition is in fact of category P since the principles incorporated into Davra allow nouns and verbs to appear anywhere prepositions can with the same semantic consequences. One must add further principles from Universal Grammar to Davra in order to allow her to distinguish prepositions. Incorporating a variant of case theory which states both that noun phrases must receive case and that nouns are not case assigners would allow Davra to determine that English prepositions could not be nouns since their complements would not receive case. Furthermore, noticing that English prepositions are never inflected would give indirect negative evidence (Lasnik, 1989) that they are not verbs. Adding such principles to Davra would remove any remaining uncertainty from the above training session.

## 5 A Japanese Example

Maimra, a predecessor of Davra discussed in Siskind (1990), is often criticized as being unrealistic due to its assumption of a fixed, built in grammar prior to lexical acquisition. Davra attempts to address this criticism by utilizing a parameterized variant of $\overline{\mathrm{X}}$ theory instead of a fixed context free grammar, and acquiring the $\overline{\mathrm{X}}$ parameter settings simultaneously with the lexicon from the same training session. To demonstrate the success of this approach, we translated the utterances of the training session from Figure 1 into Japanese, while leaving the non-linguistic input unchanged, and presented this new session to Davra. The translated utterances are given below:

*Taro ga korogashimashita.*
*Eriko ga korogashimashita.*
*Yasu ga korogashimashita.*
*Chawan ga korogashimashita.*
*Yasu ga Eriko ni hashirimashita.*
*Yasu ga Taro kara hashirimashita.*
*Yasu ga chawan ni hashirimashita.*
*Chawan ga Taro kara Eriko ni suberimashita.*
*Taro ga Eriko ni tachimukau.*

From these utterances, Davra produced the following lexicon as output:

| Head Final, SPEC Initial. | | |
|---|---|---|
| *Taro*: | [N] | $\mathbf{person}_1$ |
| *Eriko*: | [N] | $\mathbf{person}_2$ |
| *Yasu*: | [N] | $\mathbf{person}_3$ |
| *chawan*: | [N] | $\mathbf{object}_1$ |
| *ga*: | $[\mathrm{V}_{\mathrm{SPEC}}]$ | $\perp$ |
| *korogashimashita*: | [V] | $\mathrm{GO}(x, [_{\mathrm{Path}}\ ])$ |
| *hashirimashita*: | [V] | $\mathrm{GO}(x, y)$ |
| *suberimashita*: | [V] | $\mathrm{GO}(x, [_{\mathrm{Path}}\ y, z])$ |
| *tachimukau*: | [V] | $\mathrm{ORIENT}(x, y)$ |
| *kara*: | [N,V,P] | $\mathrm{FROM}(x)$ |
| *ni*: | [N,V,P] | $\mathrm{TO}(x)$ |

Again, DAVRA was able to uniquely determine the $\overline{\mathrm{X}}$ parameter settings for Japanese, as well as unique meaning and syntactic category assignments for most words in the training session. Like before, the only uncertainty which DAVRA was unable to resolve was the assignment of category P to the words *kara* and *ni*. Methods similar to those discussed previously could remove this remaining uncertainty.

# 6 Conclusion

We emphasize that we have *not* demonstrated an algorithm that converges to parameter settings and a lexicon for all possible input of the form a child might encounter. While such a result is crucial for a complete account of child language acquisition it is still beyond our current understanding. What we have done is to demonstrate, by way of a single example, how *in principle*, an algorithm can infer $\overline{\mathrm{X}}$ parameter settings and a lexicon with neither semantic or syntactic bootstrapping assumptions. We also acknowledge that the linguistic theory incorporated into DAVRA has limited syntactic and semantic coverage. Nonetheless, we believe that the techniques discussed in this paper can be can be applied to build language acquisition models using more elaborate theories of syntax and semantics as such theories are developed.

**Acknowledgments**

The author would like to thank Linda Hershenson, Michael Caine and Yasuo Kagawa for help in translating the training session from English to Japanese.

# References

[1] Paul Bloom. *Semantic Structure and Language Development.* PhD thesis, Massachusetts Institute of Technology, September 1990.

[2] Michael Brent. Semantic classification of verbs from their syntactic contexts: Automated lexicography with implications for child language acquisition. In *Proceedings of the* $12^{\mathrm{th}}$ *Annual Conference of the Cognitive Science Society*, Massachusetts Institute of Technology, Cambridge, MA, 1990.

[3] W. Neil Elliott and Kenneth N. Wexler. A principle theory of categorial acquisition. In *NELS*, 1986.

[4] Lila Gleitman. The structural sources of verb meanings. *Language Acquisition*, 1(1):3–55, 1990.

[5] Richard H. Granger, Jr. FOUL-UP a program that figures out meanings of words from context. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, pages 172–178, August 1977.

[6] Jane Grimshaw. Complement selection and the lexicon. *Linguistic Inquiry*, 10:279–326, 1979.

[7] Jane Grimshaw. Form, function, and the language acquisition device. In C. L. Baker and J. J. McCarthy, editors, *The logical problem of language acquisition.* M. I. T. Press, Cambridge, MA, 1981.

[8] Jane Grimshaw and Steven Pinker. Using syntax to deduce verb meanings. In *The Fifteenth Annual Boston University Conference on Language Development*, page 32, October 1990.

[9] Shirley Heath. *Ways with Words: Language, life and work in communities and classrooms.* Cambridge Press, 1983.

[10] Shirley Heath. The learner as cultural member. In Mabel L. Rice and R. L. Schiefelbusch, editors, *The Teachability of Language*, pages 333–350. Paul Brookes, Baltimore, 1989.

[11] Ray Jackendoff. *Semantics and Cognition.* M. I. T. Press, Cambridge, MA, 1983.

[12] Howard Lasnik. On certain substitutes for negative data. In Robert J. Matthews and William Demopoulos, editors, *Learnability and Linguistic Theory*, pages 89–105. Kluwer Academic Publishers, Norwell, MA, 1989.

[13] John Locke. *An essay concerning human understanding.* Meridian Books, Cleveland, 1964. (Original work published 1690).

[14] Steven Pinker. *Language Learnability and Language Development.* Harvard University Press, Cambridge MA, 1984.

[15] Jeffrey Mark Siskind. Acquiring core meanings of words, represented as Jackendoff-style conceptual structures, from correlated streams of linguistic and non-linguistic input. In *Proceedings of the* 28th *Annual Meeting of the Association for Computational Linguistics*, pages 143–156, University of Pittsburgh, Pittsburgh, PA, June 1990.

[16] Jeffrey Mark Siskind. Naive physics, event perception, lexical semantics and language acquisition. In *The AAAI Spring Symposium Workshop on Machine Learning of Natural Language and Ontology*, pages 165–168, March 1991.

[17] Jeffrey Mark Siskind. *Screaming Yellow Zonkers.* Massachusetts Institute of Technology, 1991. forthcomming.