CHAPTER 27

COMPUTATIONAL THEORIES OF LEARNING AND DEVELOPMENTAL PSYCHOLINGUISTICS

JEFFREY HEINZ

27.1 Introduction

A computer is something that computes, and since modern theories of cognition assume that humans make computations when processing information, humans are computers. What kinds of computations do humans make when they learn languages?

Answering this question requires the collaborative efforts of researchers in several different disciplines and sub-disciplines, including language science (e.g. theoretical linguistics, psycholinguistics, language acquisition), computer science, psychology, and cognitive science. The primary purpose of this chapter is explain to developmental psycholinguists and language scientists more generally the main conclusions and issues in computational learning theories. This chapter is needed because

1. the mathematical nature of the subject makes it largely inaccessible to those without the appropriate training (though hopefully this chapter shows that the amount of training required to understand the main issues is less than what is standardly assumed);
2. the literature contains a number of unfortunate, yet widely cited, misunderstandings of the relevance of work in computational learning for language learning. I will try to clarify these in this chapter.

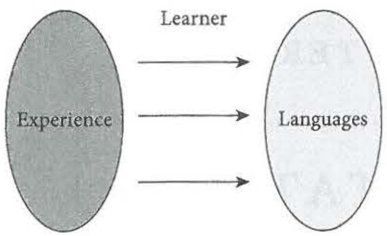


FIGURE 27.1 Learners are functions from experience to languages.

The main points in this chapter are:

1. The central problem of learning is generalization.
2. Consensus exists that, for feasible learning to occur at all, restricted, structured hypothesis spaces are necessary.
3. Debates pitting statistical learning against symbolic learning are misplaced. To the extent meaningful debate exists at all, it is about the learning criterion; i.e. how "learning" ought to be defined. In particular, it is about what kinds of experience learners are required to succeed on in order to say that they have "learned" something.
4. Computational learning theorists and developmental psycholinguists can profitably interact in the design of meaningful artificial language learning experiments.

In order to understand how a computer can be said to learn something, a definition of learning is required. Only then does it become possible to ask whether the behavior of the computer meets the necessary and sufficient conditions of learning required by the definition. Computational learning theories provide definitions of what it means to learn and then asks, under those definitions: What can be learned, how and why? Which definition is "correct" of course is where most of the issues lie.

At the most general level, a language learner is something that comes to know a language on the basis of its experience. All computational learning theories consider learners to be functions which map experience to languages (Figure 27.1). Therefore in order to define learning, both languages and experience need to be defined first.

27.2 Languages, Grammars, and Experience

27.2.1 Languages

Before we can speak of grammars, which are precise descriptions of languages, it will be useful to talk about languages themselves. In formal language theory, languages

are mathematical objects which exist independently of any grammar. They are usually defined as subsets of all logically possible strings of finite length constructible from a given alphabet. This can be generalized to probability distributions over all those strings, in which case they are called stochastic languages.

The alphabet can be anything, so long as it is unchanging and finite. Elements of the alphabet can represent IPA symbols, phonological features, morphemes, or words in the dictionary. If desired, the alphabet can also include structural information such as labeled phrasal boundaries. It follows that any description of sentences and words that language scientists employ can be described as a language or stochastic language with a finite alphabet.[1]

It is useful to consider the functional characterizations of both languages and stochastic languages because they are the mathematical objects of interest to language scientists. As functions, a language L maps strings to one only if the string is in the language and all other logically possible strings are mapped to zero. Stochastic languages, as functions, map all logically possible strings to real values between zero and one such that they sum to one. Figure 27.2 illustrates functional characterizations of English as a language and as a stochastic language. The functional characterization of English as a language only makes binary distinctions between well-formed and ill-formed sentences. On the other hand, the functional characterization of English as a stochastic language makes multiple distinctions. In both cases, the characterizations are infinite in the sense that both assign nonzero values to infinitely many possible sentences. This is because there is no principled upper bound on the length of possible English sentences.[2]

How stochastic languages are to be interpreted ought to always be carefully articulated. For example, if the real numbers are intended to indicate probabilities of occurrence then the functional characterization in Figure 27.2 says that "John sang" is twice as likely to occur as "John and Mary sang" On the other hand, if the real numbers are supposed to indicate well-formedness, then the claim is that "John sang" is twice as well-formed (or acceptable) as "John sang and Mary danced."[3]

As explained in the next section, from a computational perspective, the distinction between stochastic and non-stochastic languages is often unimportant. I use the word *pattern* to refer to both stochastic and non-stochastic languages in an intentionally ambiguous manner.

[1] Languages with infinite alphabets are also studied (Otto 1985), but they will not be discussed in this chapter.

[2] If there were, then there would be a value n such that "John sang $\underbrace{\text{and John sang}}_{n\text{-1times}}$ would be well-formed but "John sang $\underbrace{\text{and John sang}}_{n\text{times}}$" would be as ill-formed as "John and sang."

[3] There is a technical issue here. If there are infinitely many nonzero values, then it is not always the case that they can be normalized to yield a well-formed probability distribution. For example, if each sentence is equally acceptable, we would expect a uniform distribution. But the uniform distribution cannot be defined over infinitely many elements since the probability for each element goes to zero.

English as a language	English as a stochastic language
John sang → 1	John sang → 1.2×10^{-12}
John and sang → 0	John and sang → 0
John sang and Mary danced → 1	John sang and Mary danced → 2.4×10^{-12}

FIGURE 27.2 Fragments of functional characterizations of English as a language and a stochastic language.

27.2.2 Grammars

Grammars are finite descriptions of patterns. It is natural to ask whether every conceivable pattern has a grammar. The answer is No. In fact most logically possible patterns cannot be described by *any* grammar at all of any kind. There is an analogue to real numbers. Real numbers are infinitely long sequences of numbers and some are unpredictable in an important kind of way: no algorithm exists (nor can ever exist) which can generate the real number correctly up to some arbitrary finite length; such reals are called uncomputable. Sequences for which such algorithms do exist (like π) are *computable.*

More concretely, a real number is computable if and only if a Turing machine exists which can compute the exact value of the real number to any arbitrary degree of precision (and so can always provide the nth digit in its decimal expansion). A Turing machine is one of the most general kinds of computing device, and, by the Church–Turing thesis, Turing machines can instantiate any algorithm. Turing's (1937) discovery was that *uncomputable* real numbers turn out to be the most common kind of real number and so most real numbers cannot be computed by any algorithm! Such a result may be initially hard to understand (after all, what is an example of an uncomputable real number?),[4] but it is the foundation for the modern study of computation.

Like real numbers, most logically possible patterns cannot be described by any Turing machine or other kind of grammar. Grammars are algorithmic in the sense that they are of finite length but describe potentially infinitely-sized patterns. In this way, grammars are just like machines or any other computing device. The Chomsky Hierarchy classifies logically possible patterns into sets of nested regions (Figure 27.3). *Recursively Enumerable* (r.e.) patterns are those for which there exists a Turing machine which answers affirmatively when asked, for any nonzero valued string s belonging to the pattern, whether s in fact has a nonzero value (Turing 1937; Rogers 1967; Harrison 1978).[5]

[4] See Chaitin (2004).

[5] In contemporary theoretical computer science, the name "computably enumerable" is often used instead of "recursively enumerable." This class is also called "semi-decidable."

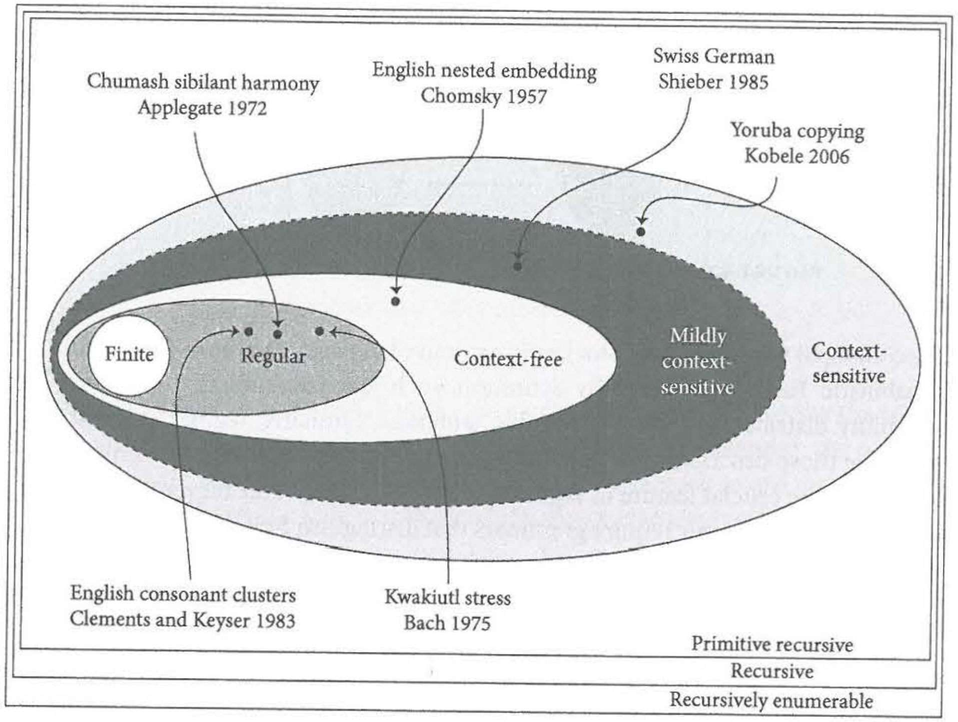


FIGURE 27.3 The Chomsky Hierarchy with natural language patterns indicated.

Recursive patterns are those for which a Turing machine exists, which, when asked what value the pattern assigns to any logically possible string, returns the right value.[6] Therefore, language scientists which attribute the ability to discriminate well-formed from ill-formed sentences as part of linguistic competence, are tacitly asserting that sentence patterns in natural language are recursive. Recursive patterns are also called computable, or Turing-computable.

Smaller regions correspond to patterns describable with increasingly less powerful machines (grammars). For example, the *regular* patterns are all those that can be described by machines that admit only finitely many internal states. In contrast, machines which generate nonregular patterns must have infinitely many internal states. The smallest region, the class of *finite* patterns, are those whose functional characterizations have only finitely many sentences with nonzero values. For further details regarding the Chomsky Hierarchy, readers are referred to Partee et al. (1993) and Sipser (1997).[7]

If the machines are probabilistic, then the stochastic counterparts of each class is obtained. Probabilistic machines are simply ones that may use random information

[6] This class is also called "decidable" because for any recursive pattern, it is always possible to decide for any input, what its value is (0 or 1 or something else). This is in contrast to the r.e. (or semi-decidable) class where the machine may not answer—and run forever—on inputs with zero values.

[7] Harrison (1978), Hopcroft *et al.* (1979, 2001), and Thomas (1997) offer more technical treatments.

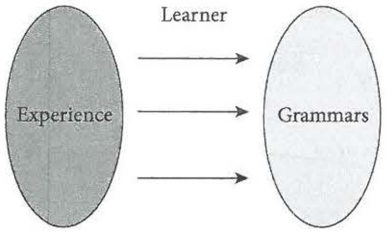


FIGURE 27.4 Learners are functions from experience to grammars.

(like coin flips) while running. Stochastic recursive languages are those describable with probabilistic Turing machines. By definition, such machines describe all computable probability distributions over all possible sentences. Similarly, regular stochastic languages are those describable by probabilistic machines which admit only finitely many states. Thus the crucial feature of regular patterns is not whether they are stochastic or not, but the fact they only require grammars that distinguish finitely many states.

It is of course of great interest to know what kinds of patterns natural languages are. Figure 27.3 shows where some natural language patterns fall in the Chomsky Hierarchy. For example, phonological patterns do not appear to require grammars that distinguish infinitely many states unlike some syntactic patterns, which appear to require grammars that do.[8] This distinction between these two linguistic domains is striking (Heinz and Idsardi 2011, 2013).

It is also important to understand the goals of computational research of natural language patterns. In particular, establishing complexity bounds is different from hypotheses which state both necessary *and sufficient* properties of possible natural language patterns. For example the hypothesis that natural language patterns are mildly context-sensitive (Joshi 1985), is a hypothesis that seeks to establish an upper bound on the complexity of natural language. Joshi is not claiming, as far as I know, that *any* mildly context-sensitive pattern is a possible natural language one. In my opinion, it is much more likely that possible natural language patterns belong to subclasses of the major regions of the Chomsky Hierarchy. For example, Heinz (2010a) hypothesizes that all phonotactic patterns belong to particular subregular classes. I return to these ideas in section 27.5.

Although from the perspective of formal language theory, grammars are the mathematical objects of secondary interest, it does matter that learners return a grammar, instead of a language. This is for the simple reason that, as mathematical objects, grammars are of finite length and the functional characterizations of patterns are infinitely long. Thus while Figure 27.1 describes learners as functions from experience to languages, they are more accurately described as functions from experience to grammars (Figure 27.4).

[8] For more on the hypothesis that all phonological patterns are regular Kaplan and Kay (1994); Eisner (1997); and Karttunen (1998). Readers are referred to Chomsky (1956) and Shieber (1985) for arguments concerning the nonregular nature of grammars for human syntax.

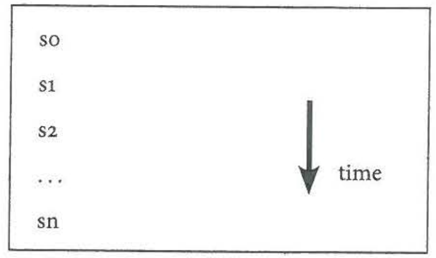


FIGURE 27.5 The learner's experience.

While the distinctions in the Chomsky Hierarchy can be used to classify the computational complexity of language patterns, they are much more general in the sense that they can be used to classify the complexity of many objects, such as real numbers, functions, or, as we will see, the kind of experience language learners receive in the course of learning.

27.2.3 Experience

There are many different kinds of experience learning theorists consider, but they agree that the experience is a finite sequence (Figure 27.5). It is necessary to decide what the elements s_i of the sequence are. In this chapter we distinguish four kinds of experience. *Positive evidence* refers to experience where each s_i is known to be a nonzero-valued sentence of the target pattern. *Positive and negative evidence* refers to experience where each s_i is given as belonging to the target pattern (has a nonzero value) or as not belonging (has a zero value). *Noisy evidence* refers to the fact that some of the experience is incorrect. For example, perhaps the learner has the experience that some s_i belongs to the target language, when in fact it does not (perhaps the learner heard a foreign sentence or someone misspoke). *Queried evidence* refers to experience learners may have because they specifically asked for it. In principle, there are many different kinds of queries learners could make. This chapter does not address these last two kinds; readers are referred to Angluin and Laird (1988) and Kearns and Li (1993) for noisy evidence; and to Angluin (1988a, 1990), Becerra-Bonache et al. (2006), and Tîrnauca (2008) for queries.

27.2.4 Learners as Functions

Armed with the basic concepts and vocabulary all learning theorists use to describe target languages, grammars, and experience, it is now possible to define learners. They are simply functions that map experience to grammars. For the most part formal learning theorists are concerned with *computable* functions. This is because an uncomputable learning function cannot be instantiated on any known computing device—such as a human brain—and furthermore by the Church–Turing thesis it is impossible for it to be instantiated on any computing device.

The characterization of learners in this section is very precise, but it is also very broad. Any learning procedure can be thought of as a function from experience to grammars, including connectionist ones (e.g. Rumelhart and McClelland 1986b), Bayesian ones (Griffiths et al. 2008), learners based on maximum entropy (e.g. Goldwater and Johnson 2003), as well as those embedded within generative models (Wexler and Culicover 1980; Berwick 1985; Niyogi and Berwick 1996; Tesar and Smolensky 2000; Niyogi 2006). Each of these learning models, and I would suggest any learning model, takes as its input a finite sequence of experience and outputs some grammar, which defines a language or a stochastic language. Consequently, all of these particular proposals are subject to the results of formal learning theory.

27.3 What is Learning?

It remains to be defined what it means for a function which maps experiences to grammars to be successful. After all, there are many logically possible such functions, but we are interested in evaluating particular learning proposals. For example, we may be interested in those learning functions that are human-like, or which return human-like grammars.[9]

27.3.1 Learning Criteria

It is important to define what it means to learn so that it is possible to determine what counts as a success. The general idea in the learning theory literature is that learning has been successful if the learner has *converged* to the right language. Is there some point n after which the learner's hypothesis does not change (much)? Convergence can be defined in different ways, to which I return in the next paragraph. Typically, learning theorists conceive of an infinite stream of experience to which the learner is exposed so that it makes sense to talk about a convergence point. Is there a point n such that for all $m \geq n$, Grammar $G_m \simeq G_n$ (given some definition of $\simeq$)? Figure 27.6 illustrates. The infinite streams of experience are also called *texts* (Gold 1967) and *data presentations* (Angluin 1988b). All three terms are used synonymously here.

[9] This section draws on a large set of learning literature. Readers are referred to Nowak et al. (2002) for an excellent, short introduction to computational learning theory. Niyogi (2006), de la Higuera (2010), and Clark and Lappin (2011) provide detailed, accessible treatments, and Anthony and Biggs (1992), Kearns and Vazirani (1994), Jain et al. (1999), Lange et al. (2008), and Zeugmann and Zilles (2008) provide technical introductions. I have also relied on the following research: Gold (1967); Horning (1969); Angluin (1980); Osherson et al. (1986); Angluin (1988b); Angluin and Laird (1988); Vapnik (1995, 1998); Case (1999).

datum	The Learner ϕ and its Hypotheses over time	
s_0	$\phi(\langle s_0\rangle) = G_0$	
s_1	$\phi(\langle s_0, s_1\rangle) = G_1$	
s_2	$\phi(\langle s_0, s_1, s_2\rangle) = G_2$	$\downarrow$ time
...		
s_n	$\phi(\langle s_0, s_1, s_2, \ldots, s_n\rangle) = G_n$	
...		
s_m	$\phi(\langle s_0, s_1, s_2, \ldots, s_m\rangle) = G_m$	
...		

FIGURE 27.6 If, for all **m≥n**, it is the case that $G_m \simeq G_n$ (given some definition of $\simeq$), then the learner is said to have converged.

Convergence has been defined in different ways, but there are generally two kinds. *Exact convergence* means that the learner's final hypothesis must be 100 percent correct. Alternatively, *approximate convergence* means the learner's final hypothesis need not be exact, but somehow "close" to 100 percent correct.

Defining successful learning as convergence to the right language after some point n, raises another question with respect to experience: on which infinite streams must a learner converge? Generally two kinds of requirements have been studied. Some infinite streams are *complete*; that is, every possible kind of information about the target language occurs at some point in the presentation of the data. For example, in the case of positive evidence, each sentence in the language would occur at some finite point in the stream of experience.

The second requirement is about whether the infinite streams are *computable*. This has two aspects. First, there are as many infinite texts as there are real numbers and so most of these sequences are not computable. Should learners be required to succeed on these? Or should learners only be required to succeed on those data sequences generable by Turing machines? The second aspect is more technical. Even if every sequence itself is computable, it may be the case that the *set* of all such sequences is *not* computable. This happens because, for each individual infinite sequence s in such a set, an algorithm exists which generates s, but no algorithm exists which can generate (all the algorithms for) all the sequences belonging to this set.[10]

[10] As an example, consider the halting problem. This problem takes as input a program p and an input i for p, and asks whether p will run forever on i, or if p will eventually halt. It is known that there are infinitely many programs which do not halt on some inputs. For each such program p choose some input i_p. Since i_p is an input, it is finitely long and can be generated by *some* program. But no program exists which can generate *every* such i_p. This is because if it could, it would follow that there is a solution to the halting problem. But in fact, the halting problem is known to be uncomputable; that is, no algorithm exists which solves it (Turing 1937).

The computablity of the data presentations is much more important than it may initially appear. In fact, its importance has been largely overlooked in interpreting the results of computational learning theory. As we will see, requiring learners to succeed on either *all* or *only computable* data presentations has important consequences for learnability.

27.3.2 Definitions of Learning

Table 27.1 summarizes the kinds of choices to be made when deciding what learning means. The division of the choices into columns labeled "Makes learning easier" and "Makes learning harder" ought to be obvious. Learners only exposed to positive evidence have more work to do than those given both positive and negative evidence. Similarly, learners who have to work with noisy evidence will have a more difficult task than those given noise-free evidence. Learners allowed to make queries have access to more information than those not permitted to make queries. Exact convergence is a very strict demand, and approximate convergence is less so. Finally, requiring learners to succeed for every logically possible presentation of the data makes learning harder than requiring learners only to succeed for complete or computable presentations simply because there are far fewer complete and/or computable presentations. Figure 27.7 shows the proper subset relationships among complete and computable presentations of data.

Using the coarse classification provided by Table 27.1, I now classify several definitions of learning (these are summarized in Table 27.2). The major results of these definitions are discussed in the next section.

1. Identification in the limit from positive data. Gold (1967) requires that the learner succeed with positive evidence only (A), noiseless evidence (b), and without queries (C). Exact convergence (D) is necessary: even if the grammar to which the learner converges generates a language which differs only in one sentence from the target language, this is counted as a failure. On the other hand, this framework is generous in that learners are only required to succeed on complete data presentations (e) but must succeed for any such sequence, not just computable ones (F).

2. Identification in the limit from positive and negative data. This is the same except the learner is exposed to both positive and negative evidence (a) (Gold 1967).

3. Identification in the limit from positive data with probability p. In this learning paradigm (Wiehagen et al. 1984; Pitt 1985), learners are probabilistic (i.e. have access to coin flips). Convergence is defined in terms of whether learners can identify the target language in the limit given any text with probability p. Thus this learning criterion is less strict than *identification in the limit from positive data* because exact convergence is replaced with a kind of approximate convergence (d). Otherwise, it is the same as identification in the limit from positive data.

Table 27.1 Choices providing a coarse classification of learning frameworks according to whether they make the learning problem easier or harder

	Makes learning easier		Makes learning harder
a.	positive and negative evidence	A.	positive evidence only
b.	noiseless evidence	B.	noisy evidence
c.	queries permitted	C.	queries not permitted
d.	approximate convergence	D.	exact convergence
e.	complete infinite streams	E.	any infinite sequence
f.	computable infinite streams	F.	any infinite sequence

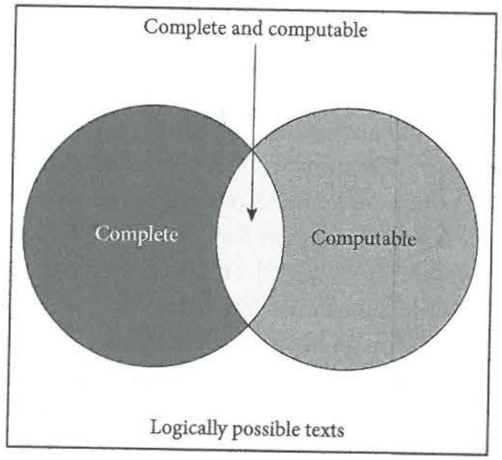


FIGURE 27.7 Subset relationships between all logically possible classes of texts, classes of complete texts, computable classes of texts, and both complete and computable classes of texts.

4. Identification in the limit from distribution-free positive stochastic data with probability p. Angluin (1988b) considers a variant of Pitt's framework immediately above where the data presentations are generated probabilistically from fixed, but arbitrary, probability distributions (including uncomputable ones). The term *distribution-free* refers to the fact that the distribution generating the data presentation is completely arbitrary. Like the previous framework, it is similar to *identification in the limit from positive data* but makes an easier choice with respect to convergence (d).

5. Identification in the limit from positive recursive data. Wiehagen (1977) considers a paradigm which is similar to *identification in the limit from positive data* except that the

learner is only required to succeed on complete, computable streams (f), and not any stream. The particular streams that learners are required to succeed on are those generable by *recursive* functions.

6. Identification in the limit from positive primitive recursive data. This paradigm, also studied by Gold (1967), is similar to the one at point 5. In fact, in terms of the classification scheme in Table 27.1, it is exactly the same. However, this paradigm makes stronger assumptions about the nature of the experience language learners receive as input. Here the data presentations that learners are required to succeed on are only those generable by *primitive recursive* functions. This class is nested between the recursive class and the context-sensitive class (see Figure 27.3). Therefore, learning in this framework is "easier" than in the one in point 5 because there are *fewer* data presentations learners need to succeed on.

7. Identification in the limit from computable positive stochastic data. Horning (1969), Osherson et al. (1986), and Angluin (1988b) study learning stochastic languages from positive data. Horning studies stochastic languages generated by context-free grammars where the rules are assigned probabilities with rational values. I focus on Angluin's framework since she generalizes his study (and those of earlier researchers) to obtain the strongest result.

Angluin studies *approximately computable* stochastic languages. Recall that a stochastic language, or distribution, D maps a string s to a real number, so $D(s) = r$. A distribution is approximately computable if and only if, for all strings s and for all positive rational numbers ε, there is a total recursive function f which is a rational approximation of D within ε; that is, such that $|D(s) - f(s,\varepsilon)| < \varepsilon$. The approximately computable stochastic languages properly include the context-free ones.

In Angluin (1988b), as in Horning (1969), the data presentations must be generated according to the target distribution, which is fixed and is approximately computable. In this way, this definition of learning is like *identification in the limit from positive recursive texts* because learners do need to succeed on any data presentation, but only on complete and computable ones (f).[11] On the other hand, instead of exact convergence, convergence need only be approximate (d).

8. Probably Approximately Correct (PAC). This framework makes a number of different assumptions (Valiant 1984; Anthony and Biggs 1992; Kearns and Vazirani 1994). Both positive and negative evidence are permitted (a). Noise and queries are not permitted (b,c). Convergence need only be approximate (d), but the learner must succeed for any kind of data presentation, both non-complete and uncomputable (E,F). What counts as convergence is tied to the degree of "non-completeness" of the data presentation.

[11] If a data presentation is being generated from a computable stochastic language, then it is also complete. This is because for any sentence with nonzero probability, the probability of this sentence occurring increases monotonically to one as the size of the experience grows. For example, it is certain that the unlikely side of a biased coin will appear if it is flipped enough times.

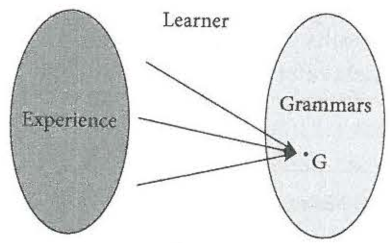


FIGURE 27.8 Learners which are constant functions map all possible experience to a single grammar.

To summarize this subsection, there have been many different definitions of what it means to "learn." In the next section, the major results within each of these frameworks will be discussed. The factorization of these frameworks by the general properties listed in Table 27.1 makes it easier to interpret the results presented in the next section.

27.3.3 Classes of Languages

Before continuing to section 27.4, it is important to recognize that computational learning theories are concerned with learners of *classes of languages* and not just single languages. This is primarily because every language can be learned by a constant function (Figure 27.8). For example, with any of definitions given in the list above, it is easy to state a learner for English (and just English). Just map all experience (no matter what it is) to a grammar for English. Even if we do not know what this grammar is yet, the learning problem is "solved" once we know it. Obviously, such "solutions" to the learning problem are useless, even if mathematically correct.

For this reason, computational learning theories ask whether a collection of more than one language can be learned by the same learner. This more meaningfully captures the kinds of question language scientists are interested in: Is there a single procedure that not only learns English, but also Spanish, Arabic, Inuktitut, and so on?

27.4 Results of Computational Learning Theories

Computational learning theorists have identified, given the definitions in the previous section, classes of languages that can and cannot be learned. Generally, formal learning theorists are interested in large classes of learnable languages because they want to see what is possible in principle. If classes of languages are learnable in principle, the next important question is whether they are *feasibly* learnable. This means whether learners can succeed with reasonable amounts of time and effort where reasonable is defined in

Table 27.2 Foundational results in computational learning theory. Letters in square brackets refer to properties in Table 27.1

Definition of learning	Feasible learnability of the major regions of the Chomsky Hierarchy
1. Identification in the limit from positive data [A b c D e F]	Finite languages are learnable but no superfinite class of languages is learnable, and hence neither are the regular, context-free, context-sensitive, recursive, nor r.e. languages.
2. Identification in the limit from positive and negative data [a b c D e F]	Recursive languages are learnable but the regular languages are not feasibly learnable.
3. Identification in the limit from positive data with probability p [A b c d e F]	For all $p > 2/3$: same as those identifiable in the limit from positive data.
4. Identification in the limit from distribution-free positive stochastic data with probability p [A b c d e F]	For all $p > 2/3$: same as those identifiable in the limit from positive data.
5. Identification in the limit from positive recursive data [A b c D e f]	Same as those identifiable in the limit from positive data.
6. Identification in the limit from positive primitive recursive data [A b c D e f]	R.e. languages are learnable but not feasibly.
7. Identification in the limit from computable positive stochastic data [A b c d e f]	Recursive stochastic languages are learnable but not feasibly.
8. Probably Approximately Correct [a b c d E F]	The finite languages are not learnable and hence neither are the regular, context-free, context-sensitive, recursive, nor r.e. languages.

standard ways according to computational complexity theory (Garey and Johnson 1979; Papadimitriou 1994).[12]

This section provides the largest classes known to be provably learnable under the different definitions of learning above. Where possible, I also indicate whether such classes can be feasibly learned. If one is not familiar with the regions in the Chomsky Hierarchy, it will be helpful to familiarize oneself with them before continuing (Figure 27.3). Table 27.2 summarizes the following discussion.

27.4.1 No Major Region of the Chomsky Hierarchy is Feasibly Learnable

Gold (1967) proved three important results. First, a learner exists which identifies the class of recursive languages in the limit from positive and negative data. Second, a learner exists which identifies the finite languages in the limit from positive data, but no learner exists which can identify any superfinite class in the limit from positive data. Superfinite classes of languages are those that include all finite languages and at least one infinite language. It follows from this result that none of the major regions of the Chomsky Hierarchy are identifiable in the limit from positive data by *any* learner which can be defined as mapping experience to grammars. It is this result with which Gold's paper has become identified. Gold's third (and usually overlooked) result is that if learning is defined so that learners need only succeed given complete, positive, primitive recursive texts, then a learner does exist which can learn the class of r.e. languages.

Wiehagen (1977) shows that if learning is defined so that learners need only succeed given complete, positive, recursive texts, then only those classes identifiable in the limit from positive data are learnable. Therefore no superfinite class is learnable in this setting. In other words, comparison of this result with Gold's third result shows that restricting the data presentations to recursive texts does not increase learning power, but restricting them to primitive recursive texts does (see also Case 1999).

Angluin (1988b), developing work begun in Horning (1969) and extended by Osherson et al. (1986), presents a result for stochastic languages similar in spirit to the ones above. She shows that under the learning criteria that learners are only required to succeed for presentations of the positive data generable by the target stochastic language, then the class of recursive stochastic languages is learnable.[13]

This result contrasts sharply with other frameworks that investigate the power of probabilistic learning frameworks. Wiehagen et al. (1984) and Pitt (1985) show that the class of languages identifiable in the limit from positive data with probability p is the same as the class of languages identifiable in the limit from positive data whenever. Angluin (1988b: 5) concludes "These results show that if the probability of identification is required to be above some threshold, randomization is no advantage."

Angluin also shows that for all p, the class of languages identifiable in the limit from positive data with probability p from distribution-free stochastic data is exactly the same as the the class of languages identifiable in the limit from positive data with probability p. Angluin observes that the "assumption of positive stochastic rather than positive [data presentations] is no help, if we require convergence with any probability greater than 2/3." She concludes "the results show that if no assumption is made about the probability distribution [generating the data presentations], stochastic input gives no greater power than the ability to flip coins."

Finally, in the PAC learning framework (Valiant 1984), not even the class of finite languages is learnable (Blumer et al. 1989).

In the cases where learners are known to exist in principle, we may examine their feasibility. In the case of the identification in the limit from positive and negative data,

[12] Discussion of how to measure the computational complexity of learning algorithms is discussed in detail in Valiant (1984); Pitt (1989); de la Higuera (1997, 2010); and Clark and Lappin (2011).

[13] Technically, she shows that the class of *approximately computable* distributions is learnable. The crucial feature of this class is that its elements are enumerable and computable, which is why I take some liberty in calling them recursive stochastic languages.

Gold (1978) shows that there are no feasible learners for even the regular class of languages. In other words, while learners exist in principle for the recursive class, they consume too much time and resources in the worst cases. In the case of identification in the limit from primitive recursive texts and identification in the limit from computable positive stochastic data, the learners known to exist in principle are also not feasible.[14]

Table 27.2 summarizes the results discussed in this section. It is worth examining this table to see exactly what makes learning the recursive class possible in principle. I will return to understanding this in section 27.5.

27.4.2 Other Results

The facts as presened appear to paint a dismal picture—either large regions of the Chomsky Hierarchy are not learnable even in principle, or if they are, they are not feasibly learnable.

However, there are many feasible learners for classes of languages even in the frameworks with the most demanding criteria, such as identification in the limit from positive data and PAC-learning. This rich literature includes Angluin (1980, 1982); Muggleton (1990); Garcia et al. (1990); Anthony and Biggs (1992); Kearns and Vazirani (1994); García and Ruiz (1996, 2004); Fernau (2003); Oates et al. (2006); Clark and Eyraud (2007); Heinz (2008, 2009, 2010a, 2010b); Yoshinaka (2008, 2011); Becerra-Bonache et al. (2010); Clark et al. (2010); Kasprzik and Kötzing (2010); Clark and Lappin (2011); and many others (see for example de la Higuera 2005, 2010). The language classes discussed in those works are not major regions of the Chomsky Hierarchy, but are *subclasses* of such regions.

Some of these language classes are of infinite size and include infinite languages—but they crucially exclude some finite languages so they are not superfinite language classes. Figure 27.9 illustrates the nature of these classes. I return to this point in section 27.5.4 when discussing why the fundamental problem of learning is generalization.

Also, the proofs that these classes are learnable are constructive, so concrete learning algorithms whose behavior is understood exist. The algorithms are successful because they utilize the structure inherent in the class, or equivalently, of its defining properties, to generalize correctly. Often the proofs of the algorithm's success involve characterizing the kind of finite experience learners need in order to make the right generalizations.

To sum up, even though identification in the limit from positive data and PAC-learning make the learning problem harder by requiring learners to succeed for any data presentation, so that no learners exist for superfinite classes of languages even in principle, there are feasibly learnable language classes in these frameworks. Furthermore, many of the above researchers have been keen to point out the patterns resembling natural language, which belong to these learnable subclasses.

[14] These learners essentially compute an ordered list of grammars for the patterns within the target class. With each new data point, they find the first grammar in this list compatible with the experience so far.

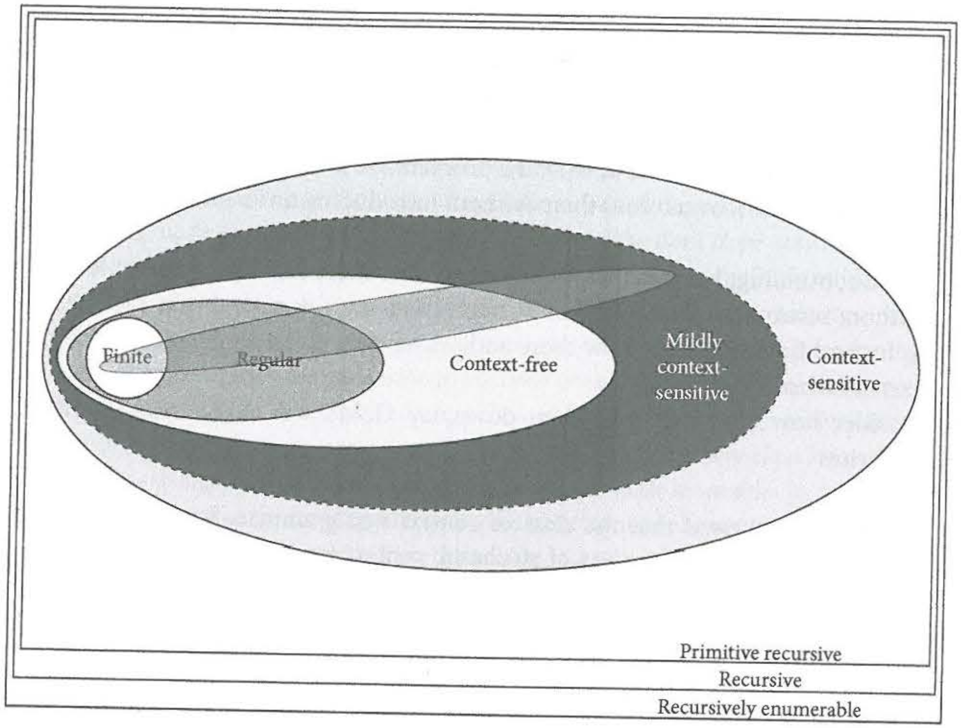


FIGURE 27.9 A non-superfinite class of patterns which cross-cuts the Chomsky Hierarchy.

27.5 Interpreting Results of Computational Learning Theories

27.5.1 Wrong Reactions

How have the above results been interpreted? Are those interpretations justified? Perhaps the most widespread myth about formal learning theory is the oft-repeated claim that Gold (1967) is irrelevant because

1. Gold's characterization of the learning problem makes unrealistic assumptions;
2. Horning (1969) showed that statistical learning is more powerful than symbolic learning.

As this section shows, these claims have been made by influential researchers in cognitive science, computer science, computational linguistics, and psychology.

In this section I rebut these charges. The authors cited here repeatedly fail to distinguish different definitions of learnability, fail to identify Gold (1967) with anything other than identification in the limit from positive data, and/or make false statements about the kinds of learning procedures Gold (1967) considers. With respect to the claim that identification in the limit makes unrealistic assumptions, I believe it is fair to debate the assumptions underlying any learning framework. However, the arguments put forward by the authors discussed in this section are not convincing, usually because they say very little about what the problematic assumptions are and how their proposed framework overcomes them without introducing unrealistic assumptions of their own.

Before continuing, I would like to make clear that these criticisms are not leveled at the authors' research itself, which is often interesting, important, and valuable in its own right. Instead I am critical of how these authors have motivated their work within the context of formal learning theory.

Consider how Horning is used to downplay Gold's work. For example, Abney (1996) writes

> though Gold showed that the class of context free grammars is not learnable, Horning showed that the class of stochastic context free grammars is learnable. (Abney 1996: 21)

The first clause only makes sense if, by "Gold," Abney is referring to identification in the limit from positive data. After all, Gold did show that the context-free languages are learnable not only from positive and negative data, but also from positive data alone if the learners are only required to succeed on positive, primitive recursive data presentations (#5 in Table 27.2).

As for the second clause, Abney leaves it to the reader to infer that Gold and Horning are studying different definitions of learnability. Abney emphasizes the stochastic nature of Horning's target grammars as if that is the key difference in their results, but it should be clear from section 27.4 and Table 27.2 that the gain in learnability is not coming *solely* from the stochastic nature of the target patterns.

The fact that the only data presentations learners are required to succeed on are computable ones also plays an important role. Several comparisons make this clear. First, approximate, probabilistic convergence *itself* does not appreciably increase learning power. This is made clear by comparing *identification in the limit from positive data* with *identification in the limit from positive data with probability p* (#1 and #3 in Table 27.2). Second, learning stochastic languages instead of non-stochastic languages also does not increase learning power. This is made clear by comparing *identification in the limit from positive data with probability p* with *identification in the limit from positive stochastic data with probability p* (#3 and #4 in Table 27.2). Consideration of the PAC learning paradigm bolsters these comparisons. PAC allows approximate convergence and target classes of stochastic languages (in addition to positive and negative data), yet not even the finite class of languages is learnable.

What is responsble for these results? In those frameworks, learners are required to succeed for *any* data presentation. As Gold (1967) established in a non-stochastic setting (*identification in the limit from positive primitive recursive data*), the picture changes dramatically

when learners are only required to succeed on data presentations which are not arbitrarily complex. Likewise, Horning's results follow in no small part from the fact that learners are only required to succeed on *computable* data presentations, instead of all arbitrary ones (choice f/F in Table 27.1). The same holds true for Angluin's (1988b) extension of Horning's work to recursive stochastic languages (approximately computable distributions).

However, computability of the data presentations is not the only factor in Angluin's result. This is made clear by comparing *identification in the limit from positive data* with *identification in the limit from positive recursive data* (#1 and #5 in Table 27.2). In both cases, no superfinite class of languages is learnable. In non-stochastic settings, one has to reduce the complexity of the data presentations to primitive recursive ones for the r.e. class to become learnable (*identification in the limit from positive recursive data*). In other words, in non-stochastic settings, reducing the complexity of the data presentations to the computable, recursive class is not sufficient to make the recursive class learnable, but in stochastic settings, it is enough to make the recursive class learnable. In other words, the stochastic nature of the target patterns *in combination with* the reduced complexity of the data presentations is what makes the difference in Angluin's (and Horning's) results.

However, most researchers fail to appreciate the distinctions drawn here. For example, in the introductory text to computational linguistics, Manning and Schütze (1999) write

> Gold (1967) showed that CFGs [context-free grammars] cannot be learned (in the sense of identification in the limit—that is whether one can identify a grammar if one is allowed to see as much data produced by the grammar as one wants) without the use of negative evidence (the provision of ungrammatical examples). But PCFGs [probabilistic context-free grammars] can be learned from positive data alone (Horning 1969). (However, doing grammar induction from scratch is still a difficult, largely unsolved problem, and hence much emphasis has been placed on learning from bracketed corpora . . .). (Manning and Schütze 1999: 386–7)

Like Abney (1996), Manning and Schütze do not mention Gold's third result that CFGs can be learned if the data presentations are limited to primitive recursive ones. To their credit, they acknowledge the hard problem of learning PCFGs despite Horning's (and later Angluin's) results. Horning's and Angluin's learners are completely impractical and are unlikely to be the basis for any feasible learning strategy for PCFGs. For this reason, these positive learning results offer little insight on how PCFGs which describe natural language patterns may actually be induced from the kinds of corpus data that Manning and Schütze have in mind.

Similarly, in his influential and important thesis on the unsupervised learning of syntactic structure, Klein (2005) writes:

> Gold's formalization is open to a wide array of objections. First, as mentioned above, who knows whether all children in a linguistic community actually do learn the

same language? All we really know is that their languages are similar enough to enable normal communication. Second, for families of probabilistic languages, why not assume that the examples are sampled according to the target language's distribution? Then, while a very large corpus won't contain every sentence in the language, it can be expected to contain the common ones. Indeed, while the family of context-free grammars is unlearnable in the Gold sense, Horning (1969) shows that a slightly softer form of identification is possible for the family of probabilistic context-free grammars if these two constraints are relaxed (and a strong assumption about priors over grammars is made). (Klein 2005: 4–5)

Again, by "Gold's formalization," Klein must be referring to identification in the limit from positive data. Klein's first point is that it is unrealistic to use exact convergence as a requirement because we do not know if children in communities all learn exactly the same language, and it is much more plausible that they learn languages that are highly similar, but different in some details. Hopefully by now it is clear that Klein is misplacing the reason why it is impossible to identify in the limit from positive data superfinite classes of languages. It is not because of exact convergence; it is because learners are required to succeed for any complete presentation of the data, not just the computable ones. In frameworks that allow looser definitions of convergence (PAC-learning, identification in the limit from positive data with probability p), the main results are more or less the same as in identification in the limit from positive data. A crucial component of Horning's success is made clear in Angluin (1988b): identification in the limit from computable positive stochastic data only requires learners to succeed for data presentations which are computable. As for the unrealistic nature of exact convergence, is it not a useful abstraction? It lets one ignore the variation that exists in reality to concentrate on the core properties of natural language that make learning possible.

Klein then claims that it is much more reasonable to assume that the data presentations are generated by a fixed unchanging probability distribution defined by the target PCFG. This idealization may lead to fruitful research, but it is hard to accept it as realistic. That would mean that for each of us, in our lives, every sentence we have heard up until this point, and will hear until we die, is being generated by a fixed unchanging probability distribution. It is hard to see how this could be true given that what is actually said is determined by so many non-linguistic factors.[15] So if realism is one basis for the "wide array of objections" that Klein mentions, the alternative proposed does not look any better.

Like Klein (2005), Bates and Elman (1996) also argue that Gold (1967) is irrelevant because of unrealistic assumptions. They write:

A formal proof by Gold [1967] appeared to support this assumption, although Gold's theorem is relevant only if we make assumptions about the nature of the learning

[15] Even if we abstract away from actual words and ask whether strings of linguistic categories are generated by fixed underlying PCFGs, the claim is probably false. Imperative structures often have different distributions of categories than declaratives and questions, and the extent to which these are used in discourse depends entirely on nonlinguistic factors in the real world.

device that are wildly unlike the conditions that hold in any known nervous system [Elman et al. 1996]. (Bates and Elman 1996: 1849)

By now we are familiar with authors identifying Gold (1967) solely with identification in the limit from positive data. What assumptions does Gold make that are "wildly unlike the conditions that hold in any known nervous system?" Gold only assumes that learners are functions from finite sequences of experience to grammars. It is not clear to me why this assumption is not applicable to nervous systems, or any other computer. Perhaps Bates and Elman are taking issues with exact convergence, but as mentioned above, learning frameworks that allow looser definitions of convergence do not change the main results, and even Elman et al. (1996) employ abstract models.

Perfors et al. (2010) partially motivate an appealing approach to language learning that balances preferences for simple grammars with good fits to the data with the following:

Traditional approaches to formal language theory and learnability are unhelpful because they presume that a learner does not take either simplicity or degree of fit into account (Gold 1967). A Bayesian approach, by contrast, provides an intuitive and principled way to calculate the tradeoff. . . . Indeed it has been formally proven that an ideal learner incorporating a simplicity metric will be able to predict the sentences of the language with an error of zero as the size of the corpus goes to infinity (Solomonoff 1978, Chater and Vitànyi 2007); in many more traditional approaches, the correct grammar cannot be learned even when the number of sentences is infinite (Gold 1967). However learning a grammar (in a probabilistic sense) is possible, given reasonable sampling assumptions, if the learner is sensitive to the statistics of language (Horning 1969). (Perfors et al. 2010: 163)

The first sentence is simply false. While it is true that Gold does not specifically refer to learners which take either simplicity or degree of fit into account, that in no way implies his results do not apply to such learners. Gold's results apply to any algorithms that can be said to map finite sequences of experience to grammars, and the Bayesian models Perfors et al. propose are such algorithms. The fact that Gold does not specifically mention these particular traits emphasizes how general and powerful Gold's results are. If Perfors et al. really believe Bayesian learners can identify a superfinite class of languages in the limit from positive data, they should go ahead and try to prove it. (Unfortunately for them, Gold's proof is correct so we already know it is useless trying.)

I address Chater and Vitànyi's (2007) work in section 27.5.2 so let us move now to the statement that learners that are "sensitive to the statistics of language" can learn probabilistic grammars. This is attributed to Horning with no substantial discussion of the real issues. Readers are left believing in the power of statistical learning, unaware of the real issue of whether learning has been defined in a way as to require learners to succeed on complete and computable data presentations versus all complete data presentations. Again Gold showed that any r.e. language can be learned from positive primitive recursive texts. Angluin (1988b) showed that learners that are "sensitive to the statistics of language" are not suddenly more powerful (*identification in the limit from distribution-free positive stochastic data with probability p*). Finally Perfors et al. (2010) hide another issue behind the phrase "under reasonable sampling conditions." As mentioned previously in the discussion of Klein, I think there is every reason to question how reasonable those assumptions are. But I would be happy if the debate could at least get away from "the sensitivity of the learner to statistics" rhetoric to whether the assumption that actual data presentations are generated according to fixed unchanging computable probability distributions is reasonable. That would be progress and would reflect one actual lesson from computational learning theory.

27.5.2 Chater and Vitànyi (2007)

Chater and Vitànyi (2007), who extend work begun in Solomonoff (1978), provide a more accurate, substantial, and overall fairer portrayal of Gold's (1967) paper than these others, and corroborate some of the points made in this chapter. However, a couple of inaccuracies remain. Consider the following passage:

> Gold (1967) notes that the demand that language can be learned from every text may be too strong. That is, he allows the possibility that language learning from positive evidence may be possible precisely because there are restrictions on which texts are possible. As we have noted, when texts are restricted severely, e.g., they are independent, identical samples from a probability distribution over sentences, positive results become provable (e.g., Pitt, 1989); but the present framework does not require such restrictive assumptions. (Chater and Vitànyi 2007: 153)

This quote is misleading in a couple of ways. First, Gold (1967) goes much farther than just suggesting learning from positive evidence alone may be possible if the texts are restricted; in fact, he shows this (*identification in the limit from positive primitive recursive texts*).

Second, the claim that their framework does not assume that the streams of experience which are the inputs to the "ideal language learner" is not "restrictive" depends on what one considers to be "restrictive."[16] Section 2.1 of Chater and Vitànyi's (2007) paper explains exactly how the input to the learner is generated. They explain very clearly that they add probabilities to a Turing machine, in much the same way as probabilities can be added to any automaton. In this case, the consequence is they are able to describe recursive stochastic languages. In fact they conclude this section with the following sentence:

[16] The reference to Pitt (1989) is also odd given that this paper does not actually provide the positive results the authors suggest as it discusses identification in the limit from positive data with probability *p*. Horning (1969), Osherson et al. (1986), or Angluin (1988b) are much more appropriate references here.

> The fundamental assumption concerning the nature of the linguistic input outlined in this subsection can be summarized as the assumption that the linguistic input is generated by some monotone computable probability distribution $\mu_c(x)$. (Chater and Vitànyi 2007: 138)

Thus in one sense their assumption is restrictive because the linguistic input is limited to computable presentations (option F in Table 27.1). One important lesson from computational learning theory that this chapter is trying to get across is that assuming that the data presentations (the linguistic input in Chater and Vitànyi's terms) are drawn from a computable class is a *a primary factor* in determining whether all recursive patterns can be learned in principle or whether only superfinite classes can be.

On the other hand, the quoted passage from Chater and Vitànyi (2007: 153) above is correct that they are able to relax an assumption made by Angluin (1988b) (and Horning). The data presentations in Chater and Vitànyi's learning scenario do not need to be generated by a *fixed* probability distribution that does not change over time. Instead they obtain their result even allowing *non-stationary* distributions. This means the probability distribution at any given point in the data presentation depends on the sequence of data up to that point. In this way their learning framework overcomes the criticisms I leveled in earlier sections at other researchers who claim that Horning's learning framework is more realistic than identification in the limit from positive data. On these grounds, Chater and Vitànyi's result represents a real advance.

But at what cost? There is another important difference between the "ideal language learner" and Angluin's (1988b) learner which should not be overlooked. As Chater and Vitnàyi state clearly in their introduction (2007: 136): "Indeed, the ideal learner we consider here is able to make calculations that are known to be uncomputable." In other words, not only is the ideal language learner not feasibly computable, it is not computable at all! The fact that the "ideal language learner" can learn all recursive stochastic languages from data presentations generated by computable, non-stationary probability distributions therefore significantly departs from the learning results described in Table 27.2, all of which were assuming learners themselves must be computable functions!

If uncomputable learners are worthy of discussion, then it is important to know that the picture changes dramatically in *non*-stochastic settings. In particular, uncomputable learners with recursive data presentations can learn the r.e. class (Jain et al. 1999: 183)! In other words, permitting uncomputable learners significantly changes the results for *identification in the limit from positive recursive data* (#5 in Table 27.2). Jain et al. write (1999: 183) "It should be noted that if caretakers and natural phenomena are assumed to be computer simulable, then there is no reason to consider ... noncomputable scientists and children."

Chater and Vitànyi also discuss the feasibility of the learner again towards the end of their paper, where they point to a "crucial set of open questions" regarding "how rapidly learners can converge well enough" with the kinds of data in a child's linguistic environment. Of course it may be that there is some subclass of the recursive stochastic languages that the algorithm is able to learn feasibly, and which may include natural language patterns. In my view, research in this direction would be a positive development.

27.5.3 Clark and Lappin (2011)

Let us now turn to the landmark text by Clark and Lappin (2011). This book provides a thorough and welcome discussion of different computational learning theories and natural language acquisition. Many of the learning frameworks discussed in this chapter are surveyed there, and in many instances Clark and Lappin (2011) presents the same facts presented here. Nonetheless, Clark and Lappin argue forcefully against *identification in the limit from positive data* as an insightful learning paradigm, instead favoring probabilistic learning frameworks. It is remarkable to me how the same set of facts can be interpreted so differently.

Clark and Lappin fault *identification in the limit from positive data* for making "overly pessimistic idealizing assumptions" (2011: 89). In particular, they identify the "the major problem with the Gold paradigm" as the fact that "it requires learning under every presentation" (2011: 102), including "an adversarial presentation of the data designed to undermine learning" (2011: 97). As they emphasize throughout, this learning paradigm "does not rule out an adversarial teacher who organizes a presentation in a way designed to undermine learning, for example by presenting a string an indefinite number of times at the beginning of a data sequence" (2011: 208).

Instead, Clark and Lappin come down squarely in favor of probabilistic learning paradigms. They write "Recent work in probabilistic learning theory offers more realistic frameworks within which to explore the formal limits of human language acquisition" (2011: 98) and that "it is formally more convenient to model language acquisition in a probabilistic paradigm" (2011: 106). Also: "When we abstract away from issues of computational complexity, learning [within a probabilsitic paradigm] is broadly tractable. The first results along these lines are from Horning (1969)" (2011: 109).

I find many of Clark and Lappin's arguments selective. For example, the last statement about ignoring issues of computational complexity is odd because 12 pages earlier this was a criticism of the paradigms in Gold (1967): They "suffer from a lack of computational realism in that they disregard complexity factors and permit the learner unlimited quantities of time and data" (2011: 97). (An excellent discussion of computational complexity occurs in Chapter 7 to which I return in section 27.5.4.) Another example comes from a defense of Horning's research: "Horning's work is indeed limited, but it is not the endpoint of this research. Subsequent work greatly extends his results." (2011: 109). Surely such a defense applies to *identification in the limit from positive data*! Gold (1967) was certainly not the endpoint of research and has been extensively studied, extended, and used to better understand learning, notably by Angluin (1980, 1982, 1988a), respectively, and even by Clark in his own research with his colleagues (Clark and Eyraud 2007; Clark et al. 2010; among others). Chapter 8 of Clark and Lappin (2011) is titled "Positive Results in Efficient Learning," and highlights results set in the *identification*

in the limit from positive data paradigm! Gold's (1967) research has led to many variants and variations as described in the books by Osherson et al. (1986); Jain et al. (1999) and in the surveys by Lange et al. (2008) and Zeugmann and Zilles (2008), including variants that specifically address questions relevant to natural language acquisition, such as U-shaped learning (Carlucci et al. 2004, 2007, 2013).

Returning to the substantive argument regarding adversarial data presentations, it is true that requiring learners to succeed on every data presentation is a significant factor which contributes to the result that only superfinite classes of languages are learnable under the *identification in the limit from positive data* paradigm.[17] But, as discussed above, this is a factor even in stochastic settings! Clark and Lappin are clearly aware of this. For example, on page 99 when comparing the results of *identification in the limt from positive data* with *identification in the limit from positive and negative data* and *identification in the limit from distribution-free positive stochastic data with probability p* ((#1, #2 and #4 in Table 27.2), they write:

> [Angluin 1988b] summarizes the situation with respect to various probabilistic models that we discuss later: "These results suggest that the presence of probabilistic data largely compensates for the absence of negative data."
> However, this conclusion must be qualified, as it depends heavily on the class of distributions under which learning proceeds. (Clark and Lappin 2011: 99)

And later, they discuss Angluin's (1988b) identification in the limit from distribution-free positive stochastic data with probability p and explain that:

> allowing an adversary to pick the distribution over a presentation has the same effect as permitting an adversary to pick a presentation. This effect highlights an important fact: selecting the right set of distributions in a probabilistic learning paradigm is as important as selecting the right set of presentations in the [identification in the limit from positive data] paradigm. (Clark and Lappin 2011: 111)

This is one of the primary lessons of computational learning theory that this chapter has presented.[18]

Finally, it is worth emphasizing that frameworks which require learners to succeed only on complete and computable data presentations are weaker than frameworks which require learners to succeed on all complete data presentations, computable and uncomputable, for the simple reason that there are more data presentations of

[17] However, the example given of an adversarial teacher is not persuasive to me because the problem is not adversarial teachers per se, but adversarial teachers that can generate data presentations that are more complex than those generable by primitive recursive functions.

[18] I suspect Clark and Lappin may have misread the sentence quoted on page 99 of their book from Angluin (1988b: 2). They present Angluin's sentence as a conclusion she has drawn from her study. But this sentence, which occurs in the introduction of Angluin's paper, is referring to earlier results, which *suggest* that probabilistic data play this kind of role. She is setting up the topic which her paper

the latter type (Figure 27.7). Learners successful in these more difficult frameworks (mentioned in section 27.4.2) are more robust in the sense that they are guaranteed to succeed for *any* data presentation, even uncomputable ones. The fact that there are feasible learners which can learn interesting classes of languages under such strong definitions of learning underscores how powerful the positive learning results in these frameworks are.

27.5.4 Right Reactions

Gold (1967) provides three ways to interpret his three main results:

> 1. The class of natural languages is much smaller than one would expect from our present models of syntax. That is, even if English is context-sensitive, it is not true that any context-sensitive language can occur naturally. . . . In particular the results on [identification in the limit from positive data] imply the following: The class of possible natural languages, if it contains languages of infinite cardinality, cannot contain all languages of finite cardinality.
> 2. The child receives negative instances by being corrected in a way that we do not recognize . . .
> 3. There is an a priori restriction on the class of texts [presentations of data; i.e. infinite sequences of experience] which can occur. . . (Gold 1967: 453–4)

The first possibility follows directly from the fact that no superfinite class of languages is identifiable in the limit from positive data. The second and third possibilities follow from Gold's other results on *identification in the limit from positive and negative data* and on *identification in the limit from positive primitive recursive data* (#2 and #6 in Table 27.2).

Each of these research directions can be fruitful, if honestly pursued. For the case of language acquisition, Gold's three suggestions can be investigated empirically. We ought to ask

1. What evidence exists that possible natural language patterns form subclasses of major regions of the Chomsky Hierarchy?
2. What evidence exists that children receive positive and negative evidence in some, perhaps implicit, form?

investigates. The next sentence in Angluin (1988) reads "These results also invite comparison with a new criterion for finite learnability proposed by Valiant [(Valiant 1984)]." And she continues:

> Our study is motivated by the question of what has to be assumed about the probability distribution in order to achieve the kinds of positive results on language identification. We define a variant of Valiant's finite criterion for language identification, and show that in this case, the assumption of stochastically generated examples does not enlarge the class of learnable sets of languages.

In other words, Angluin's actual conclusion is not what Clark and Lappin (2011: 99) suggest it is.

3. What evidence exists that each stream of experience each child is exposed to is guaranteed to be generated by a fixed, computable process (i.e. computable probability distribution or primitive recursion function)? More generally, what evidence exists that the data presentations are a priori limited?

My contention is that we have plenty of evidence with respect to question (1), some evidence with respect to (2), and virtually no evidence with respect to (3).

Consider question (1). Although theoretical linguists and language typologists repeatedly observe an amazing amount of variation in the world's languages, there is consensus that there are limits to the variation, though stating exact universals is difficult (Greenberg 1963, 1978; Mairal and Gil 2006; Stabler 2009). Even language typologists who are suspicious of hypothesized language universals, once aware of the kinds of patterns that are logically possible, agree that not any logically possible pattern could be a natural language pattern.

Here is a simple example: many linguists have observed that languages do not appear to count past two (Berwick 1982, 1985; Heinz 2007, 2009). For example, no language requires sentences with at least $n \geq 3$ main constituents to have the nth one be a verb phrase (unlike verb-second languages like German). This is a logically possible language pattern. Here is another one: if an even number of adjectives modify a noun, then they follow the noun in noun phrases, but if an odd number of adjectives modify a noun they precede the noun in noun phrases. These are both recursive patterns; in fact, they are regular.

According to Chater and Vitànyi (2007), if the linguistic input a child received contained sufficiently many examples of noun phrases which obeyed the even–odd adjective rule above, they would learn it. It is an empirical hypothesis, but I think children would fail to learn this rule no matter how many examples they were given. Chater and Vitànyi can claim that there is a simpler pattern consistent with data (e.g. adjectives can optionally precede or follow nouns) that children settle on because their lives and childhoods are too short for there to be enough data to move from the simpler generalization to the correct one. This also leads to an interesting, unfortunately untestable, prediction, that if humans only had longer lives and childhoods, we could learn such bizarre patterns like the even–odd adjective rule. In other words, they might choose to explain the absence of an even–odd adjective rule in natural languages as just a byproduct of short lives and childhoods, whereas I would attribute it to linguistic principles which exclude it from the collection of hypotheses children entertain. But there is a way Chater and Vitànyi can address the issue: How much data does "the ideal language learner" require to converge to the unattested pattern?

The harder learning frameworks—identification in the limit from positive data and PAC—bring more insight into the problem of learning and the nature of learnable classes of patterns. First, these definitions of learning make clear that the central problem in learning is generalizing beyond one's experience. This is because under these definitions, generalizing to infinite patterns *requires* the impossibility of being able to learn certain finite patterns (Gold's first point above). I think humans behave like this.

Consider the birds in Table 27.3. If I tell you birds (a,b) are "warbler-barblers" and ask which other birds (c,d,e,f,g) are warbler-barblers you are likely to decide that birds (c,f,g) could be warbler-barblers but birds (d,e) definitely not. You would be very surprised to learn that in fact birds (a,b) are the only warbler-barblers of all time ever. Humans never even consider the possibility that there could just be exactly two "warbler-barblers." This insight is expressed well by Gleitman (1990: 12):

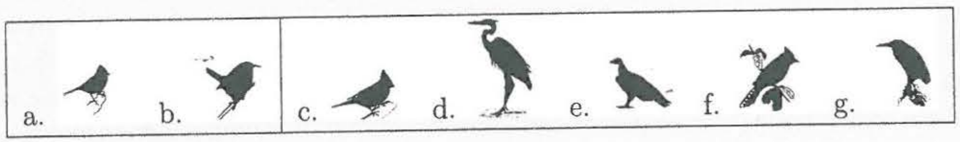
> The trouble is that an observer who notices *everything* can learn *nothing* for there is no end of categories known and constructible to describe a situation. (Gleitman 1990: 12; emphasis in original)

Chater and Vitànyi (2007) can say that grammars to describe finite languages are more complex than regular or context-free grammars, and they are right, provided the finite language is big enough. Again, the question is what kind experience does "the ideal language learner" need in order to learn a finite language with exactly n sentences, and is this human-like? This question should be asked of all proposed language learning models. It is interesting to contrast "the ideal language learner" with Yoshinaka's (2008, 2011) learners which generalize to context-free patterns (a^nb^n) and context-sensitive patterns ($a^nb^nc^n$) with at most a few examples (and so those learners cannot learn, under any circumstances, the finite language that contains only those few examples).

Second, classes which are learnable within these difficult frameworks have the potential to yield new kinds of insights about which properties natural languages possess which make them learnable. As discussed in section 27.4.2, there are many positive results of interesting subclasses of major regions of the Chomsky Hierarchy which are identifiable in the limit from positive data and/or PAC-learnable, and which describe natural language patterns. The learners for those classes succeed because of the structure inherent to the class—a structure which can reflect deep, universal properties of natural language. Under weaker definitions of learning, where the recursive class of patterns is learnable, such insights are less likely to be forthcoming.

Clark and Lappin (2011) have anticipated one way such insight could be forthcoming. I have mentioned that many of the learners which can learn the recursive class of languages in particular learning frameworks require more time and resources in the worst case than is considered to be reasonable. In chapter 7, Clark and Lappin (2011) provide excellent discussion on interpreting the computational complexity of learning algorithms

Table 27.3 Birds (a,b) are "warbler-barblers." Which birds (c–g) do you think are "warbler-barblers"?



themselves. As they point out, such infeasibility results provide "a starting point for research on a set of learning problems, rather than marking its conclusion" (2011: 148). This is because infeasible learning results always consider the worst-case. It may be that only some of the languages in a class require enormous time resources, but that others can be learned within reasonable time limits. Clark and Lappin explain that "to achieve interesting theoretical insight into the distinction between these cases, we need a more principled way of separating the hard problems from the easy ones" (2011: 148) which will "distinguish the learnable grammars from the intractable ones" (2011: 149). They go on to suggest that one possibility is to "construct algorithms for subsets of existing representation classes, such as context-free grammars" (2011: 149). In other words, in the learning frameworks where the entire recursive class is learnable, one way to proceed would to be find those subclasses which are *feasibly* learnable (recall Figure 27.9).

As for Gold's second point, there has been some empirical study into whether children use negative evidence in language acquisition (Brown and Hanlon 1970; Marcus 1993). Also learning frameworks which permit queries (Angluin 1988a, 1990), especially correction queries (Becerra-Bonache et al. 2006; Tîrnauca 2008), can be thought of as allowing learners to access implicit negative evidence.

As for the third question at the start of this section, I do not know of any research that has addressed it. It is a hypothesis that the universe is computable (and therefore all data presentations would be as well). It is not clear to me how this hypothesis could ever be tested.

Nonetheless, it should be clear that the commonly-cited statistical learning frameworks that have shown probabilistic context-free languages are learnable (Horning 1969), and in fact the recursive stochastic languages are learnable (Angluin 1988b; Chater and Vitànyi 2007) are pursuing Gold's third suggestion. It also ought to be clear that the positive results that show recursive patterns can be learned from positive, complete, and computable data presentations are "in principle" results. As far as is known, they cannot learn these classes feasibly. Of course, as Clark and Lappin (2011) suggest, it may be possible that such techniques can feasibly learn interesting *subclasses* of major regions of the Chomsky Hierarchy which are relevant to natural language. If shown, this would be an interesting complement to the research efforts pursuing Gold's first suggestion, and could also reveal universal properties of natural language that contribute to their learnability.

27.5.5 Summary

There are many ways to define learning and how best to define learning to study the acquistion of natural language remains an active area of research, which is unfortunately sometimes contentious. Nonetheless, I hope the discussion in this section has made clear that feasible learning can only occur when target classes of patterns are restricted and structured appropriately. I have emphasized that the central problem of learning is generalization. Also, I hope to have made clear that debates pitting statistical learning against symbolic learning have been largely misplaced. The real issue is about which data presentations learners should succeed on. Stochastic learning paradigms provide some benefit in learning power, but only when the class of presentations is limited to computable ones. Whether such paradigms are a step towards realism is debatable and not a fact to be taken for granted.

27.6 Artificial Language Learning Experiments

Some key questions raised in the last section can in principle be addressed by artificial language learning experiments. These experiments probe the generalizations people make on the basis of brief finite exposure to artificial languages (Gómez and Gerken 2000; Petersson et al. 2004; Folia et al. 2010). The performance of human subjects can then be compared to the performance of computational learning models on these experiments (Dell et al. 2000; Chambers et al. 2002; Onishi et al. 2003; Saffran and Thiessen 2003; Goldrick 2004; Wilson 2006; Frank et al. 2007; Finley and Badecker 2009; Seidl et al. 2009; Finley 2011; Jäger and Rogers 2012; Koo and Callahan 2012; Lai 2015).

But the relationship can go beyond comparison and evaluation to design. Well-defined learnable classes which contain natural language patterns are the bases for experiments. As mentioned, there are non-trivial interesting classes of languages which are PAC-learnable, which are identifiable in the limit from positive data, and which contain natural language patterns. The proofs are constructive and a common technique is identifying exactly the finite experience the proposed learners need to generalize correctly to each language in a given class. This critical finite experience is called the characteristic sample. The characteristic sample essentially becomes the training stimuli for the experiments. Other sentences in the language that are not part of the characteristic sample become test items. Finally, more than one learner can be compared by finding test items in the symmetric difference of the different patterns multiple learners return from the experimental stimuli. These points are also articulated by Rogers and Hauser (2010), and I encourage readers to study their paper.

Let me provide a simple example to illustrate. Consider the mildly context-sensitive pattern ($a^nb^nc^n$) which can learned in principle by both Chater and Vitànyi's (2007) ideal language learner and Yoshinaka's (2011) learner. However, each model requires a different amount of data to converge to this target. Which is more human-like? What about a mildly context-sensitive pattern outside the class of patterns learnable by Yoshinaka's learner? Such a pattern can be learned by Chater and Vitànyi's ideal language learner from some data presentation. Can humans replicate this feat? This is just the tip of the iceberg, and many such experiments are currently being conducted in many linguistic subfields including phonology, morphology, and syntax.

27.7 Conclusion

In this chapter I have tried to explain what computational learning theories are, and the lessons language scientists can draw from them. I believe there is a bright future for research which honestly integrates the insights of computational learning theories with the insights and methodologies of developmental psycholinguistics.

Acknowledgments

I thank Lorenzo Carlucci, John Case, Alexander Clark, and Katya Pertsova for helpful discussion and an anonymous reviewer for valuable feedback.

REFERENCES

Abney, Steven. 1996. Statistical methods and linguistics. In *The balancing act: Combining symbolic and statistical approaches to language*, edited by J.L. Klavans ad P. Resnik, 1–26. Cambridge, MA: MIT Press.

Angluin, D. 1988a. Queries and concept learning. *Machine Learning* 2:319–342.

Angluin, Dana. 1980. Inductive inference of formal languages from positive data. *Information Control* 45:117–135.

Angluin, Dana. 1982. Inference of reversible languages. *Journal for the Association of Computing Machinery* 29:741–765.

Angluin, Dana. 1988b. Identifying languages from stochastic examples. Tech. Rep. 614, Yale University, New Haven, CT.

Angluin, Dana. 1990. Negative results for equivalence queries. *Machine Learning* 5:121–150.

Angluin, Dana, and Philip Laird. 1988. Learning from noisy examples. *Machine Learning* 2:343–370.

Anthony, M., and N. Biggs. 1992. *Computational Learning Theory*. Cambridge University Press.

Applegate, R.B. 1972. Ineseño Chumash grammar. Doctoral dissertation, University of California, Berkeley.

Bach, Emmon. 1975. Long vowels and stress in Kwakiutl. *Texas Linguistic Forum* 2:9–19.

Bates, Elizabeth, and Jeffrey Elman. 1996. Learning rediscovered. *Science* 274:1849–1850.

Becerra-Bonache, Leonor, John Case, Sanjay Jain, and Frank Stephan. 2010. Iterative learning of simple external contextual languages. *Theoretical Computer Science* 411:2741–2756.

Becerra-Bonache, Leonor, Adrian Horia Dediu, and Cristina Tîrnauca. 2006. Learning DFA from correction and equivalence queries. In *ICGI*, vol. 4201 of *Lecture Notes in Computer Science*, 281–292. Springer.

Berwick, Robert. 1982. Locality principles and the acquisition of syntactic knowledge. Doctoral dissertation, MIT.

Berwick, Robert. 1985. *The acquisition of syntactic knowledge*. Cambridge, MA: MIT Press.

Blumer, Anselm, A. Ehrenfeucht, David Haussler, and Manfred K. Warmuth. 1989. Learnability and the Vapnik-Chervonenkis dimension. *J. ACM* 36:929–965.

Brown, R., and C. Hanlon. 1970. Derivational complexity and order of acquisition in child speech. In *Cognition and the developmental of language*, edited by J. Hayes, 11–53. New York: Wiley.

Carlucci, L., J. Case, S. Jain, and F. Stephan. 2004. U-shaped learning may be necessary. 37*th Annual Meeting of the Society for Mathematical Psychology*, Ann Arbor, Michigan, July 2004, abstract in *Journal of Mathematical Psychology*, 49(1):97, 2005.
Carlucci, L., J. Case, S. Jain, and F. Stephan. 2007. Memory-limited U-shaped learning. *Information and Computation* 205:1551–1573.

Case, J. 1999. The power of vacillation in language learning. *SIAM Journal on Computing* 28:1941–1969.

Chaitin, Gregory. 2004. How real are real numbers? ArXiv:math/0411418v3.

Chambers, Kyle E., Kristine H. Onishi, and Cynthia Fisher. 2002. Learning phonotactic constraints from brief auditory experience. *Cognition* 83:B13–B23.

Chater, Nick, and Paul Vitányi. 2007. 'Ideal learning' of natural language: Positive results about learning from positive evidence. *Journal of Mathematical Psychology* 51:135–163.

Chomsky, Noam. 1956. Three models for the description of language. *IRE Transactions on Information Theory* 113124. IT-2.

Clark, Alexander, François Coste, and Laurent Miclet, eds. 2008. *Grammatical Inference: Algorithms and Applications, 9th International Colloquium, ICGI 2008, Saint-Malo, France, September 22-24, 2008, Proceedings*, vol. 5278 of *Lecture Notes in Computer Science*. Springer.

Clark, Alexander, and Rémi Eyraud. 2007. Polynomial identification in the limit of substitutable context-free languages. *Journal of Machine Learning Research* 8:1725–1745.

Clark, Alexander, Rémi Eyraud, and Amaury Habrard. 2010. Using contextual representations to efficiently learn context-free languages. *Journal of Machine Learning Research* 11:2707–2744.

Clark, Alexander, and Shalom Lappin. 2011. *Linguistic Nativism and the Poverty of the Stimulus*. Wiley-Blackwell.

Clements, George, and Jay Keyser. 1983. *CV phonology: a generative theory of the syllable*. Cambridge, MA: MIT Press.

Dell, G. S., K. D. Reed, D. R. Adams, and A. S. Meyer. 2000. Speech errors, phonotactic constraints, and implicit learning: A study of the role of experience in language production. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 26:1355–1367.

Eisner, Jason. 1997. Efficient generation in primitive Optimality Theory. In *Proceedings of the 35th Annual ACL and 8th EACL*, 313–320. Madrid.

Elman, Jeffrey L., Elizabeth A. Bates, Mark H. Johnson, Annette Karmiloff-Smith, Domenico Parisi, and Kim Plunkett. 1996. *Rethinking Innateness: A Connectionist Perspective on Development*. Cambridge, MA: MIT Press and Bradford Book.

Fernau, Henning. 2003. Identification of function distinguishable languages. *Theoretical Computer Science* 290:1679–1711.

Finley, Sara. 2011. The privileged status of locality in consonant harmony. *Journal of Memory and Language* 65:74–83.

Finley, Sara, and William Badecker. 2009. Artificial language learning and feature-based generalization. *Journal of Memory and Language* 61:423–437.

Folia, Vasiliki, Julia Uddén, Meinou de Vries, Christian Forkstam, and Karl Magnus Petersson. 2010. Artificial language learning in adults and children. *Language Learning* 60:188–220. Supplement 2.

García, Pedro, and José Ruiz. 1996. Learning k-piecewise testable languages from positive data. In *Grammatical Interference: Learning Syntax from Sentences*, edited by Laurent Miclet and Colin de la Higuera, vol. 1147 of *Lecture Notes in Computer Science*, 203–210. Springer.

García, Pedro, and José Ruiz. 2004. Learning k-testable and k-piecewise testable languages from positive data. *Grammars* 7:125–140.

Garcia, Pedro, Enrique Vidal, and José Oncina. 1990. Learning locally testable languages in the strict sense. In *Proceedings of the Workshop on Algorithmic Learning Theory*, 325–338.

Garey, M. R., and D. S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman.

Gleitman, Lila. 1990. The structural sources of verb meanings. *Language Acquisition* 1:3–55.

Gold, E.M. 1967. Language identification in the limit. *Information and Control* 10:447–474.

Gold, E.M. 1978. Complexity of automata identification from given data. *Information and Control* 37:302–320.

Goldrick, Matthew. 2004. Phonological features and phonotactic constraints in speech production. *Journal of Memory and Language* 586–603.

Goldwater, Sharon, and Mark Johnson. 2003. Learning OT constraint rankings using a maximum entropy model. In *Proceedings of the Stockholm Workshop on Variation within Optimality Theory*, edited by Jennifer Spenader, Anders Eriksson, and Osten Dahl, 111–120.

Gómez, R.L., and L.A. Gerken. 2000. Infant artificial language learning and language acquisition. *Trends in Cognitive Sciences* 4:178–186.

Greenberg, Joseph. 1963. Some universals of grammar with particular reference to the order of meaningful elements. In *Universals of Language*, 73–113. Cambridge: MIT Press.

Greenberg, Joseph. 1978. Initial and final consonant sequences. In *Universals of Human Language: Volume 2, Phonology*, edited by Joseph Greenberg, 243–279. Stanford University Press.

Griffiths, T.L., C. Kemp, and J. B. Tenenbaum. 2008. Bayesian models of cognition. In *The Cambridge handbook of computational cognitive modeling*, edited by Ron Sun. Cambridge University Press.

Harrison, Michael A. 1978. *Introduction to Formal Language Theory*. Addison-Wesley Publishing Company.

Heinz, Jeffrey. 2007. The inductive learning of phonotactic patterns. Doctoral dissertation, University of California, Los Angeles.

Heinz, Jeffrey. 2008. Left-to-right and right-to-left iterative languages. In *Grammatical Inference: Algorithms and Applications, 9th International Colloquium*, edited by Alexander Clark, François Coste, and Lauren Miclet, vol. 5278 of *Lecture Notes in Computer Science*, 84–97. Springer.

Heinz, Jeffrey. 2009. On the role of locality in learning stress patterns. *Phonology* 26:303–351.

Heinz, Jeffrey. 2010a. Learning long-distance phonotactics. *Linguistic Inquiry* 41:623–661.

Heinz, Jeffrey. 2010b. String extension learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 897–906. Uppsala, Sweden: Association for Computational Linguistics.

Heinz, Jeffrey, and William Idsardi. 2011. Sentence and word complexity. *Science* 333:295–297.

de la Higuera, Colin. 1997. Characteristic sets for polynomial grammatical inference. *Machine Learning* 27:125–138.

de la Higuera, Colin. 2005. A bibliographical study of grammatical inference. *Pattern Recognition* 38:1332–1348.

de la Higuera, Colin. 2010. *Grammatical Inference: Learning Automata and Grammars*. Cambridge University Press.

Hopcroft, John, Rajeev Motwani, and Jeffrey Ullman. 1979. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley.

Hopcroft, John, Rajeev Motwani, and Jeffrey Ullman. 2001. *Introduction to Automata Theory, Languages, and Computation*. Boston, MA: Addison-Wesley.

Horning, J. J. 1969. A study of grammatical inference. Doctoral dissertation, Stanford University.

Jain, Sanjay, Daniel Osherson, James S. Royer, and Arun Sharma. 1999. *Systems That Learn: An Introduction to Learning Theory (Learning, Development and Conceptual Change)*. 2nd ed. The MIT Press.

Joshi, A. K. 1985. Tree-adjoining grammars: How much context sensitivity is required to provide reasonable structural descriptions? In *Natural Language Parsing*, edited by D. Dowty, L. Karttunen, and A. Zwicky, 206–250. Cambridge University Press.

Kaplan, Ronald, and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics* 20:331–378.

Karttunen, Lauri. 1998. The proper treatment of optimality in computational phonology. In *FSMNLP'98*, 1–12. International Workshop on Finite-State Methods in Natural Language Processing, Bilkent University, Ankara, Turkey.

Kasprzik, Anna, and Timo Kötzing. 2010. String extension learning using lattices. In *Proceedings of the 4th International Conference on Language and Automata Theory and Applications (LATA 2010)*, edited by Henning Fernau Adrian-Horia Dediu and Carlos Martín-Vide, vol. 6031 of *Lecture Notes in Computer Science*, 380–391. Trier, Germany: Springer.

Kearns, Michael, and Ming Li. 1993. Learning in the presence of malicious errors. *SIAM Journal of Computing* 22.

Kearns, Michael, and Umesh Vazirani. 1994. *An Introduction to Computational Learning Theory*. MIT Press.

Klein, D. 2005. The unsupervised learning of natural language. Doctoral dissertation, Stanford University.

Kobele, Gregory. 2006. Generating copies: An investigation into structural identity in language and grammar. Doctoral dissertation, University of California, Los Angeles.

Koo, H., and L. Callahan. 2012 Tier-adjacency is not a necessary condition for learning phonotactic dependencies. *Language and Cognitive Processes 27:1425–1432.*

Lange, Steffen, Thomas Zeugmann, and Sandra Zilles. 2008. Learning indexed families of recursive languages from positive data: A survey. *Theoretical Computer Science* 397:194–232.

Mairal, Ricardo, and Juana Gil, eds. 2006. *Linguistic Universals*. Cambridge University Press.
Manning, Christopher, and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press.

Marcus, Gary. 1993. Negative evidence in language acquisition. *Cognition* 46:53–85.

Michael C. Frank, Sharon Goldwater, Vikash Mansinghka, Tom Griffiths, and Joshua Tenenbaum. 2007. Modeling human performance on statistical word segmentation tasks. In *Proceedings of the 29th Annual Meeting of the Cognitive Science Society*.

Muggleton, Stephen. 1990. *Inductive Acquisition of Expert Knowledge*. Addison-Wesley.

Niyogi, Partha. 2006. *The Computational Nature of Language Learning and Evolution*. Cambridge, MA: MIT Press.

Niyogi, Partha, and Robert Berwick. 1996. A language learning model for finite parameter spaces. *Cognition* 61:161–193.

Nowak, Martin A., Natalia L. Komarova, and Partha Niyogi. 2002. Computational and evolutionary aspects of language. *Nature* 417:611–617.

Oates, Tim, Tom Armstrong, and Leonor Becerra Bonache. 2006. Inferring grammars for mildly context-sensitive languages in polynomial-time. In *Proceedings of the 8th International Colloquium on Grammatical Inference (ICGI)*, 137–147.

Onishi, Kristine H., Kyle E. Chambers, and Cynthia Fisher. 2003. Infants learn phonotactic regularities from brief auditory experience. *Cognition* 87:B69–B77.

Osherson, Daniel, Scott Weinstein, and Michael Stob. 1986. *Systems that Learn*. Cambridge, MA: MIT Press.

Otto, F. 1985. Classes of regular and context-free languages over countably infinite alphabets. *Discrete Applied Mathematics* 12:41–56.

Papadimitriou, Christon. 1994. *Computational Complexity*. Addison Wesley.

Partee, Barbara, Alice ter Meulen, and Robert Wall. 1993. *Mathematical Methods in Linguistics*. Dordrect, Boston, London: Kluwer Academic Publishers.

Perfors, Amy, Joshua B. Tenenbaum, Edward Gibson, and Terry Regier. 2010. How recursive is language. In *Recursion and Human Language*, edited by Harry van der Hulst, chap. 9, 159–175. Berlin, Germany: De Gruyter Mouton.

Petersson, K. M., C. Forkstam, and M. Ingvar. 2004. Artificial syntactic violations activate brocas region. *Cognitive Science* 28:383–407.

Pitt, Leonard. 1985. Probabilistic inductive inference. Doctoral dissertation, Yale University. Computer Science Department, TR-400.

Pitt, Leonard. 1989. Inductive inference, DFAs and computational complexity. In *Proceedings of the International Workshop on Analogical and Inductive Inference*, 18–44. Springer-Verlag. Lecture Notes in Artificial Intelligence (v. 397).

Rogers, Hartley. 1967. *Theory of Recursive Functions and Effective Computability*. McGraw Hill Book Company.

Rogers, James, and Marc Hauser. 2010. The use of formal languages in artificial language learning: a proposal for distinguishing the differences between human and nonhuman animal learners. In *Recursion and Human Language*, edited by Harry van der Hulst, chap. 12, 213–232. Berlin, Germany: De Gruyter Mouton.

Rumelhart, D. E., and J. L. McClelland. 1986. On learning the past tenses of English verbs. In *Parallel Distributed Processing, volume 2*, edited by J.L. McClelland and D. E. Rumelhart, 216–271. Cambridge MA: MIT Press.

Saffran, J. R., and E. D. Thiessen. 2003. Pattern induction by infant language learners. *Developmental Psychology* 39:484–494.

Seidl, A., A. Cristià, A. Bernard, and K. H. Onishi. 2009. Allophonic and phonemic contrasts in infants' learning of sound patterns. *Language Learning and Development* 5:191–202.

Shieber, Stuart. 1985. Evidence against the context-freeness of natural language. *Linguistics and Philosophy* 8:333–343.

Sipser, Michael. 1997. *Introduction to the Theory of Computation*. PWS Publishing Company.
Solomonoff, Ray J. 1978. Complexity-based induction systems: Comparisons and convergence theorems. *IEEE Transactions on Information Theory* 24:422–432.

Stabler, Edward P. 2009. Computational models of language universals. In *Language Universals*, edited by Christiansen, Collins, and Edelman, 200–223. Oxford: Oxford University Press.

Tesar, Bruce, and Paul Smolensky. 2000. *Learnability in Optimality Theory*. MIT Press.

Thomas, Wolfgang. 1997. Languages, automata, and logic. vol. 3, chap. 7. Springer.

Tîrnauca, Cristina. 2008. A note on the relationship between different types of correction queries. In Clark *et al.* (2008), 213–223.

Turing, Alan. 1937. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society* s2:230–265.

Valiant, L.G. 1984. A theory of the learnable. *Communications of the ACM* 27:1134–1142.

Vapnik, Vladimir. 1995. *The nature of statistical learning theory*. New York: Springer.

Vapnik, Vladimir. 1998. *Statistical Learning Theory*. New York: Wiley.

Wexler, Kenneth, and Peter Culicover. 1980. *Formal Principles of Language Acquisition*. MIT Press.

Wiehagen, R. 1977. Identification of formal languages. In *Mathematical Foundations of Computer Science*, edited by A.L. de Oliveira, vol. 53 of *Lecture Notes in Computer Science*, 571–579. New York: Springer-Verlag.

Wiehagen, R., R. Frievalds, and E. Kinber. 1984. On the power of probabilistic strategies in inductive inference. *Theoretical Computer Science* 28:111–133.

Wilson, Colin. 2006. Learning phonology with substantive bias: An experimental and computational study of velar palatalization. *Cognitive Science* 30:945–982.

Yoshinaka, Ryo. 2008. Identification in the limit of k, l -substitutable context-free languages. In Clark *et al.* (2008), 266–279.

Yoshinaka, Ryo. 2011. Efficient learning of multiple context-free languages with multidimensional substitutability from positive data. *Theoretical Computer Science* 412:1821–1831.

Zeugmann, Thomas, and Sandra Zilles. 2008. Learning recursive functions: A survey. *Theoretical Computer Science* 397:4–56.