Learning strategies

Daniel N. Osherson
Institut Dalle Molle D'intelligence Artificielle Perceptive

Michael Stob
Calvin University

Scott Weinstein
University of Pennsylvania

Follow this and additional works at: https://digitalcommons.calvin.edu/calvin_facultypubs

Part of the Service Learning Commons

Recommended Citation
Osherson, Daniel N.; Stob, Michael; and Weinstein, Scott, "Learning strategies" (1982). University Faculty Publications. 524.
https://digitalcommons.calvin.edu/calvin_facultypubs/524

This Article is brought to you for free and open access by the University Faculty Scholarship at Calvin Digital Commons. It has been accepted for inclusion in University Faculty Publications by an authorized administrator of Calvin Digital Commons. For more information, please contact dbm9@calvin.edu.
Learning Strategies

DANIEL N. OSHERSON,* MICHAEL STOB,†
AND SCOTT WEINSTEIN‡

*Center for Cognitive Science, Massachusetts Institute of Technology,
Cambridge, Massachusetts 02139,
†Department of Mathematics, Calvin College,
Grand Rapids, Michigan 49506
‡Department of Philosophy, University of Pennsylvania,
Philadelphia, Pennsylvania 19104

1. INTRODUCTION

1.1 Learning Strategies and Natural Languages

A language is called natural just in case it can be acquired by normal human infants in the casual fashion of first language learning.¹ Since not every language is natural, it makes sense to seek a nontrivial property that distinguishes the natural from the nonnatural. One interesting suggestion is that children master a language by selecting a grammar for it from an innately determined class of candidates; the natural languages are those for which a grammar is included among these candidates, nonnatural languages enjoy no such candidate. An adequate theory of natural language, on this view, would characterize the natural languages by characterizing the class of grammars innately available to the child.

This equation of the natural languages with those for which a grammar is innately available requires a nontrivial assumption, namely, that children can determine which grammar of those available fits the incoming linguistic data. For, to master a language it is not sufficient to be able to devise a grammar for it; it is also necessary to recognize the adequacy of such a grammar, and the inadequacy of competing grammars. Indeed, even a creature of quite limited intelligence can generate all possible grammars (in the form of Turing machines) by employing simple enumeration techniques; selecting an appropriate grammar in response to samples from an arbitrary r.e. set is quite another matter. Should the assumption prove false—should children be

¹ The second author was partially supported by NSF Grants MCS 80-02937 and 82-00032.

² That is, considering only languages of expressive power roughly comparable to, say, English. One word languages are learnable but not natural on grounds of inexpressiveness. We leave the expressiveness proviso tacit.
unable to deploy their available grammars to maximum effect— then the natural languages may be only a proper subset of the languages determined by the innate stock of grammars.

It is the learning strategy to which the child conforms that determines the subset of innate grammars that can be successfully paired with incoming languages. As an example of a learning strategy, suppose that children are not able to store past data, and therefore revise conjectured grammars only on the basis of recent inputs; the result may be that children cannot learn every language for which they can generate hypotheses (cf., Section 3.3). Again, suppose that children never abandon a grammar that is consistent with all the data received so far; it is possible that the learnable class of languages is restricted thereby (cf., Section 3.5). It is evident that a given learning strategy is restrictive only in the context of a given set of candidate grammars. Consider two creatures, one equipped with a given learning strategy \( S \) and an innate stock \( G \) of grammars, the other equipped with \( S \) and an extension \( G' \) of \( G \). The first creature might be able to acquire every language projected by \( G \) whereas the second might be restricted to a proper subset of the languages projected by \( G \).

The “optimality” of such learning strategies is not at issue; the strategies might represent compromises in the language faculty that arise from more global design features of the nervous system. Note, too, that the child’s strategies may so restrict the possible natural languages that acquisition, being narrowly channeled, can be speedy and efficient.

Little information is currently available about the learning strategies employed by children in the course of language acquisition. More fundamentally, there is no taxonomy of learning strategies that reveals which strategies are restrictive relative to given sets of candidate grammars. To aid in developing such a taxonomy the present paper characterizes several kinds of learning strategies, and considers their restrictiveness in very general terms. For this purpose it will be necessary to give precise formulations of some of the notions figuring informally in this introductory discussion; Section 2 is devoted to such matters. Section 3 offers a parade of different kinds of learning strategies, taken singly and in combination with each other. Before turning to the formal definitions of Section 2, it will be useful to place our study in the context of related work on language learnability.

1.2 Learning Strategies and Models of Language Acquisition

Let us call a language child-learnable if it can be acquired by normal human infants on the basis of the usual sort of linguistic input provided to the young. A condition of adequacy on theories of natural language is that they specify a child-learnable collection of languages. By a model of language acquisition is meant a specification of (a) the kind of linguistic input available to human infants in the course of normal language
acquisition, (b) the internal calculations performed on this input by the child, and (c) a criterion of successful language acquisition. Plainly, the more explicit and faithful our model of language acquisition, the more useful the associated condition of child-learnability.

Seminal papers by Gold (1967) and by Wexler and his associates (see Wexler and Culicover, 1980, and citations therein) have provided a plausible and explicit model of language acquisition. The associated child-learnability condition has already figured in the evaluation of contemporary theories of natural language (see Wexler and Culicover, 1980; Pinker, 1979). The basic model embodies several critical assumptions corresponding to (a)–(c). Regarding (a), essential linguistic input is thought to be restricted to sentences of the target language; ill-formed strings, so designated, are thought to be rarely and inessentially available to the child. All (but only) the sentences of the target language are potentially available to the learner; these sentences are conceived as an infinite list, each sentence occupying one or more positions some finite distance from the start of the list. Regarding (b), the learner is construed as examining, piecemeal, ever longer initial segments of the input list of sentences; from time to time she conjectures a grammar for the language on the list. The function that maps initial segments of a list into the learner's conjectures is assumed to be computable. Regarding (c), should there come a moment after which the learner always conjectures the same grammar, and that grammar is a correct grammar for the language listed, then she is said to have “converged” on the given list to the language listed. If, no matter which way a given language is listed, the learner will ultimately converge on the list to that language, then we count the learner as being able to acquire that language. It is thus assumed that the order in which sentences are presented to the child may affect the speed of acquisition but not its ultimate success. The assumptions associated with (a)–(c) are discussed by Wexler and Culicover (1980, Chaps. 1 and 2).

Section 3 of the present inquiry focusses on part (b) of the language acquisition model. We investigate potential strategies that may govern children’s successive conjectures about the input language. For purposes of Section 3, parts (a) and (c) of the model are assumed to be adequate.

---

2 See Wexler and Hamburger (1973) for a general discussion of such models of language acquisition.

3 In the work of Wexler and Culicover (1980), not sentences but fragments of sentential derivations are assumed to be available to the learner. But the crucial assumption is preserved, namely, that negative information about grammaticality is not presented explicitly.

4 For discussion of alternative construals of “environment” for learning see Osherson, Stob, and Weinstein (1983b).

5 For an extensive discussion of alternative criteria of success in language acquisition see Osherson and Weinstein (1982a).
2. PRELIMINARIES

In this section, basic definitions, construals, and notation are set forth.

2.1 Exact Identifiability

Languages are here construed extensionally, simply as sets of sentences. Sentences may be taken to be ordered collections of phrase markers, or as phrase markers paired with a representation of meaning. So long as each sentence is a finite object, and the collection of all possible sentences can be coded by Goedel numbers, the exact nature of sentences is not relevant; indeed, for maximum generality, we shall construe languages simply as subsets of the set \( N \) of natural numbers. In the same spirit, grammars will be taken to be Turing machine indices; so, grammars are also natural numbers, appropriately interpreted. A Turing machine index is said to be for a language if the Turing machine with that index accepts that language. A language for which there is such an index is called recursively enumerable.\(^6\) By an elementary result of automata theory, there are an infinity of distinct (but equivalent) indices for each recursively enumerable language. Henceforth, we restrict attention to recursively enumerable languages.

A total function, \( t: N \to N \), is called a text. A text onto a language \( L \) is called a text for \( L \). A function \( \sigma \) from some finite ordinal into \( N \) is called a (finite) sequence. SEQ denotes the set of all finite sequences. A sequence that constitutes an initial segment (in the obvious sense) of a text \( t \) is said to be in \( t \). The length of a sequence \( \sigma \) is denoted by \( \text{lh}(\sigma) \).

A learning function \( \varphi \) is any partial function from SEQ into \( N \). Henceforth we consider only computable learning functions, and we often speak of the learning machine \( M \) in place of the learning function computed by \( M \). Given a learning machine \( M \) a text \( t \) and an index \( n \) we say that \( M \) converges to \( n \) on \( t \) just in case (a) \( M \) is defined on every sequence in \( t \), and (b) there exists an \( m \) such that for all sequences \( \sigma \) in \( t \), if \( m < \text{lh}(\sigma) \), then \( M(\sigma) = n \). We say that \( M \) converges on \( t \) just in case there is an \( n \) such that \( M \) converges to \( n \) on \( t \) and we say that \( M \) converges to \( L \) on \( t \) just in case there is an \( n \) such that \( M \) converges to \( n \) on \( t \) and \( n \) is a Turing machine index for \( L \). \( M \) is said to identify \( L \) just in case \( M \) converges to \( L \) on every text for \( L \).

Let \( L \) be a class of languages. \( M \) is said to identify \( L \) just in case \( M \) identifies every language in \( L \). \( M \) is said to exactly identify \( L \) just in case \( M \) identifies \( L \), and \( M \) identifies no proper superset of \( L \). We let \( L(M) \) denote the class of languages exactly identified by \( M \). A class \( L \) of languages is said to be identifiable just in case there is a Turing machine \( M \) such that \( L \subseteq L(M) \); \( L \) is exactly identifiable just in case \( L = L(M) \) for some \( M \).

We make several observations about these developments.

---

\(^6\) For notions of recursion theory introduced here and elsewhere see Rogers (1967).
(1) To grasp the definition of exact identifiability it helps to bear the following intuitive picture in mind. The learner is to be conceived as implementing some learning function \( f \). A text for a language \( L \) is fed into the learner, one cell at a time (each cell holds one member of \( L \)). With each new cell, the learner is faced with a new finite sequence of numbers. If \( f \) is defined on a given sequence, the corresponding value is put out by the learner; otherwise, the learner is “blocked,” and no more text is put in. The learner converges to \( L \) on the input text just in case (a) blocking never occurs on that text, and (b) after some finite number of cells, the learner always puts out the same index for \( L \).

(2) There are several ways that a machine \( M \) can fail to converge to a language \( L \) on a text \( t \) for \( L \): (i) \( M \) could fail to be defined on some sequence in \( t \); (ii) \( M \) might yield an infinite number of distinct indices; (iii) \( M \) might perpetually alternate among a finite set of distinct indices; or (iv) for some \( n \), \( M \) might yield the same index \( m \) for all sequences in \( t \) with length greater than \( n \), but \( m \) not be an index for \( L \).

(3) Each text for a language \( L \) represents a possible order in which the sentences of \( L \) might arise in the experience of the (idealized) learner. Only the sentences of \( L \) appear in these potential linguistic environments; information about nonsentences is not directly provided. (See Wexler and Culicover, 1980, Sect. 2.7 for discussion.)

2.2 Strategies and Restrictions

A (learning) strategy is a subset of learning machines. A learning strategy (i.e., subset of machines) \( S \) is called a restriction on exact identifiability just in case there is an exactly identifiable class \( L \) of languages such that for all \( M \in S \), \( L \neq L(M) \).

Strategies can be partially ordered as follows: With each strategy \( S \) is associated a family,

\[
F_S = \{ L : \text{for some machine, } M \in S, \ L(M) = L \}.
\]

We say that strategy \( S \) is more restrictive for exact identifiability than strategy \( S' \) just in case \( F_S \subseteq F_{S'} \).

Substitution of “identifiability” for “exact identifiability” in the above yields definitions of a restriction on identifiability and more restrictive for identifiability than.

When the context permits, we often speak simply of “restrictions,” without qualification.
2.3 Notation

We abide by the following conventions. The Turing machine with index $i$ is denoted $M_i$. $M_i(x_0)$ signifies that $M_i$ is defined on $x_0$. $W_i$ is the language accepted by $M_i$: $K = \{ i : M_i(i) \downarrow \}$.

A collection of languages $L$ is r.e. indexable if and only if there is an r.e. set $X$ such that $L = \{ W_i : i \in X \}$.

Given a text $t$, $\hat{t}(n)$ denotes the finite sequence $\langle 0, t(0) \rangle, \ldots, \langle n, t(n) \rangle$ of length $n + 1$. Similarly, given a sequence $\sigma$ of length $m$, for $n < m$, $\hat{\sigma}(n)$ denotes the sequence $\langle 0, \sigma(0) \rangle, \ldots, \langle n, \sigma(n) \rangle$ of length $n + 1$. $\sigma^-$ denotes the sequence resulting from omitting the last element of $\sigma$.

For simplicity, we often denote a sequence $\langle 0, t(0) \rangle, \langle 1, t(1) \rangle, \ldots$ by $\langle t(0), t(1), \ldots \rangle$; similarly for finite sequences. $p_j$ is the $j$th prime number.

3. Strategies, Restrictive and Nonrestrictive

Strategies are restraints on how a learner may respond to a sample of a language $L$. To see how a learner might be restrained, consider the "freedom" available to an arbitrary learning machine $M$. If $M$ learns $L$, $M$ must eventually, on each presentation of $L$, produce a single index for $L$. But there is no restriction on which index for $L$ may be used; nor is $M$ required to begin its convergence at any particular time. Again, $M$'s incorrect conjectures need not have any particular relation to the data that prompts them, nor to earlier conjectures. Finally, if $M$ does not learn $L$, then no particular behavior of $M$ is required on any presentation of $L$; $M$ need not even make a single conjecture.

In this section we consider the effects of limiting a learner's freedom in various ways. Six kinds of constraints are examined. These pertain to (1) the response of a machine $M$ to languages it can not identify, (2) assumptions $M$ makes about the character of languages it will encounter (3) the manner in which $M$ employs past information in responding to current input, (4) the speed with which $M$ generates conjectures, (5) the conditions under which $M$ abandons old conjectures, and (6) the conditions under which $M$ adopts new conjectures. One or more specific strategies are discussed under each rubric. We ask whether the strategy is restrictive, or whether it acts to further restrict some other strategy. The psychological reality of various strategies is also considered.
3.1 Responding to Languages That Are Not Identified

Under our definition of learning, if $M$ identifies $L$, then $M(\sigma)$ is defined whenever $\text{range}(\sigma) \subseteq L$. But suppose $M$ does not identify $L$. Various behaviors on texts for $L$ are possible. $M$ may, for instance, not be defined on some $\sigma$ such that $\text{range}(\sigma) \subseteq L$. For example, $M$ may "realize" that $\sigma(0)$ is not an element of any language that $M$ can learn. At the other extreme, $M$ may converge on some text for $L$, or even converge on every text for $L$, albeit not always to grammars for $L$. The present subsection examines four means of constraining the relationship of $M$ to languages it can not identify. The corresponding strategies are called "totality," "reliability," "confidence," and "prudence."

Totality. We first consider machines that respond to every input.

**Definition.** A machine $M$ is called **total** just in case $M$ computes a total function, i.e., $M(\sigma) \downarrow$ for every $\sigma$.

**Proposition 3.1A.** For every machine $M$ there is a total machine $M'$ such that $L(M) \subseteq L(M')$.

**Proof.** If $\sigma$ is a sequence, define $\sigma_s$ to be the longest initial segment of $\sigma$ such that $M_s(\sigma_s) \downarrow$ if there is such. Define

$$M'(\sigma) = M_{\text{th}(\sigma)}(\sigma_{\text{th}(\sigma)}) \text{, if } \sigma_{\text{th}(\sigma)} \text{ exists,}$$

$$= 0, \text{ otherwise.}$$

Evidently, $M'$ is total and $L(M) \subseteq L(M')$.

Informally, $M'$ does not wait forever for $M(\sigma)$ to be defined, but settles for the guess $M(\sigma')$, for some initial segment $\sigma' \subseteq \sigma$. But for longer strings $\sigma$, $M'$ waits longer for $M$ to be defined. 

Thus, totality is not a restriction on the classes of languages that can be identified. It is, however, a restriction on the classes of languages that can be exactly identified.

**Proposition 3.1B.** There is a class of languages $L$ such that $L = L(M)$ for some machine $M$ but $L \neq L(M')$ for any total machine $M'$.

**Proof.** The class $L$ consists of those languages $L_i = \{i\}$ such that if $t_i$ is the text given by $t_i(n) = i$ for all $n$, then $M_i(t_i(n))$ is defined for all $n$, but $M_i$ does not identify $L_i$ on $t_i$.

First we show that no total machine $M'$ can exactly identify $L$. Suppose that $M_i$ is total and exactly identifies $L$. If $L_i \subseteq L$, then $M_i$ does not identify $L_i$. Thus $L_i \notin L$. However, $M_i(t_i(j))$ is defined for every $j$; thus, since $L_i \subseteq L$, $M_i$ must identify $L_i$ on $t_i$. But this contradicts $L_i \notin L$. 

We now construct an \( M \) such that \( L = L(M) \). Obviously we need only describe how \( M \) behaves on the texts \( t_i \). \( M(\bar{t}_i(n)) = \) an index for \( L_i \) if \( M(\bar{t}_i(n)) \) is defined. \( M(\bar{t}_i(n)) \) is defined if and only if there is an \( m > n \) such that \( M_i(\bar{t}_i(m)) \) is defined and \( M_i(\bar{t}_i(m)) \neq M_i(\bar{t}_i(n)) \) or \( W_{M_i(\bar{t}_i(m))} = \{i\} \). Thus, \( M \) converges to \( L_i \) on \( t_i \) if and only if \( M_i \) is total and \( M_i \) changes its conjecture infinitely often or converges to a wrong conjecture.

We suspect that totality is a feature of the human language learner. It seems unlikely that there is a possible speech input that would send the infant into an endless computation, depriving her forever of language; eventually, she must terminate her deliberations, if only to default to some empty or otherwise primitive grammar.

**Reliability.** If \( M \) occasionally converges to an incorrect language, then \( M \) may be termed “unreliable.”

**Definition** (Minicozzi, cited in Blum and Blum, 1975, Sect. 5). A machine \( M \) is **reliable** just in case for all languages \( L \) and texts \( t: N \rightarrow L \), if \( M \) converges on \( t \), then \( M \) converges to \( L \).

To show that reliability is restrictive, we rely upon a useful lemma due to Blum and Blum (1975, Sect. 4).

**Lemma 3.1C.** Suppose that \( M \) identifies \( L \). Then there is a finite sequence \( \sigma \) such that \( \operatorname{range}(\sigma) \subseteq L \) and for every \( \tau \supseteq \sigma \), if \( \operatorname{range}(\tau) \subseteq L \), then \( M(\tau) \) is an index for \( L \), and \( M(\tau) = M(\sigma) \).

**Proposition 3.1D.** Suppose that \( M \) is reliable. Then if \( L \in L(M) \), \( L \) is finite.

**Proof.** Suppose that \( L \in L(M) \), but \( L \) is infinite. By Lemma 3.1C, let \( \sigma \) be such that \( \operatorname{range}(\sigma) \subseteq L \), and if \( \tau \supseteq \sigma \), and \( \operatorname{range}(\tau) \subseteq L \), then \( M(\tau) \) is an index for \( L \) and \( M(\tau) = M(\sigma) \). This implies that for some text, \( t: N \rightarrow \operatorname{range}(\sigma) \), \( M \) converges to \( L \) on \( t \). But \( \operatorname{range}(\sigma) \) is finite and \( L \) is infinite, so \( M \) does not converge to \( \operatorname{range}(\sigma) \) on \( t \). Hence, \( M \) is not reliable.

Assuming (as we do) that natural languages are infinite, Proposition 3.1D shows that reliability is not a feature of human learners.

**Confidence.** Whereas the reliable machines converge on a text only if they converge to the language for which it is a text, the confident machines converge on every text.

---

7 For an extended discussion of this lemma and its connection with topological considerations in learning theory see Osherson, Stob, and Weinstein (1983a).
DEFINITION. A machine $M$ is **confident** just in case $M$ converges on $t$, for every text, $t$.

**Proposition 3.1E.** There is a collection of languages $L$ such that $L$ is exactly identifiable by a total, reliable machine $M$ but $L$ is not identifiable by any confident machine $M'$.

**Proof.** Let $L$ be the set of finite languages and define $M(\sigma)$ to be an index for $\text{range}(\sigma)$. Then $M$ is a total, reliable machine with $L(M) = L$.

Suppose for reductio that $M'$ is a confident machine with $L \subseteq L(M')$, and let $t$ be a text for some infinite language. Let $\sigma^n$ be $\sigma$ concatenated with itself $n$ times. Define a sequence of finite sequences as follows: Let $\sigma_0 = \langle t(0) \rangle$. Let $\sigma_{i+1} = \sigma_i \cdot \sigma^n$, where $\sigma$ is the shortest initial segment of $t$ such that $\text{range}(\sigma_i \cdot \sigma) \neq W_{M'(\sigma_i)}$ and $n$ is the least $m$ such that $W_{M'(\sigma_i \cdot \sigma^m)} = \text{range}(\sigma_i \cdot \sigma)$. Let $t' = \bigcup \{\sigma_i : i \in \mathbb{N}\}$. Then $t'$ is a text on which $M'$ does not converge, which contradicts the hypothesis that $M'$ is confident. \]

**Prudence.** A final restriction on how a machine may respond to languages it cannot identify is a constraint on what the machine may respond if it chooses to respond at all.

DEFINITION. A machine $M$ is **prudent** just in case $M(\sigma) \downarrow$ implies that $M$ identifies $W_{M(\sigma)}$.

Before stating our results about prudent machines it will be necessary to introduce the notion of an order-independent machine and a lemma concerning the existence of such machines.

**Definition (Blum and Blum, 1975, Sect. 4).** A machine $M$ is called order-independent just in case for every $L \in L(M)$ there is an index $j$ for $L$ such that for every text $t : \mathbb{N} \to L$ for $L$ there is an $n$ such that $M(i(n + i)) = j$ for every $i$.

**Lemma 3.1F (Blum and Blum, 1975).** For every machine $M$ there is an order-independent machine $M'$ such that $L(M) \subseteq L(M')$.

**Proof.** While our notion of identification is different from that of Blum and Blum (1975), the proof they give works here with minor modifications. \]

**Proposition 3.1G.** $L$ is exactly identifiable by a prudent, order-independent machine $M$ if and only if $L$ is identifiable and r.e. indexable.

**Proof.** One direction is straightforward, since a prudent machine iden-
tifies all and only languages for which it conjectures an index and the set of indices conjectured by any machine is r.e.

In the other direction, suppose that $L$ is identifiable by some machine $M'$ and $X$ is an r.e. set of indices for $L$. By Lemma 3.1F we may, without loss of generality, suppose that $M'$ is order-independent. Let $f$ be a total recursive function which enumerates $X$. We construct an order-independent, prudent machine $M$ such that $L(M) = L$ as follows:

If $lh(\sigma) = 1$, let $M(\sigma) = f(0)$; if $lh(\sigma) > 1$, then $M$ simulates $M'$ on input $\sigma$. If $M'$ diverges on $\sigma$, then $M$ does also. If $M'(\sigma) = y$, then $M$ uniformly constructs recursive enumerations $g_0, \ldots, g_{lh(\sigma)}$ of $W_{f(0)}, \ldots, W_{f(lh(\sigma))}$ and searches for the least $i$ such that $M'(\bar{g}_i(lh(\sigma))) = y$. If such an $i$ is found, let $M(\sigma) = f(i)$; if not, $M(\sigma) = M(\sigma^-)$.

Since $M'$ is order-independent and identifies $L$, $M$ is also order-independent and identifies $L$. But, range($M$) $\subseteq X$, hence $M$ exactly identifies $L$ and is prudent. $
$

**Corollary 3.1H.** There is a collection $L$ of languages such that $L$ is exactly identifiable, but $L$ is not exactly identifiable by any prudent machine.

**Proof.** By Proposition 3.1G it suffices to show that there is a collection of languages $L'$ such that $L'$ is exactly identifiable but $L'$ is not r.e. indexable. The collection $L'$, of singleton languages

$$\{i\}, \quad \text{for } i \in \bar{K}$$

is such a collection. $L'$ is exactly identifiable by a machine that conjectures an index for $\{i\}$ on initial input $i$, then begins an internal enumeration of $K$; if $i$ shows up in the enumeration, the machine begins an infinite alternation of indices for $\emptyset$ and $N$. And $L'$ is not an r.e. collection, since that would exhibit $\bar{K}$ as r.e. $
$

It is an open question whether or not every identifiable collection of languages can be identified by a prudent machine. By Proposition 3.1G this is equivalent to the question whether every identifiable collection of languages can be extended to an identifiable collection of languages which is r.e. indexable.

Children acquiring language may well be prudent learners, especially if "prestorage" models of language acquisition are correct. A prestorage model posits an internal list of candidate grammars that coincides exactly with the natural languages; language acquisition amounts to the selection of a grammar from this list in response to linguistic input. Such a prestorage learner is prudent inasmuch as his hypotheses are limited to grammars from the list, that is, to grammars corresponding to natural (i.e., learnable) languages.
3.2 Assumptions about Languages to be encountered

Next we consider machines which are constrained by assumptions they make about the character of languages they will encounter. The sole constraint we consider is the assumption that the languages to be encountered are infinite.

**DEFINITION.** $M$ is **nontrivial** if and only if $\forall x \ (x \in \text{range}(M) \rightarrow W_x$ is infinite).

Nontriviality is clearly a restrictive strategy, since no nontrivial machine identifies a finite language. More interesting, however, is the fact that nontriviality even restricts the collections of infinite languages that are available to the learner.

**PROPOSITION 3.2A.** There is an identifiable collection of infinite languages which is not identifiable by a nontrivial machine.

**Proof.** Let $L_i = \{p_i^{n+1} : n \in W_i\}$ and let $L = \{L_i : W_i$ is infinite}. Let $g$ be a total recursive function such that $L_i = W_{g(i)}$ and let $M(\sigma) = g(\mu i(\exists n(\sigma(lh(\sigma) - 1) = p_i^{n+1})))$ if there is such an $i$, and an index for $\emptyset$ otherwise. Then $M$ exactly identifies $\{L_i : i \in N\}$ and hence identifies $L$.\(^8\)

It remains to show that $L$ is not identifiable by a nontrivial machine. Suppose, for reductio, that $M$ is a nontrivial machine which identifies $L$, and let $X = \text{range}(M)$. Then $X$ is an r.e. set such that for every $n \in X$, $W_n$ is infinite, and $L \subseteq \{W_n : n \in X\}$. But, there is a total recursive function $h$ such that $W_{h(n)}$ is infinite if $W_n$ is infinite, and if $W_n = L_i \in L$, then $W_{h(n)} = W_i$. But then $h[X]$ is an index set for $\text{INF} (= \{L : L$ is infinite}) and this contradicts the fact that $\text{INF}$ is not r.e. indexable (which can be shown via a simple diagonalization). \(\blacksquare\)

Linguists rightly emphasize the infinite quality of natural languages; no natural language, it appears, includes a longest sentence. If this universal feature of natural language corresponds to an innate constraint on children’s linguistic hypotheses, then they would be barred from conjecturing a grammar for a finite language. To this extent child language learners may implement a nontrivial learning strategy.

3.3 Using Past Information

In this subsection, two constraints are examined that pertain to the effect of past information or a learner’s current conjecture. The corresponding strategies are called “set-drivenness” and “memory-limitation.”

\(^8\) The reader should note that $M$ is a 1-memory limited machine in the sense of Section 3.3. In addition, it can be shown that $L$ is exactly identifiable using methods which are developed in Osherson, Stob, and Weinstein (forthcoming).
Set-drivenness. Suppose that $M(\sigma)$ depends only upon the elements appearing in range(\sigma), not upon their order. Then we have the “set-driven” machines.

**Definition** (Wexler and Culicover, 1980, Sect. 2.2). A machine $M$ is set-driven if \( \text{range}(\sigma) = \text{range}(\tau) \) implies $M(\sigma) = M(\tau)$.

**Proposition 3.3A** (Blum and Blum, 1975). For any machine $M$ there is a set-driven machine $M'$ such that $L(M) \subseteq L(M')$.

**Proof.** The machine constructed in the Blums’ proof of Lemma 3.1F is set-driven. 

Whereas the last Proposition shows that set-drivenness is not restrictive for identification, we have been able to show that it is restrictive for exact identification.

**Proposition 3.3B.** There is a machine $M$ such that for no set-driven machine $M'$ is $L(M) = L(M')$.

**Proof.** Let $\mathbf{L} = \{ \{n\} : W_n \not\in \{n\} \}$. $\mathbf{L}$ can be exactly identified by a machine $M$ as follows:

$M(\sigma) = \text{an index for } \{n\} \text{ if and only if } \sigma(i) = n \text{ for all } i,$ and

$W_{n, \text{lh}(\sigma)} \not\in \{n\}.$ Otherwise, $M(\sigma)$ is chosen to be an index for $\{\text{lh}(\sigma)\}$.

To see that no set-driven machine $M'$ will exactly identify $\mathbf{L}$, let $M'$ be such a machine. Define a recursive function $g$ so that

\[
W_{g(n)} = \emptyset, \quad \text{if } M'((n)) \text{ does not exist,}
\]

\[
= W_{M'(\langle n \rangle)}, \quad \text{otherwise.}
\]

Function $g$ is a total recursive function, so there is an $n_0$ such that $W_{g(n_0)} = W_{n_0}$. Suppose that $W_{n_0} = \{n_0\}$. Then $\{n_0\} \not\in \mathbf{L}$. On the other hand, $M'$ converges to $g(n_0)$ on any text consisting entirely of $n_0$'s since $M'$ is set-driven and $W_{g(n_0)} = W_{M'(\langle n_0 \rangle)}$ (since $W_{g(n_0)}$ is nonempty). Thus, $M'$ identifies $\{n_0\}$. Now suppose that $W_{n_0} \not\in \{n_0\}$. Then $M'$ must identify $\{n_0\}$ but $W_{g(n_0)} = W_{n_0} \not\in \{n_0\}$ so that $M'$ does not identify $\{n_0\}$ since either $M'(\langle n_0 \rangle)$ is not defined or $W_{M'(\langle n_0 \rangle)} \not\in \{n_0\}$.

It seems obvious that human infants are not set-driven.

**Memory-limitation.** Another sort of restriction we might impose on how machines use past information is a memory restriction. The following definition might be viewed as a formalization of short-term memory.
DEFINITION (Wexler and Culicover, 1980, Sect. 3.2). A machine $M$ is $n$-memory-limited if whenever $\sigma$ and $\tau$ are finite sequences that agree on their last $n$ arguments (i.e., $(\forall i \leq n) \sigma(\text{lh}(\sigma) - i) = \tau(\text{lh}(\tau) - i)$), then $M(\sigma^-) = M(\tau^-)$ implies $M(\sigma) = M(\tau)$.

**Proposition 3.3C.** There is a class of languages $L$ such that $L = L(M)$ for some machine $M$ but for no $n$ is there an $n$-memory-limited machine $M'$ such that $L \subseteq L(M')$.

**Proof.** Let $L$ consist of the language $L = \{2^i : i \in N\}$, along with, for each $j \in N$, the languages

$$L_j = \{2^i : i \in N\} \cup \{3^j\}$$

and

$$L'_j = \{2^i : i \in N \text{ and } i \neq j\} \cup \{3^j\}.$$ 

It is easy to see that $L = L(M)$ for some machine $M$. But suppose that $L \subseteq L(M')$ for some $1$-memory-limited machine $M'$ (the case for $n > 1$ is similar). Intuitively, if $M'$ sees $3^j$ for some $j$, $M'$ cannot remember if it has seen $2^j$ a while ago, and so cannot distinguish between $L_j$ and $L'_j$. Then by Lemma 3.1C, there is some sequence $\sigma$ such that $\text{range}(\sigma) \subseteq L$ and whenever $\tau \supseteq \sigma$, $\text{range}(\tau) \subseteq L$, then $M'(\sigma) = M'(\tau)$. Let $\sigma' = \sigma^\prec (3^{j_0})$ for some fixed $j_0$ such that $2^{j_0} \not\in \text{range}(\sigma)$. Let $n + 1 = \text{lh}(\sigma)$, and let $\sigma'' = \sigma^\prec (2^{j_0})^\prec (3^{j_0})$. Now $\sigma'$ and $\sigma''$ agree on their last arguments. Also $M'(\sigma') = M'(\sigma'')$. Thus $M'(\sigma') = M'(\sigma'')$. But then considering the texts $t_1 = \sigma' \prec (2^0, 2^1, 2^2, \ldots, 2^i, \ldots)_{i \neq j_0}$ and $t_2 = \sigma'' \prec (2^0, 2^1, 2^2, \ldots, 2^i, \ldots)_{i \neq j_0}$, we have that $M'$ converges on $t_1$ and $t_2$ to the same index because of memory limitations. But $t_1$ is a text for $L'_j$, and $t_2$ is a text for $L_j$. \(\blacksquare\)

Memory limitations will be seen to increase the restrictiveness of several other strategies. There are alternative ways to formalize the intuitive idea of a limited memory, but it is clear that children operate under some such constraint.

### 3.4 Speed of Conjectures

We now examine the impact of bounding the time available to a learner to respond to an input. The sole strategy considered is called “fast-working.”

**Fast-working.**

**Definition.** Let $h$ be a recursive function. $M$ is an $h$-fast-worker just in case $M(\sigma) \downarrow$ if and only if $M_s(\sigma) \downarrow$ for some $s \leq h(\sigma)$.
Fast-working places a further restriction on what memory-limited machines can identify.

**Proposition 3.4A.** For every $n$, and for every recursive function $h$ there is a class $L$ of languages such that $L = L(M)$ for some $n$-memory-limited machine $M$, but $L \not\subseteq L(M')$ for any $h$-fast-working, $n$-memory-limited machine, $M'$.

**Proof.** We prove the proposition for the case of $n = 1$; the other cases are similar. Let $R$ be a recursive set whose characteristic function cannot be computed in $\lambda x \cdot h(\sigma \hat{x})$-time for any $\sigma$; i.e., for every total recursive function $\phi$ such that $\phi(x) = 1$ if and only if $x \in R$ and for every $\sigma$, there are infinitely many $x$ such that $\phi_{h(\sigma \hat{x})}(x)$ is not defined. Let $L = \{R \cap F : F$ is finite$\}$. Clearly, $L$ is identifiable. But no machine $M'$ of the sort asked for can exist. For, choose a finite string $\sigma$ such that $\text{range}(\sigma) \subseteq R$, and for every $\tau$, $\tau \supseteq \sigma$ and $\text{range}(\tau) \subseteq R$ implies $M'(\tau) = M'(\sigma)$. Then to see whether $n \in R$, compute $M'(\sigma \hat{n})$, $n \in R$ if and only if $M'(\sigma \hat{n}) = M'(\sigma)$ by memory-limitedness of $R$. But this constitutes an effective procedure for answering $n \in R$ in time $h(\sigma \hat{n})$. \[\square\]

3.5 Abandoning Conjectures

We next consider constraints on the freedom of a learner to abandon hypotheses. The sole strategy considered allows machines to give up a conjecture only if it fails to predict what has been seen so far on the input text.

**Conservatism.**

**Definition.** A machine $M$ is **conservative** just in case whenever $\sigma \subseteq \tau$ and $M(\sigma) \neq M(\tau)$, then $W_{M(\sigma)} \not\subseteq \text{range}(\tau)$.

The conservative machines were introduced by Angluin (1980); however, Angluin's definition of identification is quite different from ours. Note that a conservative machine will not abandon any conjecture $W_i$ if the text is for a language $L \subseteq W_i$. Thus, one might suppose the strategy to be a restriction. Indeed,

**Proposition 3.5A.** There is a class of languages $L$ such that $L = L(M)$ for some learner $M$, but $L \not\subseteq L(M')$ for any conservative learner $M'$.

**Proof.** The proof is similar to one given by Angluin (1980), although she worked with a different and weaker notion of learner. $L$ consists of languages $L_j$ and $L_j'$, for all $j \in N$, where $L_j = \{p_i^j : i \in N\}$
and \( L_j = \{ p_j \} \), unless there are integers \( s \) and \( n \) such that for the machine with index \( j \),

\[
M_j(f_j(n)) = i, \quad \text{where} \quad f_j(k) = p_j^k \quad \forall i, s, \exists \{ p_j^1, p_j^2, \ldots, p_j^n \} \quad \text{(*)}
\]

If such an \( s \) and \( n \) exist, choose the least such \( s \) and least \( n \) for that \( s \), and let \( L_j = \{ p_j^1, \ldots, p_j^n \} \). It is easy to see that \( L = L(M) \) for some machine, \( M \). But suppose that \( L \subseteq L(M_j) \) for some \( j \) such that \( M_j \) is conservative. Then there must be an \( n \) and \( s \) satisfying (*) since \( M_j \) identifies \( L_j \) and \( f_j \) is a text for \( L_j \).

But then for the least such \( s \) and \( n \), \( L_j = \{ p_j^1, \ldots, p_j^n \} \). On the text, \( t \), defined by

\[
t(m) = p_j^{m+1}, \quad \text{if} \quad m < n, \\
= p_j^n, \quad \text{otherwise},
\]

we must have that \( M_j(i(m)) = i \) for all \( m \geq n \) if \( M_j \) is conservative. Thus \( M_j \) does not learn \( L_j \) on text \( t \).

Informally, if \( M_j \) learns \( L_j \) it must eventually conjecture a language larger than the finite language it has seen so far. But then, if \( M_j \) is conservative, it cannot cut its conjecture back to this finite language.

In turn, set-drivenness restricts exact identifiability for the conservative machines.

**Lemma 3.5B.** If a class of languages \( \mathcal{L} \) can be exactly identified by a set-driven, conservative machine, then \( \mathcal{L} \) is r.e. indexable.

**Proof.** Let \( M \) be conservative and set-driven. Consider \( W = \{ i: (\exists \sigma) (M(\sigma) = i \text{ and } \text{range}(\sigma) \subseteq W_i) \} \). Obviously, \( W \) is r.e. We claim that \( i \in W \) if and only if \( i \in L \). Clearly, if \( L \subseteq L \), \( L = W_i \) for some \( i \in W \). Suppose \( i \in W \). Let \( \sigma \) be such that \( M(\sigma) = i \) and \( \text{range}(\sigma) \subseteq W_i \). If \( t \) is a text for \( W_i \) beginning with \( \sigma \), then \( M \) identifies \( W_i \) since \( M \) is conservative. But if \( t' \) is any text for \( W_i \), then \( \text{range}(t') \supseteq \text{range}(\sigma) \) so that for some \( n \), \( \text{range}(t'(n)) \supseteq \text{range}(\sigma) \). But then \( M(i'(n)) = M(\tau) \), where \( \text{range}(\tau) = \text{range}(t'(n)) \) and \( \tau \) begins with \( \sigma \). For such a \( \tau \), however, \( M(\tau) = M(\sigma) \) since \( M \) is conservative.

**Lemma 3.5C.** There is a class of languages \( \mathcal{L} \) which can be exactly identified by a conservative machine \( M \) which is not r.e. indexable.

**Proof.** The class of languages \( \mathcal{L} \) consists of languages \( L_i = \{ 2^i \} \), \( i \in N \) and \( L_i' = \{ 2^i, 3^i \} \) for \( i \in A \), where \( A \) is some fixed nonrecursive r.e. set. The conservative machine \( M \) works as follows. If \( M \) sees \( 3^i \) before \( 2^i \) it conjectures \( L_i \). If \( M \) sees \( 2^i \) before \( 3^i \) it conjectures \( L_i \) and begins an internal
LEARNING STRATEGIES

47

If \( M \) then sees \( 3^i \) it conjectures \( L_i' \) unless its internal simulation of \( A \) has enumerated \( i \in A \). If so, then \( M \) conjectures an index for \( N. \) \( M \) exactly identifies \( L \), for if \( i \in A \) there will be some text \( t \) for \( L_i' \), where \( 3^i \) first appears in \( t \) after \( i \) appears in \( A \). However, \( L \) is not r.e. indexable else \( A \) is r.e.

**Proposition 3.5D.** There is a class of languages \( L \) such that \( L = L(M) \) for a conservative machine \( M \), but \( L \neq L(M') \) for any conservative, set-driven machine \( M' \).

**Proof.** By the lemmas.

### 3.6 Adopting New Conjectures

Conservatism imposes conditions on which conjectures may be abandoned. In the present section we consider five conditions on which conjectures may be adopted.

**Consistency.** We first consider machines which require their new conjectures to explain all the data the machine has seen so far.

**Definition.** (a) A machine \( M \) is **consistent** just in case whenever \( M(\sigma)_\downarrow \), then \( W_{M(\sigma)} \supseteq \text{range}(\sigma) \). (b) A machine \( M \) is **weakly consistent** just in case whenever \( M(\sigma^-) \neq M(\sigma) \), then \( W_{M(\sigma)} \supseteq \text{range}(\sigma) \).

The notion of consistency is due to Angluin (1980).

When combined with totality, consistency is a severe restriction on what can be identified.

**Proposition 3.6A.** Suppose that \( L \subseteq L(M) \), where \( M \) is a consistent, total machine. Then \( L \) is recursive.

**Proof.** Let \( \sigma \) be such that \( \text{range}(\sigma) \subseteq L \), and if \( \tau \supseteq \sigma \), then \( \text{range}(\tau) \subseteq L \) implies \( M(\tau) = M(\sigma) \). Such a \( \sigma \) exists by Lemma 3.1C. Then, to test whether \( n \in L \), compute \( M(\sigma \hat{\langle} n \hat{\rangle}) \). If \( n \in L \), \( M(\sigma \hat{\langle} n \hat{\rangle}) = M(\sigma) \). If \( n \not\in L \), then \( M(\sigma \hat{\langle} n \hat{\rangle})_\downarrow \) since \( M \) is total, and \( M(\sigma \hat{\langle} n \hat{\rangle}) \neq M(\sigma) \) since \( M \) is consistent. This gives an effective test for membership in \( L \).

In light of children's limited memory, consistency would seem to be a psychologically unrealistic strategy.

**Caution.** Conservative machines do not overgeneralize on languages they do in fact identify (since once a conservative machine overgeneralizes, it is trapped in that conjecture); however, a conservative machine may well overgeneralize on a language it does not identify. We now examine machines that behave as if they never overgeneralize.
DEFINITION. A machine $M$ is cautious just in case whenever $M(\sigma) \neq M(\tau)$ and $\sigma \subseteq \tau$, then $W_{M(\sigma)}$ is not a proper subset of $W_{M(\tau)}$.

Like conservatism, caution is a restriction in a very strong sense.

PROPOSITION 3.6B. There is a class of languages $L$ such that $L = L(M)$ for some machine $M$, but $L \notin L(M')$ for any cautious machine $M'$.

Proof. The proof of Proposition 3.5A establishes this proposition as well: we simply observe that $M_j$ of that proof can be neither conservative nor cautious if it identifies the class $L$ of languages described therein.

Conservative machines are not necessarily cautious. However, their lack of caution does not allow them to identify any more.

PROPOSITION 3.6C. Suppose that $M$ is a conservative machine. Then there is a cautious, conservative machine $M'$ such that $L(M) \subseteq L(M')$.

Proof. For every sequence $\sigma$ let $\delta$ be the longest initial segment of $\sigma$ such that $M_s(\delta)$ is defined and $W_{M_s(\delta),s} \supseteq \text{range}(\delta)$ for some $s \leq \text{lh}(\sigma)$ if such a sequence exists, and $\langle \sigma(0) \rangle$ otherwise. Define $M'(\sigma) = M(\delta)$.

First notice that if $L \in L(M)$, then $L \in L(M')$, for if $t : N \to L$ is a text for $L$, then there is an $n$ such that $M(i(n)) = i$ for some index $i$ for $L$, and $(\forall m \geq n) (M(i(m))$ is an index for $L)$. Then, for large enough $m$, $i(m) \supseteq i(n)$, and thus $M'(i(m))$ is an index for $L$. $M'$ is cautious since if $\sigma \subseteq \tau$, then $\delta \subseteq \tilde{\delta}$, so that $M'(\sigma) = M(\delta)$ and $M'(\tau) = M(\tilde{\delta})$. If $M'(\sigma) \neq M'(\tau)$, then $M(\delta) \neq M(\tilde{\delta})$ so that $W_{M(\delta)} \not\supseteq \text{range}(\tilde{\delta})$. But $W_{M(\tau)} \supseteq \text{range}(\delta)$. Thus, $W_{M'(\tau)} \not\subseteq W_{M'(\sigma)}$. It is easy to see that $M'$ is conservative.

Decisiveness. A decisive machine never returns to a language once abandoned.

DEFINITION. A machine $M$ is decisive if whenever $\sigma \subseteq \tau$ and $W_{M(\sigma)} \neq W_{M(\tau)}$, then there is no $\gamma$ such that $\tau \subseteq \gamma$ and $W_{M(\sigma)} = W_{M(\gamma)}$.

Like the cautious machines, the decisive machines can identify everything identifiable by the conservative machines. We have not, however, been able to determine whether decisiveness is restrictive.

PROPOSITION 3.6D. If $L = L(M)$ for some conservative machine $M$, then $L \subseteq L(M')$ for some decisive machine $M'$.

Proof. The machine $M'$ in the proof of Proposition 3.6C is decisive.

Gradualism. We next study machines that employ a built-in “simplicity metric” in a certain way.
LEARNING STRATEGIES

49

DEFINITION. A machine $M$ is *gradualist* just in case there is a one-one recursive function, $f: N \rightarrow N$, and an integer $n$, such that if $M(\sigma^{-}) \neq M(\sigma)$, then $|f(M(\sigma)) - f(M(\sigma^{-}))| \leq n$; we call the least such $n$ the *modulus* of $M$.

Intuitively, a gradualist machine has a measure $f$, of complexity of grammars, and it changes conjectures only to grammars of nearby complexity. Gradualism is not restrictive if the machine is not memory limited; otherwise, gradualism imposes a constraint over and above memory limitation on what can be identified.

**PROPOSITION 3.6E.** If $M$ is any machine, then there is a gradualist machine $M'$ such that $L(M) = L(M')$.

*Proof* (Informal). $M'$ simulates $M$. If $M$ wants $M'$ to change a conjecture $m$ to conjecture $k$, $M'$ does this by using the next $|f(m) - f(k)|$ inputs to effect this change one level of the $f$ hierarchy at a time. Memory is used to store these latter inputs, which are subsequently fed to $M$ in the continuation of the simulation.

**PROPOSITION 3.6F.** There is a class of languages $L$ such that $L = L(M)$ for some memory-limited machine $M$, but $L \not\subseteq L(M')$ for any gradualist, memory-limited machine $M'$.

*Proof.* Let $L = \{\{1, m\} : m \in N\}$. Obviously, a 1-memory-limited machine $M$ exists which exactly identifies $L$. Suppose $M'$ is gradualist. Consider the texts

$$t_n(i) = 1, \quad \text{if } i \neq 1,$$

$$= n, \quad \text{if } i = 1.$$

Then $M'$ on $t_n$ must converge to an index for $\{1, n\}$ if $M'$ identifies $L$. Since $M'(t_n(0)) = M'(t_m(0))$ for all $n$ and $m$, and since $M'$ is gradualist, there are infinitely many pairs of integers $n$ and $m$ such that $n \neq m$ and $M'(i_n(1)) = M'(i_m(1))$. But then, since $M'$ is memory-limited, $M'$ converges on infinitely many of the $t_n$ to the same index. Thus $M'$ cannot identify $L$. (In fact, if $M'$ is $k$-memory-limited and $j$ is the modulus of $M'$, then $M'$ converges correctly on at most $(2j)^{k+1}$ of the texts $t_n$.)

**Induction by enumeration.** One strategy for generating conjectures is to choose the first grammar in some effective list of grammars that is consistent with the data seen so far.

**DEFINITION.** A machine, $M$, uses *induction by enumeration* just in case there is a recursive function, $f$, such that $M(\sigma) = f(i)$, where $i$ is the least number such that $\text{range}(\sigma) \subseteq W_{f(i)}$. 

Here \( \{M_{f(i)} : i \in N\} \) is the effective list of grammars. Notice that the condition on \( M(\sigma) \) given by the definition cannot be taken as a definition of \( M \); for, a function meeting that condition need not be recursive. However, if \( \{W_{f(i)} : i \in N\} \) is in fact a uniformly recursive sequence of recursive sets, then a function meeting the condition above is recursive, and, indeed, \( M \) is defined thereby. Even in these special circumstances, however, induction by enumeration is restrictive.

**PROPOSITION 3.6G.** There is a uniformly recursive class \( L \) of languages such that \( L = L(M) \) for some machine \( M \), but \( L \) cannot be identified by any machine \( M' \) that uses induction by enumeration.

**Proof.** Let \( L_n = \{x : x \geq n\} \), and let \( L = \{L_n : n \in N\} \). Obviously \( M \) exists. If \( M' \) exists, let \( f \) be the supposed ordering function. Certainly there are natural numbers \( i < j \) such that \( W_{f(i)} \not= W_{f(j)} \). But then \( M \) presented with a text for \( W_{f(j)} \) converges to \( W_{f(i)} \).

4. Conclusion

For a class of languages to be the natural languages, the class must be learnable by children. Formal learning theory is an attempt to deploy this fundamental fact in the evaluation of theories of natural language. For such a purpose, the notion “learnable by children” must be rendered precisely. The concept of “exact identifiability” defined in Section 2.1 is a step toward the needed formalization. However, children do not implement arbitrary learning functions, but rather a special kind whose characteristics derive from human developmental psychology. To the extent that we impose those characteristics on the learning functions under consideration, exact identifiability becomes a more stringent and useful condition on theories of natural language. As an aid to discovering such characteristics, Section 3 presented a variety of learning strategies; both the restrictiveness and the psychological reality of these strategies were discussed.

Totality, nontriviality, and memory-limitedness were among the more realistic of the learning strategies developed in Section 3. It was seen that totality is by itself restrictive, and that the combination of all three strategies is more restrictive than any taken singly. These considerations suggest the following condition of adequacy on theories of natural language.

A theory of natural language is adequate only if the class of languages it specifies as natural is exactly identifiable by a total, nontrivial, \( n \)-memory-limited learning machine (for some reasonable choice of \( n \)).
Refinement of (C) will likely involve further consideration of strategies. However, the criterion of learning as well as the construal of environment offered by the Gold model must also be subjected to scrutiny on both empirical and formal grounds. These latter topics are addressed in Osherson and Weinstein (1982), and Osherson et al. (in preparation).

REFERENCES


OSHERSON, D., STOB, M., AND WEINSTEIN, S. (1983b), Environments for Learning, manuscript.


