Part II

Language Learning

Chapter 2

Language Acquisition – The Problem of Inductive Inference

An appropriate point to begin our whole narrative is to consider the problem of language acquisition. Children learn with seemingly effortless ease, the language of their parents and caretakers. Let us begin by considering the computational difficulty of the problem that children solve so routinely.

In order to get started, we will consider a language to be a set of sentences. Given a finite alphabet (we take this to be the lexicon) Σ, we denote by Σ^* the universe of all possible finite strings (sentences) in the usual way. A language then is simply a subset of Σ^* – a subset consisting of the well formed strings. We have considered in the previous chapter, several examples from natural languages like English, French, and Yiddish to illustrate how some strings (sentences) are in the language[1] while others are not. The underlying grammar of a language determines which sentences are acceptable and which are not. Later in this chapter, we will consider alternative and perhaps more general conceptions of language — these will not change the fundamental import of our discussion here.

[1]Note that in a formal sense, it is quite uncontroversial to speak of a language as a subset of Σ^*. What is potentially more problematic is the notion of a *natural* language like English. We will take the position that individuals have their own individual languages and these might differ from each other. These individual languages are all natural languages. If the members of a community have significant overlap in their languages so much so that mutual intelligibility is extremely high then one might label this shared language as "English" or "French" or "Chinese" as the case may be.

Consider a child born and raised in a homogeneous English speaking community. Such a child is exposed to a *finite* number of sentences as a result of interaction with its linguistic environment. Yet on the basis of this, the child is able to *generalize* and thus form and understand novel sentences it has not encountered before. It is this ability to infer (induce) the novel elements of a set that is the cornerstone of successful language learning. Let us consider the following idealized setting for such a phenomenon.

Imagine the community speaks a language $L_t \subset \Sigma^*$. This is the target language that the learner must identify or approximate in some sense. As a result of interaction with the community, the child learner ultimately has access to a sequence (in time) of sentences

$$s_1, s_2, s_3, s_4, \ldots, s_n, \ldots$$

where $s_i \in L_t$ is the ith example available to the learner. Suppose that the learner makes a guess about the target language L_t after each new sentence becomes available.

Consider the case when the learner has no prior knowledge about the nature of L_t. Suppose that the kth example has just been received and the learner now has to make a guess about the identity of the target. What should the learner do? All it knows is that the target contains s_1 through s_k – there are an infinite number of possible languages that contain s_1 through s_k. Any of them could be the target. How many of them should (can) the learner consider? What should the learner guess?

Such is the dilemma of the learning child when confronted with finite linguistic evidence over the course of language acquisition. A theoretical analysis of this situation will rapidly lead us to the conclusion that learning in the complete absence of prior information is impossible. But first, let us consider a framework within which we can meaningfully discuss the problem of learning and inductive inference and better understand the various issues that underlie effective learning.

2.1 A Framework for Learning

The canonical problem to which much of language learning can conceptually be reduced is that of identifying an unknown set on the basis of example sentences. The framework within which analysis of (see Osherson, Stob, Weinstein, 1986; Valiant, 1984; Niyogi, 1998; Wexler and Culicover, 1980;

Jain et al, 1998) such problems can usefully be conducted consists of the following components:

1. *Target Language* $L_t \in \mathcal{L}$ is a target language drawn from a class of possible target languages ($\mathcal{L}$). This is the language the learner must identify on the basis of examples.

2. *Example Sentences* $s_i \in L_t$ are drawn from the target language and presented to the learner. Here s_i is the ith such example sentence.

3. *Hypothesis Languages* $h \in \mathcal{H}$ drawn from a class of possible hypothesis languages that child learners construct on the basis of exposure to example sentences in the environment.

4. *Learning Algorithm* $\mathcal{A}$ is an effective procedure by which languages from $\mathcal{H}$ are chosen on the basis of example sentences received by the learner.

2.1.1 Remarks

At this point, some clarifying remarks are worthwhile.

1. *Languages and Grammars:* We have formulated the question of learning a language as essentially that of identifying a set. We have not so far specified the nature of these sets. We now put the restriction that these languages must be computable. Therefore, following the computability thesis of Church and Turing, we will consider only *recursively enumerable* (*r.e.*) languages. An *r.e.* language has a potentially infinite number of sentences and has a finite representation in terms of a Turing Machine or a Phrase Structure Grammar. Thus languages, machines, and grammars are formally equivalent ways (Harrison, 1978) of specifying the same set. In our notation, g typically refers to a grammar and L_g refers to the language generated by it. With slight abuse of notation, in much of what follows, we will refer to the sets in question as grammars or languages depending upon the context. The *r.e.* languages constitute an enumerable set and effective procedures exist to enumerate the grammars (machines) that generate them (Rogers, 1958; Hopcroft and Ullman, 1979). Since a grammar is a finite representation of the language, it is reasonable to suppose that language users and learners work with these finite representations rather than the infinite languages themselves. Since many different grammars may be compatible with the same language, this raises the question of intensional versus extensional

learning – a notion that is captured in the distinction between I-language and E-language (Chomsky, 1986) that is worthwhile to keep in mind as the book develops.

Thus in a formal sense, we really have a collection $\mathcal{G}$ of possible target grammars and $\mathcal{L}$ is then defined as

$$\mathcal{L} = \{L_g | g \in \mathcal{G}\}$$

It is finally worth noting that all computable Phrase Structure Grammars may be enumerated as $g_1, g_2, \ldots$. Given any *r.e.* language L there are an infinite number of g_i's such that $L_{g_i} = L$. Then any collection of grammars may be defined by specifying their indices in an acceptable enumeration.

2. *Example Sentences:* Sentences will be presented to the learner one at a time. One might imagine many different modes of interaction with the environment as a result of which such sentences become available. However, it is worthwhile to note that the ability of children to learn a language does not seem to be sensitive to the precise order in which sentences are presented to them. (Newport, Gleitman, and Gleitman, 1977; Schieffelin and Eisenberg, 1981). Hence, in our treatment of learnability we will mostly require that a psychologically plausible learning algorithm can be shown to converge to the target in a manner that is independent of the order of presentation of sentences. In much of this book, we will almost exclusively consider examples presented in i.i.d. fashion[2] according to a fixed, underlying, probability distribution μ. The distribution μ characterizes the relative frequency

[2]In reality, linguistic experience is conducted with clear dependencies and correlations between successive sentences. This dependence is based on considerations of pragmatics and discourse and difficult to model precisely. However, it is possible that such dependencies affect the semantic content of sentences but not their syntactic structure. For example, the sentences "John ate an apple" and "Bill moved the car" have different lexical choices and correspondingly different semantic content. They may occur as part of different conversations with different probability distributions of the precise sentences that follow it. However, both have similar syntactic structure and are of the form **Subject-Verb-Object** (SVO). It may be that if sentences are viewed as strings over syntactic categories, the i.i.d assumption is a believable one. Further, it may also be that while there are immediate dependencies between sentences based on discourse, there are no long range dependencies and the i.i.d. assumption is like sampling from the stationary distribution of the corresponding Markov process. In any event, the probabilistic assumption of i.i.d. sentences is used as a convenient mathematical device to get a handle on the fact that different syntactic forms occur with different frequencies. The precise consequences of such an assumption may then be understood opening the path to understanding more complicated phenomena. This is one of the many abstractions we make in order to be able to take first steps in reasoning about what is otherwise a very complex situation.

of different kinds of sentences that children are likely to encounter during language acquisition. For example, they are more likely to hear shorter sentences than substantially embedded longer sentences. This distribution μ might have support over all of Σ^* in which case both positive (sentences in the target language) as well as negative (sentences not in the target language) examples will be presented to the learner. On the other hand, μ might have support only over L_t in which case only positive examples are presented to the learner. This latter case is psychologically the more real as considerable evidence seems to exist suggesting that children do not have much exposure to negative examples over their learning period (Brown and Hanlon, 1970; Hirsh-Pasek, Treiman, and Schneiderman, 1984; Demetras, Post, and Snow, 1986). [3]

3. *The learning algorithm* $\mathcal{A}$ is an effective procedure allowing the learning child to construct hypotheses about the identity of the target language on the basis of the examples it has received. In principle, any learner, be it the child learner over the course of language acquisition, or a machine learner, has to follow a procedure or algorithm and is therefore subject to the computational laws that govern such processes. In particular, following the Church Turing thesis, we will accept the equivalence of partial recursive functions and effective procedures, and therefore consider learning algorithms to be mappings from the set of all finite data streams to hypotheses in $\mathcal{H}$. A particular finite data stream of k example sentences may be denoted as $(s_1, s_2, \ldots, s_k)$. Let $\mathcal{D}^k = \{(s_1, \ldots, s_k) | s_i \in \Sigma^*\} = (\Sigma^*)^k$ be the set of all possible sequences of k example sentences that the learner might potentially receive. Then we can define

$$\mathcal{D} = \cup_{k>0} \mathcal{D}^k$$

to be the set of all finite data sequences. Since $\mathcal{D}^k$ is enumerable, so is $\mathcal{D}$ and we can then let $\mathcal{A}$ to be a partial recursive function

$$\mathcal{A} : \mathcal{D} \rightarrow \mathcal{H}$$

where $\mathcal{H}$ is the enumerable set of hypothesis grammars[4] (languages). Given a data stream $t \in \mathcal{D}$, the learner's hypothesis is given by $\mathcal{A}(t)$ and is an

[3] Much of the time, members of the adult community simply produce sentences in their language giving the child exposure to positive examples. The only source of negative examples therefore comes from mistakes that children make during language acquisition and the feedback from this is often absent, impoverished, or misleading.

[4] A language is a set with a potentially infinite number of sentences. However, they have finite representations in terms of grammars. It is reasonable therefore to postulate

element of $\mathcal{H}$. In much of this book, our treatment of learning will largely be in a probabilistic setting with natural ties to statistical learning theory as developed in Vapnik (1982, 1998) or Valiant (1984) as well as probabilistic settings of inductive inference in the extended Gold tradition (Jain et al, 1998). In this we will deviate from our strict formulation of the learner as a deterministic procedure to consider probabilistic learners that are allowed to flip a coin to choose hypotheses that are sometimes elements of $\mathcal{H}$ and sometimes subsets of $\mathcal{H}$. An important thing to note is that the behavior of the learner for a particular data stream $(s_1, \ldots, s_k) \in \mathcal{D}^k$ is independent of the target language or languages from which the data is drawn. It depends only on the data stream and can be predicted either deterministically or probabilistically if the learning algorithm is analyzable.

Some kinds of learning procedures are worth introducing here and we will return to them at several points over the course of this book. A *consistent* learner always maintains a hypothesis (call it h_n after n examples) that is consistent with the entire data set it has received so far. Thus, if the data set the learner has received is denoted by $(s_1, \ldots, s_n)$, then the learner's grammatical hypothesis h_n is such that each $s_i \in L_{h_n}$. An *empirical risk minimizing* learner uses the following procedure

$$h_n = \arg\min_{h \in \mathcal{H}} R(h; (s_1, \ldots, s_n))$$

The risk function $R(h, (s_1 \ldots, s_n))$ measures the fit to the empirical data consisting of the example sentences $(s_1, \ldots, s_n)$. In many cases, this minimization might not be unique in which case the learner will need a further criterion to decide which of the minima should be picked as a hypothesis language. Some kind of Occam principle is natural to consider here. For example, the learner might conjecture the smallest or simplest grammar that fits the data well.[5] A *memoryless* learning algorithm is one whose hypothesis

that human knowledge of a language has a compact encoding in terms of a grammar. Correspondingly, learners therefore conjecture (develop) grammars as they attempt to learn a language. The set $\mathcal{H}$ is the hypothesis set of possible grammars they may conjecture (develop) in the course of learning a language. The map $\mathcal{A}$ from linguistic experience $\mathcal{D}$ to grammatical hypotheses $\mathcal{H}$ may be viewed as the language learning map, the language development map, or the language growth map depending on one's point of view.

[5] This idea finds a clear instantiation in the Minimum Description Length principle (MDL) of Rissanen and its application to language learning as in Rissanen and Ristad (1992), de Marcken (1996), Brent and Cartwright (1996), Brent (1999) and Goldsmith (2001). Earlier treatments of this idea are to be found in the evaluation metric of Chomsky (1965) and complexity functions of Feldman (1972).

at every point depends only on the current data and the previous hypothesis. Let $t_n = (s_1, \ldots, s_n) \in \mathcal{D}$ be a data set with n examples and let t_{n-1} be the first $n-1$ examples (i.e., $t_{n-1} = (s_1, \ldots, s_{n-1})$) of this data set. Then $\mathcal{A}$ is such that $\mathcal{A}(t_n)$ depends only upon $\mathcal{A}(t_{n-1})$ and s_n. Learning *by enumeration* is another common strategy. Here the learner enumerates all possible grammars in $\mathcal{H}$ in some order. Let this enumeration be $h_1, h_2, \ldots$. It then begins with the conjecture h_1. Upon receiving new example sentences, the learner simply goes down this list and updates its conjecture to the first one that is consistent with the data seen so far. Variants to this basic idea may be considered.

4. *Criterion of Success:* A significant component of the learning framework involves a criterion of success so that we can measure how well the learner has learned at any point in the learning process. This takes the form of a distance measure d so that for any target grammar g_t and any hypothesis grammar h, one can define the distance between the target grammar and the hypothesis grammar as $d(g_t, h)$. If h_n is the learner's hypothesis after n sentences have been received, then *learnability* implies that

$$\lim_{n \to \infty} d(g_t, h_n) = 0$$

In other words, the learner's hypothesis converges to the target *in the limit*. If the learning algorithm $\mathcal{A}$ is such that for every possible target grammar $g_t \in \mathcal{G}$, the learner's hypothesis converges to it (when presented with a data sequence from the target grammar), then the family $\mathcal{G}$ is said to be *learnable* by $\mathcal{A}$. By varying d, and by varying whether the convergence needs to be in a probabilistic sense or not, different convergence criteria may be obtained. We will consider a few different ones in the treatment that follows.

Two related notions might be introduced here. We use the term *generalization error* to refer to the quantity $d(g_t, h_n)$ that measures the distance of the learner's hypothesis (after n examples) to the target. Learnability implies that the generalization error eventually converges to zero as the number of examples goes to infinity. In a statistical setting, the generalization error is a random variable and can only converge to zero in a probabilistic sense (Vapnik, 1982; Valiant, 1984).

4. *Generality of the framework:* It is worthwhile to emphasize that the basic framework presented here for the analysis of learning systems is quite general. The target and hypothesis classes $\mathcal{L}$ and $\mathcal{H}$ might consist of grammars in a generative linguistics tradition. Example of such grammars include those in the tradition of Government and Binding (Chomsky, 1981); Head

Driven Phrase Structure Grammar (HPSG; Pollard and Sag, 1994); Lexical Functional Grammar (LFG; Kaplan and Bresnan, 1982; Bresnan, 2001), Autolexical Grammar (Sadock, 1991) and so on. They might consist of general grammatical families such as finite state grammars (DFAs), context free grammars (CFGs), tree adjoining grammars (TAGs; Joshi, Levy, and Takahashi, 1975) and so on. They might consist of grammars in a connectionist tradition. We do not commit ourselves to any representational issues here but choose a generic enumeration scheme to enumerate the grammars in the class[6]. Again, no commitment is made just yet to learning algorithms – they could in principle be grammatical inference procedures, gradient descent schemes like those occurring in connectionist learning, Minimum Description Length (MDL) learning, maximum likelihood learning via the EM algorithm and so on. Different research traditions use different grammatical families, different representations and correspondingly different learning algorithms. Most of these are analyzable in the framework considered here.

In what follows, we always consider the case in which the hypothesis class $\mathcal{H}$ is equal to the target class $\mathcal{G}$. In other words, $\mathcal{L}_{\mathcal{H}} = \{L_h | h \in \mathcal{H}\} = \mathcal{L}$. If this were not the case, for some languages (those in $\mathcal{L} \setminus \mathcal{L}_{\mathcal{H}}$) the learner could never converge to the target because it could never hypothesize the target language (grammar). Such languages could never come to be spoken by humans and therefore would not exist as natural languages.

2.2 The Inductive Inference Approach

The first study of this problem of identifying sets was conducted in the sixties with the pioneering work of Gold (1967). Later work by Feldman (1972), Blum and Blum (1975), Angluin(1980,1988), Osherson, Stob, and Weinstein (1986), Pitt (1989), Gasarch, Smith (Gasarch and Smith, 1992), Jain et al (1998) and a host of others elaborate on this theme. Jain et al (1998) provides an excellent and updated exposition of the various technical results in this area. In this section, we provide a brief introduction to the paradigm of identification in the limit and a proof of Gold's celebrated result. We then proceed to interpret this result and its extensions in the context of the natural phenomena of language acquisition.
First, we need to define a few terms.

[6]The fact that grammars may be enumerated follows from the computability thesis we have adopted here.

Definition 1 *A text t for a language L is an infinite sequence of examples $s_1, \ldots, s_n, \ldots$ such that (i) each $s_i \in L$ (ii) every element of L appears at least once in t. We denote by $t(k)$ the kth element of the text. This is simply s_k. We denote by t_k the first k elements of the text. This is simply $s_1 \ldots s_k$. Thus $t_k \in \mathcal{D}^k$ (if represented as a k-tuple).*

Let us now define the notion of learnability in the limit for this framework.

Definition 2 *Fix a distance[7] metric d, a target grammar $g_t \in \mathcal{G}$ and a text t for the target language (L_{g_t}). The learning algorithm $\mathcal{A}$ identifies (learns) the target $g_t(L_{g_t})$ on the text t in the limit if*

$$\lim_{k \to \infty} d(\mathcal{A}(t_k), g_t) = 0$$

If the target grammar g_t is identified (learned) by $\mathcal{A}$ for all texts of L_{g_t}, the learning algorithm is said to identify (learn) g_t in the limit. If all grammars in $\mathcal{G}$ can be identified (learned) in the limit, then the class of grammars $\mathcal{G}$ (correspondingly class of languages $\mathcal{L}$) is said to be identifiable (learnable) in the limit. Thus learnability is equivalent to identification in the limit.

Now consider the case where $\mathcal{G} = \mathcal{H}$. After some more notational niceties, we will prove the following fundamental result.

[7]The choice of the distance metric d allows us to consider many different notions of convergence to a target grammar. It is worth noting that any two grammars g and h define corresponding sets of expressions (languages) L_g and L_h. However, the metric d is defined on the space of grammars and may incorporate both extensional and intensional terms. For example, a purely extensional notion of distance would be one in which $d(g, h)$ depends only on L_g and L_h and nothing else. In that case, for all grammars f such that $L_f = L_g$ we would have $d(g, h) = d(f, h)$. If i_g and i_h are the indices of g and h in some acceptable enumeration of the grammars, then a purely intensional notion of distance could be $d(g, h) = |i_g - i_h|$. In some sense, both the specific choices discussed above are unsatisfactory. In fact, from a cognitive perspective, there is a long tradition of thought in generative grammar which attributes some psychological reality to grammars. A child's developing linguistic knowledge is codified, represented and interpreted in terms of its grammar. Consequently, some grammars may be interpretable while others may not. Therefore, the child's convergence to the parent's grammar ought to include considerations of grammatical representation. On the other hand, since the index of a grammar may bear no natural relationship to the extensional set identified as the corresponding language, a distance measure like $d(g, h) = |i_g - i_h|$ might end up ignoring extensional agreement altogether resulting in an absurdity. In much of our discussion in this chapter, we will discuss learning based on extensional criteria to understand fundamental limitations of inductive inference. In subsequent chapters, we will take a more linguistic point of view and consider intensional criteria in models of language acquisition.

Definition 3 *Given a finite sequence $x = s_1, s_2, \ldots, s_n$ (of length n), we denote the length of the sequence by $lh(x) = n$. For such a sequence x, we denote by $x \subseteq L$ the fact that each s_i in x is contained in the language L. We denote by $range(x)$ the set of all unique elements of x. Thus $x \subseteq L$ is shorthand for the more set-theoretically correct statement $range(x) \subseteq L$. The concatenation of two sequences $x = x_1, \ldots, x_n$ and $y = y_1, \ldots, y_m$ is denoted by $x \circ y = x_1, \ldots, x_n, y_1, \ldots, y_m$.*

Now we are in a position to state

Theorem 1 (after Blum and Blum; ϵ-version) *If $\mathcal{A}$ identifies a target grammar g in the limit, then, for every $\epsilon > 0$, there exists a locking data set $l_\epsilon \in \mathcal{D}$ such that (i) $l_\epsilon \subseteq L_g$ (ii) $d(\mathcal{A}(l_\epsilon), g) < \epsilon$, and (iii) $d(\mathcal{A}(l_\epsilon \circ \sigma), g) < \epsilon$ for all $\sigma \in \mathcal{D}$ where $\sigma \subseteq L_g$. In other words, after encountering the locking data set, the learner will be ϵ-close to the target as long as it continues to be given sentences from the target language.*

Proof: We prove by contradiction. Suppose no locking data set exists. Then for every $l \in \mathcal{D}$ such that $l \subseteq L_g$ and $d(\mathcal{A}(l), g) < \epsilon$, there must exist some $\sigma_l \in \mathcal{D}$ (where $\sigma_l \subseteq L_g$) that has the property $d(\mathcal{A}(l \circ \sigma_l), g) \geq \epsilon$. We will use this fact to construct a text for L_g on which $\mathcal{A}$ will not identify the target. Begin with a text $r = s_1, s_2, \ldots, s_n, \ldots$ for L_g. Now construct a new text q in the following manner. Let $q^{(1)} = s_1$. If $d(\mathcal{A}(q^{(i)}), L_g) < \epsilon$, then we pick $\sigma_{q^{(i)}}$ that violates the locking property and update the text by letting $q^{(i+1)} = q^{(i)} \circ \sigma_{q^{(i)}} \circ s_{i+1}$. If $d(\mathcal{A}(q^{(i)}), L_g) \geq \epsilon$, then we simply let $q^{(i+1)} = q^{(i)} \circ s_{i+1}$. Since an s_i is added at each stage of the text creation process, it is clear that q is a valid text. At the same time, it is clear that $\mathcal{A}$ can never converge to g for the text q. This is because every time it conjectures a grammar h such that $d(h, g) < \epsilon$ (say at $q^{(j)}$), it is forced to conjecture some other grammar that is not in the ϵ-neighborhood of g by the time it reaches $q^{(j)} \circ \sigma_{q^{(j)}}$. In other words, the learner's conjectures on q are such that $d(\mathcal{A}(q_i), g) \geq \epsilon$ infinitely often. ■

This suggests that if a grammar g (correspondingly, a language L_g) is identifiable (learnable) in the limit, a locking data set exists which "locks" the learner's conjectures to within an ϵ-ball of the target grammar after encountering this locking data set. In what follows, we will consider the important and classical case of exactly identifying the target language in the limit. Here the distance metric is $0-1$-valued and given by $d(g_1, g_2) = 1$ if and only if $L_{g_1} = L_{g_2}$. Putting $\epsilon = \frac{1}{2}$ in the previous theorem, we get the following classical result

Theorem 2 (Blum and Blum, 1975) *If $\mathcal{A}$ identifies a target grammar g in the limit, then, there exists a locking data set $l \in \mathcal{D}$ such that (i) $l \subseteq L_g$ (ii) $d(\mathcal{A}(l), g) = 0$, and (iii) $d(\mathcal{A}(l \circ \sigma), g) = 0$ for all $\sigma \in \mathcal{D}$ where $\sigma \subseteq L_g$.*

Utilizing this result, we can now prove Gold's famous theorem.

Theorem 3 (Gold, 1967) *If the family $\mathcal{L}$ consists of all the finite languages and at least one infinite language, then it is not learnable (identifiable) in the limit.*

Proof: We prove by contradiction. Suppose that an algorithm $\mathcal{A}$ is able to identify the family $\mathcal{L}$. Therefore, in particular, it is able to identify the infinite language (call it L_{inf}). Therefore, by theorem 2, a finite locking data sequence must exist (call it σ_{inf}). Consider the language $L = range(\sigma_{\text{inf}})$. This is a finite language and therefore is in $\mathcal{L}$. Furthermore, one can construct a text t for this language such that $t_k = \sigma_{\text{inf}}$ where $k = lh(\sigma_{\text{inf}})$. However, by the locking property, for such a text t, the algorithm $\mathcal{A}$ converges to L_{inf}. Therefore, $\mathcal{A}$ does not identify L and therefore does not identify $\mathcal{L}$. ■

Since both regular and context free languages contain all the finite languages and many infinite ones, it immediately follows as a corollary that

Corollary 1 *The family of languages represented by (i) Deterministic Finite Automata and (ii) Context Free Grammars are not identifiable in the limit.*

Thus, all grammatical families in the core Chomsky hierarchy of grammars are unlearnable in this sense. Before we examine the implications of this result, it is worthwhile to consider a formulation of the necessary and sufficient conditions for the learnability of a family of languages. The proof of Gold's result presented here uses the notion of locking sequences that was introduced by Blum and Blum some years after Gold's original paper. The use of such locking sequences in the proof illustrates how they hold the key to learnability of language families. This is indeed the case as the following theorem shows:

Theorem 4 (Angluin, 1980) *The family $\mathcal{L}$ is learnable (identifiable) if and only if for each $L \in \mathcal{L}$, there is a subset D_L such that if $L' \in \mathcal{L}$ contains D_L, then L' is not a proper subset of L.*

Proof: First we prove the necessary part. Since $\mathcal{L}$ is learnable, therefore, an algorithm $\mathcal{A}$ exists and for every $L \in \mathcal{L}$ a locking data sequence σ_L must exist. Consider the case of a particular language L and its locking data sequence σ_L. If there is some other language $L' \in \mathcal{L}$ such that (i)$range(\sigma_L) \subseteq L'$ and (ii) $L' \subset L$ then it is possible to construct a text t for L' such that $t_k = \sigma_L$ where $k = lh(\sigma_L)$. On such a text, because of the locking property, the algorithm $\mathcal{A}$ will be such that $d(\mathcal{A}(t_j), L) \rightarrow 0$ and since $L' \subset L$, we have that $d(\mathcal{A}(t_j), L') \not\rightarrow 0$. Therefore L' and consequently the family $\mathcal{L}$ is not identifiable in the limit.

Now we prove the sufficient part. Suppose that for each L, there exists a $D_L \subseteq L$ such that if $D_L \subseteq L'$ for some $L' \in \mathcal{L}$, then $L' \not\subset L$. We now show how to construct an algorithm $\mathcal{A}$ to learn the family $\mathcal{L}$.

Let $L_1, L_2, \ldots$, be an enumeration of the languages in $\mathcal{L}$ where for each L_i there exists a D_{L_i} as discussed. Consider the following learning algorithm. On an input data stream $\sigma \in \mathcal{D}$ such that $lh(\sigma) = k$, the learner searches for the least index $i \leq k$ such that $D_{L_i} \subseteq range(\sigma) \subseteq L_i$. If no such i exists, the learner simply conjectures L_1 as a default. We need to prove that this algorithm will identify every language in $\mathcal{L}$.

Let the target language be L_j. Consider a text t for L_j. After $k \geq j$ example sentences have been encountered, the learner could potentially conjecture L_j. The only reason it may not is if it conjectures some L_i where $i < j$. We claim that for every $i < j$, the learner will either not conjecture L_i at all or eventually abandon it for good. To see this, consider some L_i $(i < j)$. There are two cases to consider:
Case I: $L_j \setminus L_i = \emptyset$. This means that $L_j \subset L_i$. But then D_{L_i} cannot be a subset of L_j. Since $range(t_k) \subseteq L_j$ for all k, it can never be the case that $D_{L_i} \subseteq range(t_k) \subseteq L_i$. The learner can never conjecture L_i at any point.
Case II: $L_j \setminus L_i \neq \emptyset$. This means there is some sentence $s \in L_j$ that is not in L_i. Eventually this sentence must occur in t_k for some k. For all $n > k$, therefore it can never be the case that $range(t_n) \subseteq L_i$. Thus the learner will never conjecture L_i after this point. ■

2.2.1 Discussion

In the light of these results, let us now take stock of the situation.
Remark 1. We have considered here the difficulty of inferring a set from examples of members of this set. The problem is clearly motivated by the inference problem children have to solve in the process of learning a language. A particular language may be viewed as a set of well-formed expressions.

The nature of this set is unknown to the child before language acquisition. In syntax, for example, this refers to the set of well-formed sentences (or phrases or constructions depending upon one's point of view); in phonology, these refer to the set of well-formed phonological forms. Every such set has a finite representation in terms of an underlying grammar. Crucially, children are not exposed directly to the grammar – they are exposed only to the expressions[8] of their language and their task is to learn the grammar that provides a compact encoding of the ambient language they are exposed to.

Remark 2. It is worth noting that the precise notion of convergence depends upon the distance metric d imposed on the space of grammars. We have considered in some of the above theorems the case in which the metric $d(g_1, g_2)$ is $0-1$ valued and depends only on whether $L_{g_1} = L_{g_2}$. In this metric the learner may converge on the correct extensional set but may not converge to the correct grammar. In fact the learner is not required to converge to any single grammar at all. Some would argue that this notion of convergence may be *behaviorally* plausible but *cognitively* implausible. Taking the view that grammars have a cognitive status and therefore that the maturation of the child's linguistic knowledge must be captured in the development of its grammar, one may impose the stronger metric where $d(g_i, g_j) = |i - j|$ or $|C(i) - C(j)|$ where i, j are indices of the grammars and $C(i)$ and $C(j)$ are measures of grammatical complexity in some sense. This notion of convergence is significantly more stringent and much less is learnable in this case. In much of this chapter, we will review results using the more relaxed extensional (behavioral) notion of convergence and appreciate the impossibility of tabula rasa learning even in this setting. It is finally worth noting that a related notion of convergence was originally proposed by Gold (1967) where it was required that the learner stabilize (converge) to some grammar that was extensionally correct. This may be viewed as a compromise between the two extremes that have just been outlined.

Remark 3. It may be argued that the proper role of language is to mediate a mapping between form and meaning or symbol and referent in the Saussurean (Saussure, 1983) sense. When one takes this point of view, one reduces a language to a mapping from Σ_1^* to Σ_2^* where Σ_1^* is the set of all linguistic expressions (over some alphabet Σ_1) and Σ_2^* is a characterization of the set of all possible meanings. In this framework, a language L is regarded as a subset of $\Sigma_1^* \times \Sigma_2^*$ – in other words a potentially infinite set

[8]See Lightfoot, 1991 for a discussion on learning from cues or phrasal fragments that seem to have a status somewhere in between an expression and a grammatical rule.

of (form,meaning) pairs or associations. There are, in fact, many possible formalizations of the notion of a language for our purposes. We enumerate below a list of possibilities.

1. $L \subset \Sigma^*$ – this is the central and traditional notion of a language both in computer science and linguistics.

2. $L \subset \Sigma_1^* \times \Sigma_2^*$ – a subset of form-meaning pairs and in a formal sense no different from notion 1.

3. $L : \Sigma^* \rightarrow [0,1]$ – a language maps every expression to a real number between 0 and 1. For any expression $s \in \Sigma^*$, the number $L(s)$ characterizes the degree of well-formedness of that expression with $L(s) = 1$ denoting perfect grammaticality and $L(s) = 0$ denoting complete lack of it. Note that notion 1 simply considers languages to realize mappings from Σ^* to $\{0,1\}$ – thus sentences are either grammatical or ungrammatical with no notion of graded grammaticality. The support of this function may be a restricted subset of Σ^* – this restricted subset may be regarded to be a language in sense of notion 1.

4. L is a probability distribution μ on Σ^* – this is the usual notion of a language in statistical language modeling.

5. L is a probability distribution μ on $\Sigma_1^* \times \Sigma_2^*$.

It is worthwhile to observe that each of the extended notions of language makes the learning problem for the child harder rather than easier. Thus the non-learnability results discussed earlier have a significance greater than the particular formal context in which they have been developed.

Remark 4. The arguments presented above suggest that if the space $\mathcal{L} = \mathcal{H}$ of possible languages is too large, then the family is no longer learnable. As a matter of fact, taking the result of Gold (1967), we see that even the space of regular languages (DFAs) is too large for learnability by this account. It is also worth emphasizing that the arguments are of an informational rather than computational nature. In other words, we consider $\mathcal{A}$ to be a mapping from $\mathcal{D}$ to $\mathcal{H}$. Even if this mapping were *not* computable, the negative results about learnability presented here would still stand.

Remark 5. One may think that the unlearnability of regular languages is due to the fact that all the finite sets are included in this family. Since, we know that natural languages are never finite, one may ask what sort of learnability results would hold for families of languages $\mathcal{L}$ where each member of $\mathcal{L}$ was

required to be infinite. For example, is the class of infinite regular languages unlearnable? While Gold's Theorem does not apply to this case, it is worth noting that the proof technique does carry over. Consider two languages L_1 and L_2 such that $L_1 \subset L_2$ and further $L_2 \setminus L_1$ is an infinite set. Clearly one may find two such languages in $\mathcal{L} = \{$ infinite regular languages $\}$. Let σ_2 be a locking sequence for L_2. Then clearly, $L_1 \cup range(\sigma_2)$ is a language that contains σ_2, is a proper subset of L_2 and is contained in $\mathcal{L}$. One sees that this language will not be learnable from a text whose prefix is σ_2 as the learner will lock on to L_2 on such a text.

Remark 6. The most compelling objection to the classical inductive inference paradigm comes from statistical quarters. It seems unreasonable to expect the learner to *exactly* identify the target on *every single* text. The natural framework that modifies both assumptions is the so called *Probably Approximately Correct* learning framework (Valiant, 1984) that tries to learn the target approximately with high probability. We discuss this in the next section but it is worthwhile to note here that the PAC model also emphasizes identification in the limit – the quantity $d(g_t, h_n)$ is now a random variable that must converge to 0 – not on every single data sequence as in in the classical Gold style inductive inference framework, but with probability 1 (weak convergence of random variables).

Before we consider the PAC approach, let us first review some additional results in more stochastic formulations that have existed in the inductive inference framework.

2.2.2 Additional Results

In the classical framework of the previous section, no assumption was made about the source that generated the text. Let us assume that the text is generated in i.i.d. fashion according to a probability measure μ on the sentences of the target language L. In much of this book, we will adopt this assumption in order to derive probabilistic bounds on the performance of the language learner that will be critical in deriving our models of language change.

The measure μ has support on L. Therefore, one may define corresponding measures on the product spaces. Thus, we have measure μ_2 on the product space $L \times L$. All texts t from the language L that have been generated according to i.i.d. draws from μ will be such that $t_2 \in L \times L$. Similarly, we can define the measure μ_3 on the space $L \times L \times L$ and so on. By the Kolmogorov extension theorem, a unique measure μ_∞ is guaranteed to exist

on the set $T = \prod_{i=1}^{\infty} L_i$ (where $L_i = L$ for each i). The set T consists of all texts that may be generated from L by i.i.d. draws according to μ, i.e., $t \in T$ is such that $t(k) \in L$ for all k and each $t(k)$ is drawn in i.i.d. fashion according to μ. The measure μ_∞ is defined on T and thus we have a measure on the set of all texts.

Theorem 5 *Let $\mathcal{A}$ be an arbitrary learning algorithm and g be an arbitrary (not necessarily target) grammar. Then the set of texts on which the learning algorithm $\mathcal{A}$ converges to g is measurable.*

Proof: Consider the set

$$A = \{t | \lim_{k \to \infty} d(\mathcal{A}(t_k), g) = 0\}$$

This is the set of all texts on which the learning algorithm converges to the grammar g. To see that this is measurable, it is enough to see that for each k, l, the set $B_{k,l}$ (defined as below) is measurable.

$$B_{k,l} = \{t | d(\mathcal{A}(t_k), g) < \frac{1}{l}\}$$

As a matter of fact, $\mu_\infty(B_{k,l}) = \mu_k(\{T \in L^k \mid d(\mathcal{A}(T), g) < \frac{1}{l}\})$. Now let

$$C_l = \cup_i \cap_{m>i} B_{m,l}$$

Thus C_l is simply the set of texts for which the learning algorithm eventually makes conjectures that lie within a $\frac{1}{l}$ ball of the grammar g. By the usual properties of sigma algebras, C_l is clearly measurable. Finally, we see that

$$A = \cap_{l=1}^{\infty} C_l$$

is measurable too. ■

As a result, it is possible to define learning with measure 1 in the following way.

Definition 4 *Let g be a target grammar and texts be presented to the learner in i.i.d. fashion according to a probability measure μ on L_g. If there exists a learning algorithm $\mathcal{A}$ such that $\mu_\infty(\{t | \lim_{k \to \infty} d(\mathcal{A}(t_k), g) = 0\}) = 1$ then the target is said to be learnable with measure 1. The family $\mathcal{G}$ is said to be learnable with measure 1 if all grammars in $\mathcal{G}$ are learnable with measure 1 by some algorithm $\mathcal{A}$.*

According to this notion of learnability, it is worthwhile to note that

1. If the measure μ is known in a certain sense, the entire family of *r.e.* sets becomes learnable with measure 1. We shall see a proof of this shortly.

2. On the other hand, if μ is unknown, the Superfinite languages (those having all the finite languages and at least one infinite language) are not learnable. Thus, the class of learnable languages is not enlarged for distribution free learning in a stochastic setting.

3. Computable distributions make languages learnable. Thus any collection of computable distributions is identifiable in the limit. This is comparable to Gold (1967) where it is shown that if examples are constrained to be produced by effective procedures (called computable presentations) the class of learnable languages is significantly enhanced.

Let us consider the set of r.e. languages and let $L_1, L_2, L_3, \ldots$ be an enumeration of them. Let measure μ_i be associated with language L_i. Thus μ_i has support on L_i and if L_i is to be the target language, then examples are drawn in i.i.d. fashion according to this measure. As discussed above, by the extension theorem a natural measure $\mu_{i,\infty}$ exists on the set of texts for L_i. Then it is possible to prove that

Theorem 6 *With strong prior knowledge about the nature of the μ_i's, the family $\mathcal{L}$ of r.e. languages is measure one learnable.*

Proof: The proof follows that in Osherson, Stob, and Weinstein (1986). Let $s_1, s_2, \ldots$ be an enumeration of all finite strings (sentences) in Σ^*. For an example sequence $\sigma \in \mathcal{D}$, let us say that σ agrees with L_j through n if for each s_i $(i \leq n)$, we have $s_i \in L_j \Leftrightarrow s_i \in range(\sigma)$. In other words, σ and L_j agree on the membership of the first n sentences of Σ^*.

Now let us first introduce the set

$$A_{j,n,m} = \{t | t \text{ is a text for } L_j \text{ and } \exists i \leq n | s_i \in L_j \setminus range(t_m)\}$$

Thus $A_{j,n,m}$ is the set of all texts for L_j such that one of the first n elements of Σ^* is in L_j but does not appear in t_m. It is easy to see that $A_{j,n,m+1} \subseteq A_{j,n,m}$. Therefore $\mu_{j,\infty}(A_{j,n,m})$ is a monotonically decreasing function of m. As a matter of fact, since $\cap_{m=1}^{\infty} A_{j,n,m} = \phi$, we have

$$\lim_{m \to \infty} \mu_{j,\infty}(A_{j,n,m}) = 0$$

for every fixed j, n.

Next we define the function $d(n)$ as

$d(n)$ = least m such that $\mu_{i,\infty}(A_{i,n,m}) \leq 2^{-n}$ for all $i \leq n$.

In other words, if one fixes n, then after seeing at least $d(n)$ examples, one is guaranteed that if the target were one of L_1 through L_n, with high probability one would get the membership values for each of the sentences $s_1 \in \Sigma^*$ through $s_n \in \Sigma^*$. It is also clear that $d(n)$ is a monotonically increasing function of n. Further, as n grows, one would eventually establish the membership of every sentence and thus identify the target language with measure 1.

The learning algorithm would work as follows. On an input sequence $\sigma \in \mathcal{D}$, let $m = lh(\sigma)$. Then, the learning algorithm finds the largest n such that (i) $n \leq m$ and (ii) $d(n) \leq m$. We will indicate such an n by $d^{-1}(m)$. It is therefore appropriate to conjecture one of L_1 through L_n. Let j be the least integer such that $j \leq n$ and L_j agrees with σ through n. If no such j exists, then let $j = 1$. The algorithm conjectures L_j on input sequence σ.

Now we need to prove that the algorithm learns the target grammar with measure 1. Let the target language be such that k is the least index for it. Consider now the set of texts on which the learning algorithm does not converge to L_k. This is given by

$$B = \{t | \mathcal{A}(t_n) \neq k \text{ for infinitely many } n\}$$

We will show that the measure of B is 0. To see this consider the following intermediate set X_k given by

$$X_k = \cap_i \cup_{m>i} A_{k,d^{-1}(m),m}$$

Now,

Claim 1: $B \subseteq X_k$.

To see this, consider a $t \in B$. This means that on t, we have $\mathcal{A}(t_m) \neq L_k$ infinitely often. There are two reasons why $\mathcal{A}(t_m)$ might not be equal to L_k. They are (i) t_m and L_k do not agree through $d^{-1}(m)$ or (ii) there exists some L_i $(i < k)$ such that L_i and t_m agree through $d^{-1}(m)$. However, for any such L_i, since t_m and L_i must eventually disagree for some m, no such L_i can be conjectured infinitely often. Therefore the only reason that something other than L_k is conjectured infinitely often is that t_m and L_k must not agree through $d^{-1}(m)$ for infinitely many m's. Therefore,

$$t \in X_k = \cap_i \cup_{m>i} A_{k,d^{-1}(m),m}$$

Claim 2: $\cap_i \cup_{m>i} A_{k,d^{-1}(m),m} \subseteq \cap_i \cup_{n>i} A_{k,n,d(n)}$.
To see this, assume $t \in \cap_i \cup_{m>i} A_{k,d^{-1}(m),m}$. Now we need to show that $t \in \cap_i \cup_{n>i} A_{k,n,d(n)}$. To show the latter, we need to show that for every i, there exists some $n > i$ such that $t \in A_{k,n,d(n)}$. However, since $t \in \cap_i \cup_{m>i} A_{k,d^{-1}(m),m}$, therefore $t \in A_{k,d^{-1}(m),m}$ for some $m > d(i+1)$. Let $n = d^{-1}(m)$ for such an m. Clearly, $n \geq i+1$. Therefore, $t \in A_{k,n,m}$ where $d(n) \leq m$ and $n \geq i+1$. Since $A_{k,n,l+1} \subseteq A_{k,n,l}$ for all l, we clearly have $t \in A_{k,n,d(n)}$ for some $n > i$.
Finally notice that since $\sum_n \mu_{i,\infty}(A_{k,n,d(n)}) < \infty$, by the Borel Cantelli lemma, and Claims 1 and 2, $\mu_{k,\infty}(X_k) = 0$. The set of texts on which the learning algorithm does not converge has measure zero. ■

It might appear that by simply changing the requirement from learning on all texts to learning on almost all texts, the class of learnable languages is significantly enlarged. This is however misleading since the measure one learnability of the above theorem requires one to know $d(n)$. This is a *very strong* assumption indeed. In particular, if the learning algorithm works for one particular set of measures $\{\mu_i\}$, it is very easy to perturb the source measures so as to ensure non-learnability. Therefore, a more natural requirement as one moves to a probabilistic framework is to require learnability in a distribution-free sense.

Definition 5 *Consider a target grammar g and a text stochastically presented by i.i.d. draws from the target language L_g according to a measure μ. If a learning algorithm exists that can learn the target grammar with measure one for all measures μ then g is said to be learnable in a distribution free sense. A family of grammars $\mathcal{G}$ is learnable in a distribution free sense if there exists a learning algorithm that can learn every grammar in the family with measure one in a distribution free sense.*

It is worthwhile to note that when one considers statistical learning, the distribution free requirement is the natural one and all statistical estimation algorithms worth their salt are required to converge in a distribution free sense. When this restriction is imposed, the class of learnable families is not enlarged. In particular, it is possible to prove

Theorem 7 (Angluin,1988) *If a family of grammars $\mathcal{G}$ is learnable with measure one (on almost all texts) in a distribution free sense, then it is learnable in the limit in the Gold sense (on all texts).*

As a matter of fact, one can prove the even stronger theorem

Theorem 8 (Pitt,1989) *If a family of grammars $\mathcal{G}$ is learnable with measure $p > \frac{1}{2}$ then it is learnable in the limit in the Gold sense.*

This immediately implies of course that regular languages are not measure one learnable in a distribution free sense. We will soon turn our attention to a model of learning that requires only weak convergence of the learner to the target. Here we will only require that

$$\lim_{k\to\infty} \mathbb{P}[d(\mathcal{A}(t_k), g) > \epsilon] = 0$$

leading to the well known PAC (Probably Approximately Correct) model of learning.

Before we do so, however, it is worthwhile to mention a few positive results on the learning of grammatical families that are worth keeping in mind for a more complete and nuanced understanding of the possibilities and limitations of learning and inductive inference.

1. We see that given a family of languages $\mathcal{L}$, if the learning algorithm knows the source distribution of each of the languages, then the family is learnable with measure one. If, on the other hand, the text is presented stochastically from some unknown distribution, the family is not learnable. One might frame the question of language learning as essentially identifying (learning in some sense) a measure μ from a collection of measures $\mathcal{M}$. This reduces to a density estimation problem which in principle is harder than function approximation (or set identification). Indeed, if no constraints are put on the family $\mathcal{M}$, then identifying the target measure is not possible. It is reasonable to ask under what conditions the family $\mathcal{M}$ becomes identifiable in the limit. One answer to this question was provided by Angluin (1988) where it was proved that a *uniformly computable* distribution could be identified in the limit in a certain sense. Let $\mathcal{M} = \{\mu_0, \mu_1, \mu_2 \dots\}$ be a computable (hence enumerable) family of distributions. Define the distance between two distributions μ_i and μ_j as

   $$d(\mu_i, \mu_j) = \sup_{x\in\Sigma^*} |\mu_i(x) - \mu_j(x)|$$

   This is the L_∞ norm on sequences. The family $\mathcal{M}$ is said to be *uniformly computable* if there exists a total recursive function $f(i, x, \epsilon)$ such that for every i, for every $x \in \Sigma^*$, and for every ϵ, $f(i, x, \epsilon)$ outputs a rational number p such that

   $$|\mu_i(x) - p| < \epsilon$$

The learner receives a text probabilistically drawn according to an unknown target measure from the family $\mathcal{M}$. After k examples are received, the learning algorithm guesses $\mathcal{A}(t_k) \in \mathcal{M}$. It is possible to construct a learning algorithm that has the property

$$\lim_{k \to \infty} d(\mathcal{A}(t_k), \mu_j) = 0$$

Special cases of this result are the learnability of stochastic finite state grammars (van der Mude and Walker, 1978) and stochastic context free grammars (Horning, 1969). It is worth noting that uniform computability is a very strong prior constraint on the set of all distributions. For example, in the context of stochastic context free grammars, one obtains probability measures on context free languages by tying probabilities to context free rules. As a result, the probability distributions are always such that longer strings become *exponentially* less likely. An arbitrary collection of probability measures with support on context free languages need not be uniformly computable and so need not be learnable.

2. A second class of positive results arise from *active learning* on the part of the learner. Here the learner is allowed to make queries about the membership of arbitrary elements $x \in \Sigma^*$ (membership queries). This allows the regular languages to be learned in polynomial time though context free grammars remain unlearnable (Angluin, 1987; Angluin and Kharitonov, 1995). Other query-based models of learning with varying degrees of psychological reality have also been considered. They enlarge the family of learnable languages but none allow all languages to be learnable (Gasarch and Smith, 1992). It is certainly reasonable to consider the possibility that children explore the environment and this active exploration facilitates learning and circumvents some of the intractability inherent in inductive inference. On the other hand, the ability to make arbitrary membership queries seems to be too strong and it is likely that the learning child possesses this ability only to a limited extent.

3. The problem of inference is seen to be difficult because the learner is required to succeed on all or almost all texts. It is natural to consider further restrictions on the set of texts on which the learner is required to succeed. These are as follows:

(a) *Recursive texts:* A text t is said to be recursive if $\{t_n | n \in \mathbb{N}\}$ is recursive. The learner is required to converge to the target language for all recursive texts that correspond to this language. It is possible to show there exists a map $\mathcal{A}$ (from data sets to grammars) such that all phrase structure grammars are learnable. Unfortunately this map is not computable, and the following theorem (see Jain et al, 1998) holds: If a computable map exists that can learn a family of languages $\mathcal{L}$ from recursive texts, then $\mathcal{L}$ is algorithmically learnable from all texts. This result implies that restricting learnability to recursive texts does not enlarge the family of learnable languages.

(b) *Ascending texts:* A text t is said to be ascending if for all $n < m$, the length of $t(n)$ is less than or equal to the length of $t(m)$, i.e., sentences are presented in increasing order of length. It is possible to show that there are language families $\mathcal{L}$ that are learnable from ascending texts but not learnable from all texts. Superfinite families (i.e., families containing all the finite languages and at least one infinite language) however, remain unlearnable in this setting.

(c) *Informant texts:* A text t for a language L is said to be an informant if it consists of both positive and negative examples. Every element of Σ^* appears in the text with a label indicating whether it belongs to the target language or not. All recursively enumerable sets are learnable from informant texts. The general consensus in empirical studies of language acquisition seems to be that children are hardly ever exposed to negative examples. While it is true that there may be ways to get indirect evidence of negative examples, it still seems unlikely that the learning child ever gets an opportunity to sample the space of negative examples with enough coverage to get an unbiased estimate of the target language.

4. One may consider weaker convergence criteria. For example, an overly weak convergence criterion is to put a metric on the family of languages defined by $d(L_1, L_2) = \sum_{s \in \Sigma^*} \mu(s) |1_{L_1}(s) - 1_{L_2}(s)|$ (where μ is a fixed measure on Σ^*) and define convergence in this norm. This strategy (Wharton, 1974) leads to the unfortunate consequence that the finite languages become dense in the space of r.e. languages and therefore a learning procedure need only output finite languages in order to learn successfully. A more satisfactory weakening is provided by the framework of anomalies where one is required to learn the target

language upto (at most) k mistakes. Gold identification corresponds to $k = 0$ and it is possible to show (Case and Smith, 1983) that a proper hierarchy of learnable families is obtained as k varies. Yet another criterion is the notion of strongly approaching (Feldman, 1972) where the learner must eventually be dislodged from all incorrect hypotheses and supply a correct hypothesis infinitely often.

5. One may consider various ways to incorporate structure into the learning problem leading to learnability results. Examples include learning context free grammars from structured examples (Sakakibara, 1990) or more recently the work on learning categorial grammars (Kanazawa, 1998). If one were able to provide a cognitively plausible justification for how such structures were made available to the learning child, then such approaches would provide a natural framework for structured learning of linguistic families.

We have tried in the previous sections to provide the central developments and results of the theory of inductive inference that continues to provide the basic formal framework to reason about language acquisition. Some caveats notwithstanding, the main implication of these results is that learning in the complete absence of any prior information is infeasible. Through the various sections we have tried to provide the reader with a feel for some of the prior knowledge that is used to prove the technical results.

2.3 The Probably Approximately Correct Model and the VC Theorem

Another significant approach to learning theory is the decidedly statistical route pursued by a large number of researchers in computer science and statistics. The central theoretical framework for such an approach was pioneered by Vapnik and Chervonenkis (1971) and elaborated fully in Vapnik (1998). In the context of computer science, this work was introduced with additional computational complexity considerations by Valiant (1984) as the PAC (Probably Approximately Correct) Model that has stimulated a rich dialog between computer science and statistics over the last two decades.

The canonical problem in statistical learning theory is the learning of functions. The concept class and the hypothesis class are classes of functions $f : X \to Y$ where X and Y are arbitrary sets. The learner is required

to converge (identify, learn) to the target in the limit. However, this convergence is probabilistic as we now clarify.

2.3.1 Sets and Indicator Functions

In the canonical framework of statistical learning theory, the class of possible target functions (usually referred to as the *concept class* in the PAC literature) — $\mathcal{F}$ and the hypothesis class $\mathcal{H}$ are both classes of functions $f : X \to Y$. In the case of language, it is natural to consider X to be the set Σ^* – the set of all possible strings, and Y to be the set $\{0, 1\}$. Therefore for a particular language $L \subset \Sigma^*$, we can define the indicator function for it as

$$1_L(x) : \Sigma^* \to \{0, 1\}$$

where $1_L(x) = 1$ if and only if $x \in L$. Thus, identifying or learning a language is equivalent to learning the indicator function corresponding to that language.

Languages now have three natural representations (i) as recursively enumerable subsets of Σ^* (ii) as Turing machines or programming systems or phrase structure grammars (iii) as indicator functions over Σ^*.

In our discussions so far, we have always taken $\mathcal{F} = \mathcal{H}$ and we will continue with this assumption.

2.3.2 Graded Distance

The discussion on language learning in the inductive inference framework is dominated by the notion of exact identification where the distance measure $d(1_L, 1_{L'}) = 1$ if and only if $L = L'$ and $= 0$ otherwise. It may reasonably be argued that such a distance does not allow for a natural graded topology on the space of possible languages. Therefore, we rectify this by considering the $L_1(P)$ topology on the space of languages as follows.

Define a probability measure P on Σ^*. Then, the $L_1(P)$ distance between two languages L and L' might be defined as

$$d(L, L') = \sum_{s \in \Sigma^*} |1_L(s) - 1_{L'}(s)| P(s)$$

Given any language L, we can therefore naturally define the ϵ-neighborhood of the language as

$$N_L(\epsilon) = \{L' | d(L, L') < \epsilon\}$$

This allows us to consider languages that are arbitrarily close to each other and potentially alleviate the apparent pathologies introduced by the notion of exact identification in the limit.

2.3.3 Examples and Learnability

We assume that examples are randomly presented to the learner according to a probability distribution P on Σ^*. In the generic framework of function learning, the learner is presented with both *positive* and *negative* examples. Undoubtedly, the task of learning the target function with balanced (positive and negative) examples is easier than learning from positive examples alone. While the latter is the more natural setting for language acquisition, let us first develop the basic insights of statistical learning theory in the context of the easier function learning problem in order to understand some essential constraints on inductive inference from finite data.

Let L be the target language. Thus, $1_L : \Sigma^* \rightarrow \{0,1\}$ is the target function corresponding to this language. Examples are (x, y) pairs where $x \in \Sigma^*$ (drawn according to P) and $y = 1_L(x)$.

On the basis of these examples, the learner hypothesizes functions in $\mathcal{H}$ (recall $\mathcal{H} : \Sigma^* \rightarrow \{0,1\}$). As before, the learner is a mapping from possible data sets to the hypothesis class. In view of the fact that positive and negative examples are received, we will need to redefine D_k – the set of all data streams of length k – to be

$$D_k = \{(z_1, \ldots, z_k) \mid z_i = (x_i, y_i); x_i \in \Sigma^*, y_i \in \{0,1\}\}$$

After receiving l data points, the learner conjectures a function $\hat{h_l} \in \mathcal{H}$. The most natural statistical learning procedure to consider is one that minimizes empirical risk. According to this procedure, the learner's hypothesis $\hat{h_l}$ is chosen as

$$\hat{h_l} = \arg\min_{h \in \mathcal{H}} \frac{1}{l} \sum_{i=1}^{l} |y_i - h(x_i)|$$

We have indicated the learner's hypothesis by $\hat{h_l}$ where the ˆsymbol explicitly denotes that the learner's hypothesis $\hat{h_l}$ is a random function. This is simply because the learner's hypothesis depends upon the randomly generated data.

More generally, a learning algorithm $\mathcal{A}$ is an effective procedure mapping data sets to hypotheses, i.e.,

$$\mathcal{A} : \cup_{i=1}^{\infty} D_k \longrightarrow \mathcal{H}$$

Therefore, $\hat{h_l} = \mathcal{A}(d_l)$ where d_l is a random element of D_l. Successful learning would require the learner's hypothesis to converge to the target as the number of data points goes to infinity. Because the learner's hypothesis is a random function, it is now natural to consider this convergence in probability. Hence, we can define the following:

Definition 6 *The learner's hypothesis $\hat{h_l}$ converges to the target (1_L) with probability 1, if and only if for every $\epsilon > 0$*

$$\lim_{l \to \infty} \mathbb{P}[d(\hat{h_l}, 1_L) > \epsilon] = 0$$

Some remarks are worthwhile. First, note that the probability distribution P does double duty for us. On the one hand, it allows us to define the distance between languages as the $L_1(P)$ distance between their corresponding indicator functions. Convergence is therefore measured in this norm. On the other hand, it also provides the distribution with which data is drawn and presented to the learner. It therefore characterizes the probabilistic behavior of the random function $\hat{h_l}$. Crucially, however, P is unknown to the learner except through the random draws of examples.

Second, note that the notion of convergence is *exactly* the notion of weak convergence of a random variable. In the standard case of inductive inference treated earlier (for a text t) the distance $d(\mathcal{A}(t_k), L)$ is a deterministic sequence that must converge to zero for learnability. Now, the text t is generated randomly in i.i.d fashion. Therefore, $d(\mathcal{A}(t_k), L)$ is a random variable that is required to converge to zero with probability one.

This notion of weak convergence is usually stated in an (ϵ, δ) style in PAC formulations of learning theory in computer science communities. If $\hat{h_l}$ converges to the target 1_L in a weak sense, it follows that for every $\epsilon > 0$ and every $\delta > 0$, there exists an $m(\epsilon, \delta)$ such that for all $l > m(\epsilon, \delta)$, we have

$$\mathbb{P}[d(\hat{h_l}, 1_L) > \epsilon] < \delta$$

In other words, *with high probability* ($> 1 - \delta$), the learner's hypothesis ($\hat{h_l}$) is *approximately close* (within an ϵ in the appropriate norm) to the target language. The quantity $m(\epsilon, \delta)$ is usually referred to as the sample complexity of learning. Finally, we are able to define the notion of learnability within this PAC framework.

Definition 7 *Consider a target language L. If there exists a learning algorithm $\mathcal{A}$ such that for any distribution P on Σ^* according to which examples*

($(x, 1_L(x))$ pairs) are drawn and presented to $\mathcal{A}$, the learner's hypothesis converges to the target with probability 1, the target language L is said to be learnable. A family of languages $\mathcal{L}$ is said to be learnable if there exists a learning algorithm $\mathcal{A}$ which can learn every language in the family.

2.3.4 The Vapnik-Chervonenkis (VC) Theorem

It is natural now to consider what classes $\mathcal{L}$ are learnable under this new framework of learnability. The most fundamental characterization of learnability was provided by the pioneering work of Vapnik and Chervonenkis (1971). We provide a treatment of their work in the current context of language learning.

Let $\mathcal{L}$ be a collection of languages and $\mathcal{H}$ be the associated collection of indicator functions on Σ^*. Thus $\mathcal{H} = \{1_L | L \in \mathcal{L}\}$. Now let the target language be $L_t \in \mathcal{L}$. Correspondingly, the target function is $1_{L_t} \in \mathcal{H}$.

Note that

$$L_t = \arg\min_{L \in \mathcal{L}} E[|1_{L_t} - 1_L|]$$

Within the framework of empirical risk minimization, the learner's hypothesis is given by

$$\hat{L}_l = \arg\min_{L \in \mathcal{L}} \frac{1}{l} \sum_{i=1}^{l} |y_i - 1_L(x)|$$

The language $\hat{L}_l$ empirically chosen by the learner corresponds to an indicator function $1_{\hat{L}_l}$ from the class $\mathcal{H}$ of hypothesis functions on Σ^*. According to our previously denoted notation $\hat{h}_l = 1_{\hat{L}_l}$. Recall, however, that the learner in general need not be an empirical risk minimizing learner. The following theorem (also developed in Blumer et al, 1986) applies in complete generality irrespective of the learning algorithm used.

Theorem 9 (Vapnik and Chervonenkis, 1971,1991) *Let $\mathcal{L}$ be a collection of languages and $\mathcal{H}$ be the corresponding collection of functions. Then $\mathcal{L}$ is learnable if and only if $\mathcal{H}$ has finite VC dimension.*

In order to make sense of this theorem, we need to define the VC dimension of the family $\mathcal{H}$ of functions. We first develop the notion of *shattering.*

Definition 8 *A set of points $x_1, \ldots, x_n$ is said to be shattered by $\mathcal{H}$ if for every set of binary vectors $\mathbf{b} = (b_1, b_2, \ldots, b_n)$, there exists a function $h_{\mathbf{b}} \in \mathcal{H}$ such that $h_{\mathbf{b}}(x_i) = 1 \Leftrightarrow b_i = 1$.*

In other words, for every different way of partitioning the set of n points into two classes, a function (a different function for every different partition) in $\mathcal{H}$ is able to implement the partition. Obviously, $\mathcal{H}$ must have at least 2^n different functions in it.

Definition 9 *The VC dimension of a set of functions $\mathcal{H}$ is d if there exists at least one set of cardinality d that can be shattered and no set of cardinality greater than d can be shattered by $\mathcal{H}$. If no such finite d exists, the VC dimension is said to be infinite.*

Theorem 9 provides a completely different characterization of learnable families from those that have emerged in our previous treatments. Before we examine its consequences for language learning, let us consider a proof of the necessity of finite VC dimension for learnability to hold.

2.3.5 Proof of Lower Bound for Learning

Recall that learnability requires that for every $\epsilon > 0$ and $\delta > 0$, there exists a finite $m(\epsilon, \delta)$ such that if the learner draws $m > m(\epsilon, \delta)$ examples, then

$$\mathbb{P}[d(\hat{h_m}, h) > \epsilon] < \delta$$

for all $h \in \mathcal{H}$ and for all distributions P on Σ^* according which instances are provided.

We will now show that if $\mathcal{H}$ has VC dimension $= d$, then

$$m(\epsilon, \delta) > \frac{d}{4}\log_2(\frac{3}{2}) + \log_2(\frac{1}{8\delta})$$

Therefore it immediately follows that if $\mathcal{H}$ has infinite VC dimension, then $m(\epsilon, \delta)$ cannot be finite. Consequently, the class $\mathcal{H}$ is unlearnable.

Preliminaries:

Assume $\mathcal{H}$ has VC dimension $= d$. We now construct a probability distribution on Σ^* that will force the learner to draw at least $m > m(\epsilon, \delta) > \frac{d}{4}\log_2(\frac{3}{2}) + \log_2(\frac{1}{8\delta})$ examples to learn some target function in $\mathcal{H}$ to ϵ-accuracy with confidence greater than $1 - \delta$.

Since the VC dimension is d, there must exist a set of points $x_1, x_2, \ldots, x_d$ with each $x_i \in \Sigma^*$ such that these points can be shattered. Consider a probability distribution P that puts measure $\frac{1}{d}$ on each of these points and zero measure on the rest of the elements of Σ^*. Let $X = \{x_1, \ldots, x_d\}$.

According to this probability distribution P, any two functions $h_1 \in \mathcal{H}$ and $h_2 \in \mathcal{H}$ will have $d(h_1, h_2) = 0$ if and only if h_1 and h_2 agree on each of the x_i's. We may consider h_1 to be equivalent to h_2 if $d(h_1, h_2) = 0$. It is easy to check that this is an equivalence relation and there are exactly 2^d different equivalence classes in $\mathcal{H}$. Accordingly, we will not distinguish between different members of the same equivalence class and for the rest of the proof, it is sufficient to assume that $|\mathcal{H}| = 2^d$.

Step 1:

Let $\mathbf{z}$ be a random draw of m i.i.d. examples from Σ^* according to P and $\mathbf{z}_h$ be the labelled data set obtained by labelling the points according to the function $h \in \mathcal{H}$. Then $\mathbf{z}_h \in D_m$ and $\mathcal{A}(\mathbf{z}_h)$ is the learner's hypothesis on receiving this labelled data set.

Imagine that $\mathbf{z}$ contained exactly l distinct elements of X, i.e., the other $d - l$ elements did not occur in the data set. There are 2^l different ways in which the l instances in $\mathbf{z}$ may be labelled by a potential target function h. Let $\mathcal{H}_i \subset \mathcal{H}$ be the collection of functions that label these instances according to the ith labelling scheme. Thus we see that $\mathcal{H}_1$ through $\mathcal{H}_{2^l}$ are a disjoint partitioning of $\mathcal{H}$. Consider the sum

$$\sum_{h \in \mathcal{H}} d(\mathcal{A}(\mathbf{z}_h), h) = \sum_{i=1}^{2^l} \sum_{h \in \mathcal{H}_i} d(\mathcal{A}(\mathbf{z}_h), h) \tag{2.1}$$

Note that for each $\mathcal{H}_i$, there are exactly 2^{d-l} different functions in it. These functions agree on the l distinct instances found in $\mathbf{z}$. Therefore, for each $h \in \mathcal{H}_i$, the data set $\mathbf{z}_h$ is the same. On the remaining $d - l$ instances that have not been seen, these functions label them in each of 2^{d-l} different ways. Consider the quantity $d(\mathcal{A}(\mathbf{z}_h), h)$. On the $d - l$ instances that have not been seen, let $\mathcal{A}(\mathbf{z}_h)$ and h disagree on j instances. In that case, we have

$$d(\mathcal{A}(\mathbf{z}_h), h) \geq \frac{j}{d}$$

There are $\binom{d-l}{j}$ different ways in which h and $\mathcal{A}(\mathbf{z}_h)$ might disagree with each other on exactly j of the unseen $d - l$ instances. Therefore, we have

$$\sum_{h \in \mathcal{H}_i} d(\mathcal{A}(\mathbf{z}_h), h) \geq \sum_{j=0}^{d-l} \binom{d-l}{j} \frac{j}{d} \geq \frac{2^{d-l}}{d} \frac{d-l}{2} \tag{2.2}$$

Combining 2.1 and 2.2, we have

$$\sum_{h \in \mathcal{H}} d(\mathcal{A}(\mathbf{z}_h), h) \geq \frac{2^d}{d} \frac{d-l}{2}$$

Step 2:
Let

$$S = \{\mathbf{z} | \mathbf{z} \text{ has } l \text{ distinct elements } \}$$

Then we have

$$\sum_{\mathbf{z} \in S} P(\mathbf{z}) \frac{1}{2^d} \sum_{h \in \mathcal{H}} d(\mathcal{A}(\mathbf{z}_h), h) \geq \frac{1}{d} \frac{d-l}{2} P(S)$$

where $P(\mathbf{z})$ denotes the probability of drawing the instance set $\mathbf{z}$ and $P(S)$ denotes the total probability of drawing instances in the set S. Changing the order of sums, we see

$$\frac{1}{2^d} \sum_{h \in \mathcal{H}} \sum_{\mathbf{z} \in S} P(\mathbf{z}) d(\mathcal{A}(\mathbf{z}_h), h) \geq \frac{1}{d} \frac{d-l}{2} P(S)$$

from which it is clear that there exists at least one $h_* \in \mathcal{H}$ such that

$$\sum_{\mathbf{z} \in S} P(\mathbf{z}) d(\mathcal{A}(\mathbf{z}_{h_*}), h_*) \geq \frac{1}{d} \frac{d-l}{2} P(S)$$

Step 3:
We see that h_* is a candidate target function for which the learner's hypothesis is potentially inaccurate quite often. Consider the set

$$S_\beta = \{\mathbf{z} \in S | d(\mathcal{A}(\mathbf{z}_{h_*}), h_*) > \beta\}$$

In other words, S_β is the set of draws of m instances (with exactly l unique elements) on which the learner's hypothesis differs from the target by more than β. We can lower bound $P(S_\beta)$ by noticing that

$$\frac{1}{d} \frac{d-l}{2} P(S) \leq \sum_{\mathbf{z} \in S_\beta} P(\mathbf{z}) d(\mathcal{A}(\mathbf{z}_{h_*}), h_*) + \sum_{\mathbf{z} \in S \setminus S_\beta} P(\mathbf{z}) d(\mathcal{A}(\mathbf{z}_{h_*}), h_*)$$

$$\leq P(S_\beta) + \beta(P(S) - P(S_\beta))$$

From this we have

$$(1-\beta) P(S_\beta) \geq \left(\frac{1}{d} \frac{d-l}{2} - \beta \right) P(S)$$

Step 4:

Let $\beta = \epsilon$. Then we see that if the target were h_* then with probability at least $P(S_\epsilon)$ the learner's hypothesis would be more than ϵ away from the target. Let us find the conditions under which

$$\left(\frac{1}{d}\frac{d-l}{2} - \epsilon\right) P(S) > \delta$$

Since l was arbitrary, we can let $l = \frac{d}{2}$. In that case for all $\epsilon < \frac{1}{8}$, it is easy to check that

$$\left(\frac{1}{d}\frac{d-l}{2} - \epsilon\right) P(S) > \frac{1}{8}P(S)$$

Therefore, it is enough to find the conditions for $P(S) > 8\delta$. Recall that $P(S)$ is the probability of drawing exactly l distinct items from d items in m i.i.d. trials. There are $\binom{d}{l}$ different ways of choosing l items. For each such choice, there are $l!$ different ways in which the items could appear in the first l positions. Consider the ith such choice. Any sequence of m examples (denoted by $\mathbf{z}$) such that (i) its first l items correspond to this choice and (ii) the remaining $m-l$ items are made up of only elements of this l-set is a member of S. Let $S^{(i)}$ be the set of all elements of S that satisfy this property. Clearly

$$P(S^{(i)}) = (\frac{1}{d})^l(\frac{l}{d})^{(m-l)}$$

Since each of the $S^{(i)}$ are disjoint subsets of S, we have that

$$P(S) \geq \binom{d}{l} l!(\frac{1}{d})^l(\frac{l}{d})^{(m-l)}$$

Step 5:
Put $l = \frac{d}{2}$. Then we have

$$\binom{d}{l} l!(\frac{1}{d})^l(\frac{l}{d})^{(m-l)} = \binom{2l}{l} l!(\frac{1}{d})^l(\frac{1}{2})^{(m-l)} = \binom{2l}{l} l!(\frac{2}{d})^l(\frac{1}{2})^m = \frac{(2l)!}{(l!)(l^l)}(\frac{1}{2})^m$$

Now,

$$\frac{(2l)!}{(l!)(l^l)} = \prod_{i=1}^{l}(1+\frac{i}{l}) \geq (1+\frac{1}{2})^{\frac{l}{2}}$$

Therefore, if

$$m < \frac{l}{2}\log_2(\frac{3}{2}) + \log_2(\frac{1}{8\delta})$$

we have

$$P(S_\epsilon) \geq \frac{1}{8} P(S) > \delta.$$

Thus if the target were h_*, the probability that the learner's hypothesis is more than ϵ away from the target is greater than δ. If $\mathcal{H}$ had infinite VC dimension we can always choose a d large enough so that for every $\epsilon < \frac{1}{8}$, the probability of making a mistake larger than δ can be correspondingly arranged The class of infinite VC dimension will therefore be unlearnable in this sense. ■

2.3.6 Implications

Thus we see that the class of languages $\mathcal{L}$ must be such that $\mathcal{H} = \{1_L | L \in \mathcal{L}\}$ must have finite VC dimension for learnability to hold. An immediate corollary is

Corollary 2 *The class of all finite languages is unlearnable in the PAC setting.*

Remark: The PAC framework is often misunderstood to be one that allows a larger collection of languages to be learned than the Gold framework. In this context it is worth making two remarks. First, we have just seen that the set of all finite languages is not PAC learnable. It is easy to check that this family is Gold learnable. Second, in the PAC setting, one learns from both positive and negative examples. If the learner is allowed such a privilege in the Gold setting (learning from informants), we have see already that the entire family of *r.e.* sets may be identified. It is also possible to define collections of languages that are PAC learnable but not Gold learnable. Hence we conclude that PAC and Gold are just different frameworks with no obvious relationship.

Corollary 3 *The class of languages represented by (i) finite automata (DFA's) (ii) context free grammars (CFGs) are both unlearnable in the PAC setting.*

Interestingly, from two very different but plausible frameworks for inferring a language from finite examples, one arrives at the unlearnability of the basic families in the Chomsky Hierarchy unless further constraints are put on the problem of language acquisition.

One possible constraint that may be put on the class of languages is a constraint on the number of rewrite rules or the size of the representation

in some notational system. For example, consider the set of all languages describable by a DFA with at most n states over an alphabet Σ where $|\Sigma| = k$. Call this family H_n. It is immediate that H_n is a finite class and an upper bound on its size is given by

$$|H_n| \leq [\binom{n}{k}]^n$$

Therefore we get that the VC dimension of H_n is bounded by

$$VC(H_n) \leq \log_2([\binom{n}{k}]^n) \leq nk\log_2(n)$$

Similar calculations may be conducted for more interesting families of languages and an Occam principle may then be used.

A second aspect of the framework of statistical learning theory merits some more discussion. The distance $d(\mathcal{A}(t_k), L_t)$ between the learner's hypothesis and the target is required to decrease eventually to zero as more and more data become available, i.e., as $k \to \infty$. It is of interest to know the rate at which this convergence occurs and this issue has been a significant direction of work in the field. In general, using Hoeffding bounds on uniform laws of large numbers, it is usually possible to guarantee a rate of $O(\frac{1}{\sqrt{k}})$. The rate of convergence takes on a particular significance in a cognitive context as "learnability in the limit" is ultimately only an idealized notion that is not realized in practice. After all, humans have only a finite amount of linguistic experience on the basis of which they must generalize to novel situations. Furthermore, there appears to be a critical (maturational) time period over which much of language acquisition takes place. At the end of this learning phase, children develop mature grammars that remain relatively unaltered over their lifetime. Therefore, it becomes of interest to characterize the probability with which a typical child might acquire the target grammar after its critical linguistic experience during the learning phase. The techniques of statistical learning theory allow us to get a handle on this question. In the rest of this book, this probabilistic characterization of learnability will be used to quantify the degree to which the grammar of children might differ from that of their parents – thereby opening the door to the study of language change.

2.3.7 Complexity of Learning

Much of inductive inference began with attempts to understand the conditions under which a learning algorithm would *eventually* converge to a target grammar. Even if learnability is possible in principle, one needs to analyse the complexity of the learning task. It could well be that convergence would take too long rendering the algorithms cognitively implausible for computational reasons. Much of the PAC and VC based analyses inject complexity-theoretic ideas into an evaluation of learnability. The discussion in prior sections focused on the informational complexity of learning with bounds on how many examples one would require to learn approximately well. It is also worth noting that given a finite sample, the task of choosing an appropriate grammar that fits the data may be an optimization problem of some difficulty. A range of hardness results clarify this phenomenon. For example, it is NP-hard to find the smallest DFA consistent with a set of positive and negative example sentences (Gold, 1978). From a different point of view, Kearns and Valiant (1989) show that efficient DFA inference is hard under certain cryptographic assumptions. Abe and Warmuth (1992) consider the computational complexity of learning more general families of probabilistic grammars (e.g. Hidden Markov Models) and show that it is the computational complexity rather than informational complexity that is the barrier to efficient learnability for such cases.

2.3.8 Final Words

In this chapter, we have reviewed frameworks for meaningfully studying the abstract problem of inferring a language from examples. The problem was studied in great generality with minimal prior commitment to particular linguistic or cognitive predispositions. It is worthwhile to qualify our position here. We believe in linguistic structure. That is, we do believe that the objects of a language like phonemes, syllables, morphemes, phrases, sentences and so on may be given a formal status and a particular language may then be characterized by the unique combinatorial and compositional structure of its linguistic objects. At the same time, the formal expressions of a language do refer (semantically) to states of affairs in the world and in this sense, language mediates a relationship between form and meaning. Beyond that, we have made in this chapter, minimal commitment to the nature of linguistic structure in the world's languages or the procedure by which they are acquired by children. Therefore, our discussion has centered largely on the

general problem of acquiring formal systems via learning algorithms. The general import of the results presented here is that *tabula rasa* learning, i.e., learning in the complete absence of prior information is infeasible. Successful language acquisition therefore must come about because of the constraints inherent in the interaction of the learning child with its linguistic environment. The nature of these constraints is a matter of some debate and in the next section we take a more linguistically oriented view of this state of affairs.

Chapter 3

Language Acquisition – A Linguistic Treatment

In the previous chapter, we considered in a somewhat abstract setting the inherent difficulty of inferring the identity of a potentially infinite set on the basis of examples from this set. We considered this problem from many different points of view to highlight, in particular, how learning with unconstrained hypothesis classes is impossible. Let us summarize the essential flow of the argument so far to appreciate the implications for linguistic theory and cognitive science.

1. Languages have a formal structure and in this sense may be viewed as sets of well-formed expressions. We have already discussed several different notions of language that are elaborations of this idea leading ultimately to perhaps the most general notion of a language as a probability distribution on permissible form-meaning pairs. One may choose to work at any appropriate point in the linguistic hierarchy – from phonology to morphology to syntax to semantics – and at any such level one may meaningfully discuss the nature of the well formed expressions in any particular natural language. Successful adult usage of a language relies on tacit knowledge of the nature of these well formed expressions – knowledge that is acquired over the course of language acquisition.

2. On exposure to finite amounts of data, children are able to learn a language and generalize to produce and understand novel expressions they may have never encountered before.

3. Language acquisition does not depend upon the order of presentation of sentences and it largely proceeds on the basis of positive examples alone. There is very little explicit instruction to children regarding the nature of the grammatical rules that underlie the well-formedness of expressions. If we view grammars as compact representations of sets of expressions, then it is reasonable to think of language acquisition as a process of grammar construction, i.e., developing compact representations of the linguistic experience encountered over the course of language acquisition. Clearly children develop these representations without explicit instruction regarding the appropriate nature of such representations.

4. All naturally occurring languages are learnable by children. Thus children raised in a Mandarin speaking environment of China would learn Mandarin, in the English speaking environment of Chicago would learn English and so on. Therefore there are no language specific (specific to a particular natural language, that is) predispositions. Furthermore, the class of possible natural languages must be such that every member of this class is learnable.

5. In the complete absence of prior information (constraints on the process of language acquisition), successful generalization to novel expressions is impossible. Therefore, it must be the case that children do not consider every possible set (of well formed expressions) that is consistent with the data they receive. They consider some hypotheses and discard others – thus the class $\mathcal{H}$ must be constrained in some fashion.

6. Therefore the real issue at hand is the nature of the constraints that guide the learning child towards the correct grammatical hypotheses. Linguistic theory in the generative tradition attempts to circumscribe the range of grammatical hypotheses that humans might entertain. In this sense, most formal grammatical theories may be viewed as theories about the nature of $\mathcal{H}$. Developmental psychologists attempt to characterize the constraints on the nature of $\mathcal{A}$ — the learning algorithm that children may plausibly use during language acquisition. Taken together, they constitute an explananation for how successful language acquisition may come about.

In this chapter, we begin our discussion by examining in some detail a language learning problem in a highly constrained setting that some linguists

and psychologists have considered to be a useful model for study. This constrained setting illustrates the interaction of constraints from

1. *Linguistic Theory* as embodied in grammar formalisms such as Government and Binding (Chomsky, 1981; Haegeman, 1991), HPSG (Pollard and Sag, 1994), Optimality Theory (Prince and Smolensky, 1993)and approaches that may be accomodated within a broad construal of the Principles and Parameters view (Chomsky, 1986).

2. *Psychological Learning Theory* as embodied in learning algorithms that make minimal demands on the learner in terms of memory and computational burdens. Examples of such algorithms include the Triggering Learning Algorithm (Gibson and Wexler, 1994), the Stratified Learning Hierarchies of Tesar and Smolensky, 1996, and related approaches described in Bertolo, 2001.

The origin of some of the ideas presented in this chapter lie in an attempt to formulate the learning theoretic underpinnings for the Triggering Learning Algorithm presented in "Triggers" (Gibson and Wexler, 1994). In the next section, we provide a glimpse of some of the linguistic reasoning that accompanies the learning-theoretic considerations underlying the generative approach to linguistics.

We then give a brief account of the Principles and Parameters framework, and the issues involved in learning within this framework. This sets the stage for our investigations, and we use as a starting point the Triggering Learning Algorithm (TLA) working on a three-parameter syntactic subsystem first analyzed by Gibson and Wexler. The significant portion of the chapter analyzes the TLA from the perspective of learnability. Issues pertaining to parameter learning in general, and the TLA in particular, are discussed at appropriate points.

In the next chapter, we continue with the analysis developed here. We show how the framework allows us to characterize the sample complexity of learning in parameterized linguistic spaces. We then generalize well beyond the Principles and Parameters approach. In particular, we discuss how the Markov chain framework developed for the analysis of the TLA in parameterized linguistic spaces is applicable to *any* learning algorithm for *any* class of grammars (languages). Some general properties and special cases are then considered for important classes of cognitively plausible learning algorithms. The techniques thus developed will allow us to characterize the behavior of

the individual learner precisely and create the learning-theoretic foundation upon which the rest of the book is developed.

3.1 Language Learning and The Poverty of Stimulus

While it is clear from the discussion in the previous chapter that the class $\mathcal{H}$ of possible grammatical hypotheses must be constrained to ensure successful learnability, the analysis was conducted in a very abstract setting providing virtually no insight into the possible nature of these constraints. Much of linguistic theory, however, is developed with the goal of elucidating the precise nature of such constraints. The learning-theoretic arguments presented formally in the Gold setting and its variants in the previous chapter are developed somewhat more informally as the *poverty of stimulus* argument in the vast literature in generative linguistics.

To develop some appreciation for the nature of the reasoning involved, consider the following paradigmatic case of question formation in English.

(1) John is bald.

(1a) Is John bald?

Users of English are able to take declarative statements as in (1) and convert them to appropriate questions as in (1a). Thus both (1) and (1a) are sequences of English words that speakers of English recognize as grammatical. Presumably, their knowledge (unconscious) of the underlying grammar of English enables them to make this judgement.

Suppose the learner is provided with (1) and (1a) as examples of English sentences. On the basis of this, there are many different rules that the learner may logically infer. For example, let us consider two different rules that are consistent with the data that may be inferred.

1. Given a declarative sentence, take the second word in the sentence and move it to the front to form the appropriate question.
2. Given a declarative sentence, take the first "is" and move it to the front.

Now consider the novel declarative statement where there are multiple instances of the word "is".

(2) The man who is running is bald.

What might the appropriate interrogative form of this statement be? We instinctively recognize that (2a) is the appropriate interrogative form as opposed to (2b) below.

(2a) Is the man who is running < > bald?

(2b*) Is the man who < > running is bald?

Examining (2),(2a), and (2b*) we see that neither rule 1 nor rule 2 is adequate for making the correct generalization from (2) to (2a) but not (2b*). In order to make the correct generalization, one must recognize that grammatical sequences have an internal structure. Thus, one may write (1) as

(1) <John> is <bald>

and write (2) as

(2) <The man who is running> is <bald>

It is now clear that rule 1 is clearly wrong whereas rule 2 may be saved with some additional consideration. To fully develop the rules for question formation will require significantly greater analysis but at this point, there are already two main conclusions that linguists would draw. First, that given a finite amount of data, there are always many grammatical rules consistent with the data but which make different generalizations about novel examples. We have already explored this issue in some mathematical detail in the previous chapter where the problem of inferring the correct rule system (grammar) was studied. Second, that sentences have an internal structure and these constituents play an important role in determining the correct grammatical system with the right generalization properties.

The *poverty of stimulus* argument[1] is developed through several paradigmatic cases of the form that we have just considered from many different

[1]The poverty of stimulus argument has been and has remained controversial. While the discussion of the previous chapter argues that some form of prior constraint is inevitable, disagreements are over the precise nature of these constraints and how much of it is language specific. For two different kinds of discussions about this question, see Pullum and Scholz (2002) and Elman et al (1996).

areas of language (from morphology to syntax) and in many languages of the world.

A rich literature exists on the kinds of generalizations children appear to make during language acquisition. This is accompanied by much theorizing about the kinds of grammatical objects and rules that are invoked in the different languages of the world. (See Atkinson, 1992; Crain and Thornton, 1998 for a treatment of language acquisition from the perspective of generative linguistics).

One particular approach to these questions has been the Principles and Parameters (P&P) approach that we discuss in the next section. Before proceeding to examine the basic tenets and some examples of the P&P approach to language, it is worthwhile to clarify that it represents only one particular point of view. Many other approaches exist and most linguistic theories including those subsumed by P&P may be regarded as tentative theses about the nature of $\mathcal{H}$ — theses that may well turn out to be inadequate or wrong. Therefore it would certainly be premature for us to commit ourselves to any one particular articulation of $\mathcal{H}$. Rather, we aim to present the basic principles of learning and evolution in a manner that is general enough to be adapted to any particular linguistic setting depending upon one's theoretical persuasion.

With that qualifying remark, let us move on.

3.2 Constrained Grammars–Principles and Parameters

Having recognized the need for constraints on the class of grammars $\mathcal{H}$, researchers have investigated several possible ways of incorporating such constraints in the classes of grammars to describe the natural languages of the world. Examples of this range from linguistically motivated grammars such as Head-driven Phrase Structure Grammars (HPSG), Lexical-Functional Grammars (LFG), Government and Binding (GB), Optimality Theory (OT) to bigrams, trigrams and connectionist schemes suggested from an engineering consideration of the design of spoken language systems. Note that every such grammar suggests a very specific model for human language, with its own constraints and its own complexity.

The Principles and Parameters framework (Chomsky, 1981) attempts to describe $\mathcal{H}$ in a parametric fashion. It tries to articulate the "universal" principles common to all the natural languages of the world and the param-

eters of variation across them. On this account, roughly speaking, there are a finite number of principles governing the production of human languages. These abstract principles can take one of several (finite) specific forms—this specific form manifests itself as a rule, peculiar to a particular language (or class of languages). The specific form that such an abstract principle can take is governed by setting an associated parameter to one of several values. In typical versions of theories constructed within such a framework, one therefore ends up with a parameterized class of grammars. The parameters are boolean valued–setting them to one set of values, defines the grammar of German (say), setting them to another set of values, defines the grammar, perhaps, of Chinese.

One may also view this as an attempt to recover the principal dimensions of language variation. A high level analogy with data analysis is perhaps appropriate here. Given a set of data points $\mathbf{x}_1, \ldots \mathbf{x}_n$ in a k dimensional space (where k is large) the well known technique of *principal components analysis* projects the data into a subspace that preserves as much of the variation in the data set as possible. Parametric modeling of the data is then conducted in this lower dimensional subspace. Each naturally occuring language may be viewed as a data point in a very high dimensional space. To reduce the problem of modeling the space of all naturally occuring languages to more manageable proportions one tries to reduce the dimensionality (perhaps drastically) by considering subspaces or modules that are "important" in some sense.

Although the syntactic framework of Government and Binding (Chomsky, 1981) is most closely associated with the P&P framework, a broad interpretation of this point of view would include several additional linguistic theories such as HPSG, varieties of LFG, Optimality Theory and so on. For example, in a constraint based theory such as Optimality Theory, there are n universal constraints and different natural languages differ in the ranking they give to each of these constraints. On this account, there are $n!$ different natural language grammars.

These ideas are best illustrated in the form of examples. We provide, now, two examples, drawn from syntax and phonology respectively.

3.2.1 Example: A 3-parameter System from Syntax

Different aspects of syntactic competence are treated in a modular fashion and here we discuss three syntactic parameters that were first extensively studied in the framework of learning theory by Gibson and Wexler (1994).

Two X-bar parameters:

A classic example of a parametric grammar for syntax comes from X-bar theory (see Jackendoff, 1977 for an early exposition). This describes a parameterized phrase structure grammar, which defines the production rules for constituent phrases, and ultimately sentences in the language. The general format for phrase structure is summarized by the following parameterized production rules:

$$XP \rightarrow \text{Spec } X'(p_1 = 0) \text{ or } X' \text{ Spec}(p_1 = 1)$$
$$X' \rightarrow \text{Comp } X'(p_2 = 0) \text{ or } X' \text{ Comp}(p_2 = 1)$$
$$X' \rightarrow X$$

XP refers to an X-phrase, where X, or the "head", is a lexical category like N (Noun), V (Verb), A (Adjective), P (Preposition), and so on. Thus, one could generate a Noun Phrase (denoted by NP) where the head $X = N$ is a noun, or a Verb Phrase (VP) and other kinds of phrases by a recursive application of these production rules. "Spec" refers to *specifier*, in other words, that part of the phrase that "specifies" it, roughly like *the old* in the Noun Phrase, *the old book*.

"Comp" refers to the *complement*, roughly a phrase's arguments, like *an ice-cream* in the Verb Phrase *ate an ice-cream*, or *with envy* in the Adjective Phrase *green with envy*. Spec and Comp are constituents and could be phrases with their own specifiers and complements. Furthermore, in a particular phrase, the spec-position, or the comp-position might be blank (in these cases, Spec $\rightarrow \emptyset$, or Comp $\rightarrow \emptyset$ respectively).

Thus we might include the additional rules

$$\text{Spec} \rightarrow XP; \text{Spec} \rightarrow \emptyset$$

and

$$\text{Comp} \rightarrow XP; \text{Comp} \rightarrow \emptyset$$

Further, these rules are parameterized. Languages can be spec-first ($p_1 = 0$) or spec-final ($p_1 = 1$). Similarly, they can be comp-first, or comp-final. For example, the parameter settings of English are (spec-first,comp-final). Applying these rules recursively, one can thus generate embedded phrases of arbitrary length in the language.

As an example, consider the English Noun Phrase (after Haegeman, 1991)

$$[\text{The investigation (of [the corpse]}_{NP})_{PP}(\text{after lunch})_{PP}]_{NP}$$

The constituents are indicated by bracketing. The square bracket [is used to indicate a Noun Phrase (NP) while a regular bracket (is used to indicate a Prepositional Phrase (PP). A partial derivation of the entire phrase is provided below:

$$XP \rightarrow [\text{Spec}X'] \rightarrow [\text{Spec}[X'\text{Comp}]] \rightarrow [\text{Spec}[X'XP]]$$

$$\rightarrow [\text{Spec}[[X'XP]XP]] \rightarrow [\text{Spec}[[XXP]XP]] \rightarrow [the[[NPP]PP]]$$

Thus, the N expands into the noun "investigation" and the two prepositional phrases expand into "of the corpse" and "after lunch" respectively by a similar application of these rules.

In general, the derivation may be described as a tree structure. Shown in Fig. 3.1 is an embedded phrase which demonstrates the use of the X-bar production rules (with the English parameter settings) to generate another arbitrary English phrase.

In contrast the parameter settings of Bengali are (spec-first,comp-first). The translation of the same sentence is provided in Fig. 3.2. Notice, how a difference in the comp-parameter setting causes a systematic difference in the word orders of the two languages. It is claimed by some linguists that as far as basic, underlying word order is concerned, X-bar theory covers all the important possibilities for natural languages[2]. Languages of the world simply differ in their settings with respect to the X-bar parameters.

One transformational parameter (V2): The two parameters described above define generative rules to obtain basic word-order combinations permitted in the world's languages. As mentioned before, there are many other aspects which govern the formation of sentences. For example, there are transformational rules which determine the production of surface word order from the underlying (base) word-order structure obtained from the production rules above. We saw an example of this earlier when we studied the relation between the interrogative form and the declarative form of the same sentence. The interrogative form was obtained by a transformation of the declarative. This transformation involved moving an appropriate constituent (in the example considered, it was the word "is") to the front.

[2]There are a variety of other formalisms developed to take care of finer details of sentence structure. This has to do with case theory, movement, government, binding and so on. See Haegeman (1991). There is also the issue of scrambling and how to deal with languages having apparently free word order.

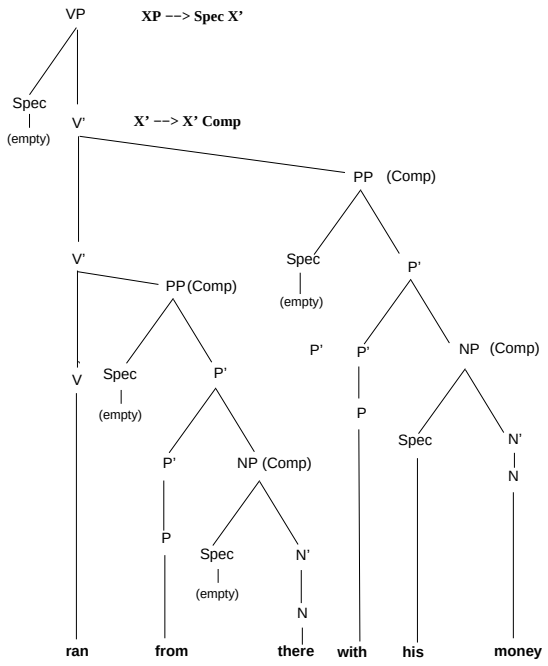

Figure 3.1: Analysis of an English sentence. The parameter settings for English are spec-first, and comp-final.

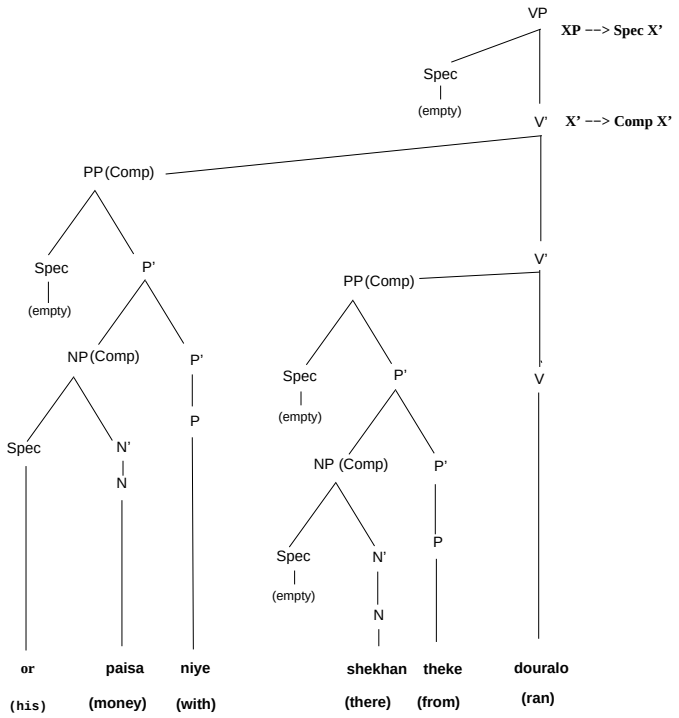

Figure 3.2: Analysis of the Bengali translation of the English sentence of the earlier figure. The parameter settings for Bengali are spec-first, and comp-first.

These transformational rules may also be parameterized and one such parameterized transformational rule that governs the movement of words within a sentence is associated with what has come to be known as the $V2$ parameter. It is observed that in German and Dutch declarative sentences, the relative order of verbs and their complements seem to vary depending upon whether the clause in which they appear is a root clause or subordinate clause. Consider, the following German clauses:

(1)...dass (that) Karl das (the) Buch (book) kauft (buys).
...that Karl buys the book.

(2)...Karl kauft das Buch.
...Karl buys the book.

This seems to present a complication in that from these sentences it is not clear whether German is comp-first (as example 1 seems to suggest where the complement "das Buch" *precedes* the verb "kauft") or comp-final (as example 2 seems to suggest). It is believed (Haegeman, 1991) that the underlying word-order form is comp-first (like Bengali, and unlike English, in this respect); however, the $V2$ parameter is set for German ($p_3 = 1$). This implies that finite verbs must move so as to appear in exactly the second position in root declarative clauses ($p_3 = 0$ would mean that this need not be the case). Therefore the surface form of (2) is derived by applying such a $V2$ constraint after generating a base form according to comp-first generative rules. This may be viewed as a specific application of a more general transformational rule Move-α. For details and analysis, see (Haegeman, 1991).

Each of these three parameters can take one of two values. There are, thus, 8 possible grammars (grammatical modules really), and correspondingly 8 languages by extension, generated in this fashion. At this stage, the languages are defined over a vocabulary of syntactic categories, like N, V and so on. Applying the three parameterized rules, one would obtain different ways of combining these syntactic units to form valid expressions (sentences) in each of the 8 languages. The appendix contains a list of the set of unembedded (degree-0) sentences obtained for each of the languages, L_1 through L_8 in this parametric system. The vocabulary has been modified so that sentences are now defined over more abstract units than syntactic categories.

3.2.2 Example: Parameterized Metrical Stress in Phonology

The previous example dealt with a parameterized family for a syntactic module of grammar. Let us now consider an example from phonology. Our example relates to the domain of metrical stress which describes the possible ways in which words in a language can be accented during pronunciation.

Consider the English word, "candidate". This is a three syllable word, composed of the three syllables, /can/,/di/,and, /date/. A native speaker of American English typically pronounces this word by stressing the first syllable of this word. Similarly, such a native speaker would also stress the first syllable of the tri-syllabic word, "/al/-/pha/-/bet/" so that it almost rhymes with "candidate". In contrast, a French speaker would stress the final syllable of both these words—a contrast which is perceived as a "French" accent by the English ear.

For simplicity, assume that stress has two levels, i.e., each syllable in each word can be either stressed, or unstressed[3]. Thus, an n-syllable long word could have, in principle, as many as 2^n different possible ways of being stressed. For a particular language, however, only a small number of these ways is phonologically well-formed. Other stress patterns sound accented, or awkward. Words could potentially be of arbitrary length[4]. Thus one could write phonological grammars—a functional mapping from these words to their correct stress pattern. Further, different languages correspond to different such functions,i.e., they correspond to different phonological grammars. Within the Principles and Parameters framework, an attempt is made to parameterize these phonological grammars.

Let us consider a simplified version of two principles associated with 3 boolean valued parameters that play a role in the Halle and Idsardi (1992) metrical stress system. These principles describe how a multisyllable word can be broken into its constituents (recall how sentences were composed of

[3]While we have not provided a formal definition of either stress, or syllable, it is hoped, that at some level, the concepts are intuitive to the reader. It should, however, be pointed out that linguists differ on their characterization of both these objects. For example, how many levels can stress have? Typically, (Halle and Idsardi, 1992) three levels are assumed. Similarly, syllables are classified into heavy and light syllables. We have discounted such niceties for ease of presentation.

[4]One shouldn't be misled by the fact that that a particular language has only a finite number of words. When presented with a foreign word, or a "nonsense" word one hasn't heard before, one can still attempt to pronounce it. Thus, the system of stress assignment rules in our native language probably dictates the manner in which we choose to pronounce it. Speakers of different languages would accent these nonsense words differently.

constituent phrases in syntax) before stress assignment takes place. This is done by a bracketing schema which places brackets at different points in the word, thereby marking (bracketing) off different sections as constituents. A constituent is then defined as a syllable sequence between consecutive brackets. In particular, a constituent must be bounded by a right bracket on its right edge, or, a left bracket on its left edge (both these conditions need not be satisfied simultaneously). Further, it cannot have any brackets in the middle. Finally, note that not all syllables of the word need be part of a constituent. A sequence of syllables might not be bracketed by either an appropriate left, or right bracket—such a sequence, cannot have a stress-bearing head, and might be regarded as an extra-metrical sequence.

1. the **edge** parameters: there are two such parameters.
   a) put a left ($p_1 = 0$) or right ($p_1 = 1$) bracket
   b) put the above mentioned bracket exactly one syllable *after* the left ($p_2 = 0$) edge or *before* the right ($p_2 = 1$) edge of the word.

2. the **head** parameter: each constituent (made up of one or more syllables) has a "head". This is the stress bearing syllable of the constituent, and is in some sense, the primary, or most important syllable of that constituent (recall how syntactic constituents, the phrases, had a lexical head). This phonological head could be the *leftmost* ($p_3 = 0$), or, the *rightmost* ($p_3 = 1$) syllable in the constituent.

Suppose, the parameters are set to the following set of values: [$p_1 = 0,\ p_2 = 0,\ p_3 = 0$]. Fig. 3.3 shows how some multisyllable words would have stress assigned to them. In this case, any n-syllable word would have stress in exactly the second position (if such a position exists) and no other. In contrast, if [$p_1 = 0,\ p_2 = 0,\ p_3 = 1$], the corresponding language would stress the final syllable of all multi-syllable words. Monosyllabic words are unstressed in both languages.

These 3 parameters represent a very small (almost trivial) component of stress pattern assignment. There are many more parameters which describe in more complete fashion, metrical stress assignment. At this level of analysis, for example, the language Koya has $p_3 = 0$, while Turkish has $p_3 = 1$; see Kenstowicz (1994) for more details. This example provides a flavor of how the problem of stress-assignment can be described formally by a parametric family of functions. The analysis of parametric spaces developed in this chapter can be equally well applied to such stress systems.

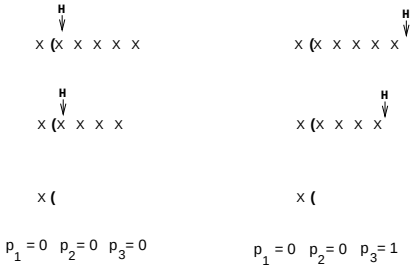

Figure 3.3: Depiction of stress pattern assignment to words of different syllable length under the parameterized bracketing scheme described in the text.

3.3 Learning in the Principles and Parameters Framework

Language acquisition in the Principles and Parameters framework reduces to the estimation of the parameters corresponding to the "target" language. A child is born in an arbitrary linguistic environment. It receives examples in the form of sentences it hears in its linguistic environment. On the basis of example sentences it hears, it presumably learns to set the parameters appropriately. Thus, referring to our 3-parameter system for syntax, if the child is born in a German speaking environment, and hears German sentences, it should learn to set the V2 parameter (to $+V2$), and the spec-parameter to spec-first. Similarly, a child hearing English sentences, should learn to set the comp-parameter to comp-final. In principle, the child is thus solving a parameter estimation problem—an unusual class of parameter estimation problems, no doubt, but in spirit, little different from the parameter estimation problems that are encountered in statistical learning. One can thus ask a number of questions about such problems. What sort of data does the child need in order to set the target parameters? Is such data readily available to the child? How often is such data made available to the child? What sort of algorithms does the child use in order to set the parameters? How efficient are these algorithms? How much data does the child need? Will the child always converge to the target "in the limit" ?

Language acquisition, in the context of parameterized linguistic theories, thus, gives rise to a class of learning problems associated with finite parame-

ter spaces. Furthermore, as emphasized particularly by Wexler in a series of works (Hamburger and Wexler, 1975; Culicover and Wexler, 1980; and Gibson and Wexler, 1994), the finite character of these hypothesis spaces does *not* solve the language acquisition problem. As Chomsky notes in *Aspects of the Theory of Syntax* (1965), the key point is how the space of possible grammars– even if finite–is "scattered" with respect to the primary language input data. It is logically possible for just two grammars (or languages) to be so near each other that they are not separable by psychologically realistic input data. This was the thrust of Wexler and Hamburger, and Culicover and Wexler's earlier work on the learnability of transformational grammars from simple data (with at most 2 embeddings). More recently, a significant analysis of specific parameterized theories has come from Gibson and Wexler (1994). They propose the Triggering Learning Algorithm—a simple, psychologically plausible algorithm which children might conceivably use to set parameters in finite parameter spaces. Investigating the performance of the TLA on the 3-parameter syntax subsystem discussed in our previous example yields the surprising result, that the TLA cannot achieve the target parameter setting for every possible target grammar in the system. Specifically, there are certain target parameter settings, for which the TLA could get stuck in *local maxima* from which it would never be able to leave, and consequently, learnability would never result.

In this chapter, our interest lies both in the *learnability,* and the *sample complexity* of the finite hypothesis classes suggested by the Principles and Parameters theory. An investigation of this sort requires us to define the important dimensions of the learning problem—the issues which need to be systematically addressed. The following figure provides a schematic representation of the space of possibilities which need to be explored in order to completely understand and evaluate a parameterized linguistic family from a learning-theoretic perspective. The important dimensions are as follows:

1. the *parameterization* of the language space itself: a particular linguistic theory would give rise to a particular choice of universal principles, and associated parameters. Thus, one could vary along this dimension of analysis, the parameterization of the hypothesis classes which need to be investigated. The parametric system for metrical stress (Example 2) is due to Halle and Idsardi (1992). A variant, investigated by Dresher and Kaye (1990), can equally well be subjected to analysis.

2. the *distribution* of the input data: once a parametric system is decided upon, one must, then, decide the probability distribution according

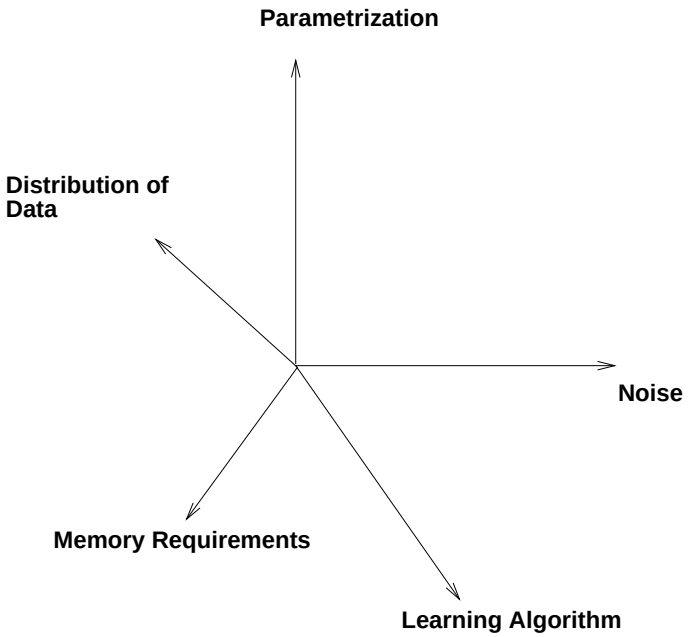

Figure 3.4: The space of possible learning problems associated with parameterized linguistic theories. Each axis represents an important dimension along which specific learning problems might differ. Each point in this space specifies a particular learning problem. The entire space represents the class of learning problems that are considered interesting within the Principles and Parameters framework.

to which data (i.e., sentences generated by some target grammar belonging to the parameterized family of grammars) is presented to the learner. Clearly, not all sentences occur with equal frequency. Some are more likely than others. How does this affect learnability? How does this affect sample complexity? One could, of course, attempt to come up with distribution-independent bounds on the sample complexity. This, as we shall soon see, is not possible.

3. the presence, and nature, of *noise*, or extraneous examples: in practice, children are exposed to noise (sentences, which are inconsistent with the target grammar) due to the presence of foreign, or idiosyncratic speakers, disfluencies in speech, or a variety of other reasons. How does one model noise? How does it affect sample complexity or learnability or both?

4. the type of learning *algorithm* involved: a learning algorithm is an effective procedure mapping data to hypotheses (parameter values). Given that the brain has to solve this mapping problem, it then becomes of interest to study the space of algorithms which can solve it. How many of them converge to the target? What is their sample complexity? Are they psychologically plausible?

5. the use of *memory*: this is not really an independent dimension, in the sense, that it is related to the kind of algorithm used. The TLA and variants, as we shall soon see, are memoryless algorithms. These can be modeled by a Markov chain.

This is the space that needs to be explored. By making a specific choice along each of the five dimensions discussed (corresponding to a single point in the 5-dimensional space of Fig. 3.4, we arrive at a specific learning problem. Varying the choices along each dimension (thereby traversing the entire space of Fig. 3.4) gives rise to the class of learning problems associated with parameterized linguistic theories. For our analysis, we choose as a concrete starting point the Gibson and Wexler Triggering Learning Algorithm (TLA) working on the 3-parameter syntactic subsystem in the example shown. In our space of language learning problems, this corresponds to (1) a 3-way parameterization, using mostly X-bar theory; (2) a uniform sentence distribution over unembedded (degree-0) sentences; (3) no noise; (4) a local gradient ascent search algorithm; and (5) memoryless (online) learning. Following our analysis of this learning system, we consider variations in learning

algorithms, sentence distribution, noise, and language/grammar parameterizations.

3.4 Formal Analysis of the Triggering Learning Algorithm

Let us start with the TLA. We first show that this algorithm and others like it are completely modeled by a Markov chain. We explore the basic computational consequences of this fundamental fact, including some surprising results about sample complexity and convergence time, the dominance of random walk over hill climbing, and the potential applicability of these results to actual child language acquisition and possibly language change — a theme that we build upon over the course of this book.

3.4.1 Background

Following Gold (1967) and Gibson and Wexler (1994) the basic framework is that of *identification in the limit*. Recall Gold's assumptions from the previous chapter. The learner receives an (infinite) sequence of (positive) example sentences from some target language. After each example presentation, the learner either (i) stays in the same state; or (ii) moves to a new state (changes its parameter settings). If after some finite number of examples the learner converges to the correct target language and never changes its guess, then it has correctly identified the target language in the limit; otherwise, it fails.

In the Gibson and Wexler model (and others) the learner obeys two additional fundamental constraints: (1) the *single-value constraint*—the learner can change only 1 parameter value each step; and (2) the *greediness constraint*—if the learner is given a positive example it cannot recognize, it will change a parameter value, only if with the new parameter settings it is now able to analyze the new sentence. The TLA can then be precisely stated as follows. See Gibson and Wexler (1994) for further details.

- [Initialize] Step 1. Start at some random point in the (finite) space of possible parameter settings, specifying a single hypothesized grammar with its resulting extension as a language;

- [Process input sentence] Step 2. Receive a positive example sentence s_i on ith iteration (examples drawn from the language of a single target

grammar, $L(G_t)$), from a uniform distribution on the degree-0 sentences of the language (we relax this distributional constraint later on);

- [Learnability on error detection] Step 3. If the current grammar parses (generates) s_i, then go to Step 2; otherwise, continue.
- [Single-step hill climbing] Step 4. Select a single parameter uniformly at random, to flip from its current setting, and change it (0 mapped to 1, 1 to 0) *iff that change allows the current sentence to be analyzed*;
- [Iterate] Step 5. Go to Step 2.

Of course, this algorithm never halts in the usual sense. Gibson and Wexler aim to show under what conditions this algorithm converges "in the limit"—that is, after some number, m, of steps, where m is unknown, the correct target parameter settings will be selected and never changed. They investigate the behavior of the TLA on a linguistically natural, 3-parameter subspace (of the complete linguistic parametric space which involves many more parameters). This subspace was discussed in Sec. 3.2.1 and will be reviewed again shortly. Note that a *grammar* in this space is simply a particular n-length array of 0's and 1's; hence there are 2^n possible grammars (languages). Gibson and Wexler's surprising result is that the simple 3-parameter space they consider is unlearnable in the sense that positive-only examples can lead to *local maxima*—incorrect hypotheses from which a learner can never escape. More broadly, they show that learnability in such spaces is still an interesting problem, in that there is a substantive learning theory concerning feasibility, convergence time, and the like, that must be addressed beyond traditional linguistic theory and that might even choose between otherwise adequate linguistic theories.

Remark. Various researchers (Clark & Roberts 1993; Frank & Kapur, 1992; Gibson & Wexler, 1994; Lightfoot, 1991, Fodor, 1998, Bertolo, 2001) have explored the notion of *triggers* as a way to model parameter space language learning. For these researchers, triggers are essentially sentences from the target that cannot be analyzed by the learner's current grammatical hypothesis and thereby indirectly inform it about the correct hypothesis. Gibson and Wexler suggest that the existence of triggers for every (hypothesis, target) pair in the space suffices for TLA learnability to hold. As we shall see later, one important corollary of our stochastic formulation shows that this condition does *not* suffice. In other words, even if a *triggered* path exists from the learner's hypothesis language to the target, the learner might,

with high probability, not take this path, resulting in nonlearnability. A further consequence is that many of Gibson and Wexler's proposed cures for nonlearnability in their example system, such as a "maturational" ordering imposed on parameter settings, simply do not apply. On the other hand, this result reinforces Gibson and Wexler's basic point that apparently simple parameter-based language learning models can be quite subtle—so subtle that even a seemingly complete computer simulation can fail to uncover learnability problems.

3.4.2 The Markov formulation

Given this background, we turn directly to the formalization of parameter space learning in terms of Markov chains. This formalization is in fact suggested but left unpursued in a footnote of Gibson and Wexler (1994).

Parameterized Grammars and their Corresponding Markov Chains

Consider a parameterized grammar (language) family with n parameters. We picture the 2^n-size hypothesis space as a set of points; see Fig. 3.5 for the 3-parameter case. Each point corresponds to one particular vector of parameter settings (languages, grammars). Call each point a *hypothesis state* or simply *state* of this space. As is conventional, we define these languages over some alphabet[5]. One state is the target language (grammar). Without loss of generality, we may place the (single) target language at the center of this space. Since by the TLA the learner is restricted to moving at most 1 binary value in a single step, the theoretically possible transitions between states can be drawn as (directed) lines connecting parameter arrays (hypotheses) that differ by at most 1 binary digit (a 0 or a 1 in some corresponding position in their arrays). (Recall that the distance between the grammars in parameter space is the so-called *Hamming distance*.)

We may further place *weights*, b_{ij}, on the transitions from state i to state j. These correspond to the probabilities that the learner will move from hypothesis state i to state j. In fact, given a probability distribution over the sentences of the target language $L(G)$, we can carry out an exact calculation of these transition probabilities themselves. Thus, we can picture the TLA learning space as a directed, labeled graph V with 2^n vertices.[6]

[5] Following standard notation, Σ denotes a finite alphabet and Σ^* denotes the set of all finite strings (sentences) obtained by concatenating elements of Σ.

[6] Gibson and Wexler construct an identical transition diagram in the description of their

As mentioned, not all these transitions will be possible in general. For example, by the single value hypothesis, the system can only move 1 bit at a time. Also, by assumption, only differences in surface strings can force the learner from one hypothesis state to another. For instance, if state i corresponds to a grammar that generates a language that is a proper subset of another grammar hypothesis j, there can never be a transition from j to i, and there might be one from i to j. Further, it is clear that once we reach the target grammar there is nothing that can move the learner from this state, since no positive evidence will cause the learner to change its hypothesis. Thus, there must be a loop from the target state to itself, and no exit arcs. In the Markov chain literature, this is known as an *Absorbing State* (A). Obviously, a state that leads only to an absorbing state will also drive the learner to that absorbing state. If a state corresponds to a grammar that generates some sentences of the target there is always a loop from that state to itself, with some nonzero probability. Finally, let us introduce the notion of a **closed set of states** C to be any proper subset of states in the Markov chain such that there is no arc from any of the states in C to any state outside C in the Markov chain (see Isaacson & Madsen, 1976; Resnick, 1992, and later in this chapter for further details). In other words, it is a set of states from which there is no way out to other states lying ouside this set. Clearly, a closed set with only one element (state) is an absorbing state.

Note that in the absence of noise, the target state is always an Absorbing State in the systems under discussion. This is because once the learner is at the target grammar, all examples it receives are analyzable and it will never exit this state. Consequently, the Markov chains we will consider always have at least one A. Given this formulation, one can immediately give a very simple learnability theorem stated in terms of the Markov chains corresponding to finite parameter spaces and learning algorithms[7]. We do this below.

---

computer program for calculating local maxima. However, this diagram is not explicitly presented as a Markov structure; it does not include transition probabilities, which we shall see lead to crucial differences in learnability results. Of course, topologically, both structures must be identical.

[7]Note that learnability requires that the learner converge to the target state from *any* initial state in the system.

Markov Chain Criteria for Learnability

We argued how the behavior of the Triggering Learning Algorithm can be formalized by a Markov Chain. This argument will be formally completed by providing details of the transition probabilities in a little while. While the formalization is provided for the TLA, every memoryless learning algorithm $\mathcal{A}$ for identifying a target grammar g_f from a family of grammars $\mathcal{G}$ via positive examples can be formalized as a Markov chain M (see the following chapter). In particular, M has as many states as there are grammars in $\mathcal{G}$ with the states in M being in 1-1 correspondence with grammars $g \in \mathcal{G}$. The target grammar g_f corresponds to a target state s_f of M. We call M the Markov chain *associated with* the triple ($\mathcal{A}$, $\mathcal{G}$, g_f), and the triple itself a *memoryless learning system*, or *learning system* for short. The triple decides completely the topology of the chain. The transition probabilities of the chain are related to the probability P with which sentences are presented to the learner.

An important question of interest is whether or not the learning algorithm $\mathcal{A}$ identifies the target grammar in the limit. The following theorem shows how to translate this conventional Gold-learnability criterion for identifiability in the limit into a corresponding Markov chain criterion for such memoryless learning systems.

We first recall the familiar definition for a probabilistic version of Gold-learnability:

Definition 10 *Consider a family of grammars $\mathcal{G}$, a target grammar $g_f \in \mathcal{G}$, and a learning algorithm $\mathcal{A}$ that is exposed to sentences from the target according to some arbitrary distribution P. Then g_f is said to be* **Gold learnable by** *$\mathcal{A}$ for the distribution P if and only if $\mathcal{A}$ identifies g_f in the limit with probability 1.*

A family of grammars $\mathcal{G}$ is Gold-learnable if and only if each member of $\mathcal{G}$ is Gold-learnable.

The learnability theorem below says that if a target grammar $g_f \in \mathcal{G}$ is to be Gold-learnable by $\mathcal{A}$, then the Markov chain associated with the particular learning system must be restricted in a certain way. To understand the statement of the theorem, we first recall the related notions of *absorbing state* and *closed set of states*. Intuitively, these terms refer to Markov chain connectivity and associated probabilities: an absorbing state has no exit link to any other state, while a *closed set of states* is the extension of the

absorbing state notion to a *set* of states. They have already introduced informally in the earlier section for pedagogical reasons. They are reproduced here again for completeness of the current formal account.

Definition 11 *Given a Markov chain M, an* **absorbing state** *of M is a state $s \in M$ that has no exit arcs to any other states of M.*

Since by the definition of a Markov chain the sum of the transition probabilities exiting a state must equal one, it follows that an absorbing state must have a self-loop with transition probability 1. In a learning system that makes transitions based on error detection, the target grammar will be an absorbing state, because once the learner reaches the target state, all examples are analyzable and the learner will never exit that state.

Definition 12 *Given a Markov chain M, a* **closed set of states** *(C) is any proper subset of states in M such that there is no arc from any of the states* in *C to any state* not in *C.*

If two states belong to the same closed set C then there may be transitions from one to the other. Further, there can be transitions from states *outside* C to states *within* C. However, there cannot be transitions from states within C to states outside C. Clearly, an absorbing state represents the special case of a closed set of states consisting of exactly one element, namely, the absorbing state itself.

We can now state the learnability theorem.

Theorem 10 *Let $< \mathcal{A}, \mathcal{G}, g_f \in \mathcal{G} >$ be a memoryless learning system. Let sentences from the target be presented to the learner according to the distribution P and let M be the Markov chain associated with this learning system. Then the target g_f is Gold-learnable by $\mathcal{A}$ for the distribution P* if and only if *M is such that every closed set of states in it includes the target state corresponding to g_f.*

Proof: This has been relegated to the appendix for continuity of reading. ■

Thus, if we are interested in the Gold-learnability of a memoryless learning system, one could first construct the Markov chain corresponding to such a system and then check to see if the closed sets of the chain satisfy the conditions of the above theorem. If and only if they do, the system is Gold-learnable.

We now provide an informal example of how to construct a Markov chain for a parametric family of languages. This is followed by a formal account of how to compute the transition probabilities of the Markov chain. Finally, we note some additional properties of the learning system that fall out as a consequence of our analysis. For example, our analysis is consistent with the subset principle, it can handle a variety of algorithms, and even noise.

Example.
Consider the following 3-parameter system studied by Gibson and Wexler (1994). Its binary parameters are: (1) Spec(ifier) first (0) or last (1); (2) Comp(lement) first (0) or last (1); and Verb Second constraint (V2) does not exist (0) or does exist (1). Recalling our discussion in the previous section, we follow standard linguistic convention. Thus, by *Specifier* we mean the part of a phrase that "specifies" that phrase, roughly, like *the old* in *the old book*; by *Complement* we mean roughly a phrase's arguments, like *an ice-cream* in *John ate an ice-cream* or *with envy* in *green with envy*. There are also 7 possible "words" in this language: S, V, O, O1, O2, Adv, and Aux, corresponding to Subject, Verb (Main), Object, Direct Object, Indirect Object, Adverb, and Auxiliary Verb. There are 12 possible surface strings for each (−V2) grammar and 18 possible surface strings for each (+V2) grammar if we restrict ourselves to unembedded or "degree-0" examples for reasons of psychological plausibility (see Wexler & Culicover, 1980; Lightfoot, 1991; and Gibson & Wexler, 1994 for discussion). Note that the "surface strings" of these languages are actually *phrases* such as [Subject, Verb, Object] as in *John ate an ice-cream*. Figure (3) of Gibson and Wexler summarizes the possible binary parameter settings in this system. For instance, parameter setting #5 corresponds to the array [0 1 0]= Specifier first, Comp last, and −V2, which works out to the possible basic English surface phrase order of Subject–Verb–Object (SVO). As shown in Gibson and Wexler's figure (3), the other possible arrangements of surface strings corresponding to this parameter setting include S V; S V O1 O2 (two objects, as in *give John an ice-cream*); S Aux V (as in *John will eat*); S Aux V O; S Aux V O1 O2; Adv S V (where Adv is an Adverb, like *quickly*); Adv S V O; Adv S V O1 O2; Adv S Aux V; Adv S Aux V O; and Adv S Aux V O1 O2.

Fig. 3.6 of the appendix gives a complete list of all degree-0 (unembedded) sentences (expressions) for each of the eight different grammars in this simple system. As shown in the table, English and French correspond to the language L_5, Bengali and Hindi correspond to L_7 while German and Dutch correspond to L_8.

The Markov chain for the 3-parameter example

Suppose the target language is SVO (Subject Verb Object, or "English" setting #5=[0 1 0]). Within the Gibson and Wexler 3-parameter system, there are $2^3 = 8$ possible hypotheses, so we can draw this as an 8-point Markov configuration space, as shown in Fig. 3.5. The shaded rings represent increasing distance in parameter space (Hamming distances) from the target. Each labeled circle is a Markov state, a possible array of parameter settings or grammar, hence specifies a possible target language. Each state is exactly 1 binary digit away from its possible transition neighbors. Each labeled, directed arc between the points is a possible transition from state i to state j, where the labels are the transition probabilities; note that the probabilities from all arcs exiting a state sum to 1. We shall show how to compute these probabilities immediately below. The target grammar, a double circle, lies at the center. This corresponds to the (English) SVO language. Surrounding the bulls-eye target are the three other parameter arrays that differ from [0 1 0] by one binary digit each; we picture these as a ring 1 Hamming distance away from the target: [0, 1, 1], corresponding to Gibson and Wexler's parameter setting #6 in their figure 3 (Spec-first, Comp-final, +V2, basic order SVO+V2); [0 0 0], corresponding to Gibson and Wexler's setting #7 (Spec-first, Comp-first, −V2, basic order SOV); and [1 1 0], Gibson and Wexler's setting #1 (Spec-final, Comp-final, −V2, basic order VOS).

Around this inner ring lie three parameter setting hypotheses, all 2 binary digits away from the target: [0 0 1], [1 0 0], and [1 1 1] (grammars #2, 3, and 8 in Gibson and Wexler's figure 3). Finally, one more ring out, three binary digits different from the target, is the hypothesis [1 0 1], corresponding to target grammar 4.

It is easy to see from inspection of the figure that there are exactly two Absorbing States in this Markov chain, that is, states that have no exit arcs with non-zero probability. One absorbing state is the target grammar (by definition). The other absorbing state is state 2 (corresponding to language VOS+V2, i.e., [1 1 1]). Finally, state 4 (parameter setting [1 0 1]), while not an absorbing state in itself, has no path to the target. It has arcs that lead only to itself or to state 2 (an absorbing state which is not the target). These two states correspond to the local maxima at the head of Gibson and Wexler's figure 4. Hence this target language is *not* learnable. In addition to these local maxima, the next section below shows that there are in fact other states from which the learner will, with high probability, never reach the correct target.

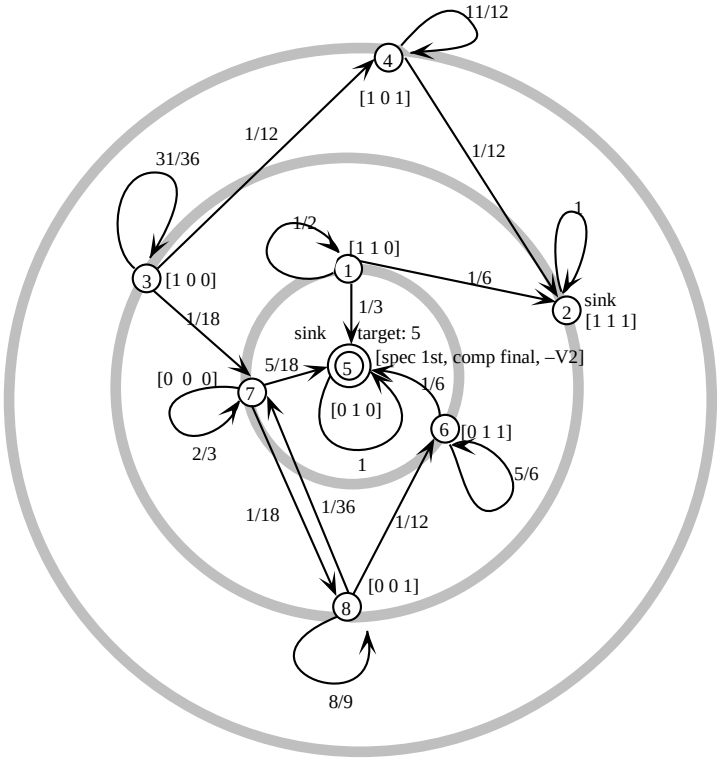

Figure 3.5: The 8 parameter settings in the GW example, shown as a Markov structure. Directed arrows between circles (states, parameter settings, grammars) represent possible nonzero (possible learner) transitions. The target grammar (in this case, number 5, setting [0 1 0]), lies at dead center. Around it are the three settings that differ from the target by exactly one binary digit; surrounding those are the 3 hypotheses two binary digits away from the target; the third ring out contains the single hypothesis that differs from the target by 3 binary digits. Note that the learner can either stay in the same state or step in or out one ring (binary digit) at a time, according to the single-step learning hypothesis; but some transitions are not possible because there is no data to drive the learner from one state to the other under the TLA. Numbers on the arcs denote transition probabilities between grammar states; these values are not computed by the original GW algorithm. The next section shows how to compute these values, essentially by taking language set intersections.

3.4.3 Derivation of the transition probabilities for the Markov TLA structure

We have discussed in the previous section how the behavior of the TLA can be modeled as a Markov chain. The argument is incomplete without a characterization of the transition probabilities of the associated Markov chain. We first provide an example and follow it with a formal exposition.

Example. Consider again the 3-parameter system in Fig. 3.5 with target language 5. What is the probability that the learner will move from state 8 to state 6 ? The learner will make such a transition if it receives a sentence that is analyzable according to the parameter settings of state 6, but not according to the parameter settings of state 8. For example, a sentence of the form (S V O1 O2) as in *Peter gave John an ice-cream* could drive the learner to change its parameter settings from 8 to 6. If one assumes a probability distribution with which sentences from the target are presented to the learner, one could find the total probability measure of all such sentences and use it to calculate the appropriate transition probability.

Formalization

The computation of the transition probabilities from the language family can be done by a direct extension of the procedure given in Gibson and Wexler (1994). Let the target language L_t consist of the strings $s_1, s_2, ...$, i.e.,

$$L_t = \{s_1, s_2, s_3, ...\}$$

Let there be a probability distribution P on these strings. Suppose the learner is in a state s corresponding to the language L_s. Consider some other state k corresponding to the language L_k. What is the probability that the TLA will update its hypothesis from L_s to L_k after receiving the next example sentence? First, observe that due to the single valued constraint, if k and s differ by more than one parameter setting, then the probability of this transition is zero. In fact, the TLA will move from s to k *only if* the following two conditions are met: (1) the next sentence it receives (say, ω occurring with probability $P(\omega)$) is analyzable by the parameter settings corresponding to k and not by the parameter settings corresponding to s; and (2) the TLA has a choice of n parameters to flip on not being able to analyze ω and it happens to pick the one which would move it to state k.

Event 1 occurs with probability $\sum_{\omega \in (L_k \setminus L_s) \cap L_t} P(\omega)$. This is simply the probability measure associated with all strings ω that are both in the target

L_t and L_k but not in the language L_s (the learner's currently hypothesized language). Event 2 occurs with probability $1/n$, since the parameter to flip is chosen uniformly at random out of the n possible choices. Thus the co-occurrence of both these events yields the following expression for the total probability of transition from s to k after one step:

$$\mathbb{P}[s \to k] = \sum_{\omega \in (L_k \setminus L_s) \cap L_t} (1/n) P(\omega)$$

Since the total probability over all the arcs out of s (including the self-loop) must be 1, we obtain the probability of remaining in state s after one step as:

$$\mathbb{P}[s \to s] = 1 - \sum_{k \text{ is a neighboring state of } s} \mathbb{P}[s \to k]$$

In other words, the probability of remaining in state s is 1 minus the probability of moving to any of the other (neighboring) states.

Finally, given any parameter space with n parameters, we have 2^n languages. Fixing one of them as the target language L_t we obtain the following procedure for constructing the corresponding Markov chain. Note that this is simply the Gibson and Wexler procedure for finding local maxima, with the addition of a probability measure on the language family.

- [Assign distribution] Fix a probability measure P on the strings of the target language L_t.
- [Enumerate states] Assign a state to each language i.e., each L_i.
- [Normalize by the target language] Intersect all languages with the target language to obtain for each i, the language $L'_i = L_i \cap L_t$. Thus with state i associated with language L_i, we now associate the language L'_i.
- [Take set differences] For any two states i and k, $i \neq k$, if they are more than 1 Hamming distance apart, then the transition $\mathbb{P}[i \to k] = 0$. If they are 1 Hamming distance apart then $\mathbb{P}[i \to k] = \frac{1}{n} P(L'_k \setminus L'_i)$. For $i = k$, we have $\mathbb{P}[i \to i] = 1 - \sum_{j \neq i} \mathbb{P}[i \to j]$.

Remark. This model captures the dynamics of the TLA completely. We note that the learner's movement from one language hypothesis to another is driven by purely extensional considerations—that is, it is determined by

set differences between language pairs. A detailed investigation of this point is beyond the scope of this chapter.

Example (continued): For our three parameter system, we can follow the above procedure to calculate set differences and build the Markov figure straightforwardly. For example, consider $\mathbb{P}[8 \to 6]$; we compute $(L_6 \setminus L_8) \cap L_5 = \{$ S V O1 O2, S Aux V O, S Aux V O1 O2$\}$. This set has three degree-0 sentences. Assuming a uniform distribution on the 12 degree-0 strings of the target L_5, we obtain the value of the transition from state 8 to state 6 to be $\frac{1}{3}(3/12) = \frac{1}{12}$. Further, since the normalized language L'_1 for state 1 is the empty set, the set difference between states 1 and 5 $(L'_5 \setminus L'_1)$ yields the entire target language, so there is a (high) transition probability from state 1 to state 5. Similarly, since states 7 and 8 share some target language strings in common, such as S V, and do not share others, such as Adv S and S V O, the learner can move from state 7 to 8 and back again.

3.5 Conclusions

In this chapter we have provided a brief account of the linguistic reasoning that lies behind the Principles and Parameters approach to linguistic theory (Chomsky, 1981). We have considered a psycho-linguistically motivated algorithm for language acquisition within this framework. Once the mathematical formalization has been given many additional properties of this particular learning system now become evident. For example, an issue that is amenable to analysis in the current formalization has to do with the existence of subset/superset pairs of languages. The existence of such pairs does not alter the procedure by which the Markov chain is computed, nor does it alter the validity of our main learnability theorem. However, it is clear by our analysis, that if the target happens to be a subset language, the superset language will correspond to an absorbing state. This is because all target sentences are analyzable by the superset language and if the error-driven learner happens to be at the state corresponding to it, it will never exit. This additional absorbing state automatically implies non-learnability by our theorem. However, following Gibson and Wexler and others working in this tradition, we will assume that such complications do not typically arise in the parametric systems under discussion in the current and the next chapter.

It is now easy to imagine other alternatives to the TLA that will avoid the local maxima problem: we can vary any of the five aspects of the language

learning models we described at the beginning of this chapter. To take just one example, as it stands, the learner is allowed to change only one parameter setting at a time. If we relax this condition so that the learner can change more than one parameter at a time, i.e., the learner can conjecture hypotheses far from its current one (in parameter space), then the problem with local maxima disappears. It is easy to see that in this case, there can be only one Absorbing State, namely the target grammar. All other states have exit arcs (under the previous assumption of no subset/superset relations). Thus, by our main theorem, such a system *is* learnable.

As another variant, consider the possibility of noise—that is, occasionally the learner gets strings that are not in the target language. Gibson and Wexler state (fn. 4) that this is not a problem: the learner need only pay attention to frequent data. But this is of course a serious problem for the model; *how* is the learner to "pay attention" to frequent data? Unless some kind of memory or frequency-counting device is added, the learner cannot know whether the examples it receives are noise or not. If the learner is memoryless, then there is always some finite probability, however small, of escaping a local maximum. Clearly, the memory window has to be large enough to ensure that sufficient statistics are computable to distinguish noise from relevant data. A serious investigation of this issue is beyond the scope of this chapter.

To explore these and other possible variations systematically, we will return, in the next chapter, to the 5-way classification scheme for learning models introduced at the beginning of this chapter. We consider first details about sample complexity. Next, we turn to questions about the distribution of the input data, and ask how this changes the sample complexity results. We also consider more realistic input distributions — in particular, those obtained using statistics computed from the CHILDES corpus (MacWhinney, 1996). Finally, we briefly consider sample complexity issues if the learning algorithms operate in batch rather than on-line mode.

Needless to say, the Principles and Parameters framework discussed here represents a very particular approach to describing the class $\mathcal{H}$ of possible natural language grammars within which learning algorithms like the Triggering Learning Algorithm have been formulated. In the next chapter, we will also see how the learning framework developed in this context is general enough to accomodate a wider variety of approaches to the problem of language acquisition. The ability to characterize rates of learning within the Markov framework developed here will take on an added significance as we move on in subsequent chapters to study the problem of language change

and evolution.

Chapter Appendix

3.6 Unembedded Sentences For Parametric Grammars

Fig. 3.6 provides the unembedded (degree-0) sentences from each of the 8 grammars (languages) obtained by setting the 3 parameters of example 1 to different values. The languages are referred to as L_1 through L_8.

3.7 Proof of Learnability Theorem

To establish the theorem, we recall three additional standard terms associated with the Markov chain states: (1) equivalent states; (2) recurrent states; and (3) transient states. We then present another standard result about the *form* of any Markov chain: its *canonical decomposition* in terms of closed, equivalent, recurrent, and transient states.

3.7.1 Markov state terminology

Definition 13 *Given a Markov chain M, and any pair of states $s, t \in M$, we say that s is* **equivalent** *to t if and only if s is reachable from t and t is reachable from s, where by* reachable *we mean that there is a path from one state to another.*

Two states s and t are equivalent if and only if there is a path from s to t and a path from t to s. Using the equivalence relation defined above, we can divide any M into equivalence classes of states. All the states in one class are reachable (from and to) the states in that class.

Definition 14 *Given a Markov chain M, a state $s \in M$ is* **recurrent** *if the chain returns to s in a finite number of steps with probability 1.*

Definition 15 *Given a Markov chain M, and a state $s \in M$, if s is not recurrent, then s is* **transient**.

We will need later the following simple property about transient states:

Lemma 1 *Given a Markov chain M, if t is a transient state of M, then, for any state $s \in M$*

$$\lim_{n\to\infty} p_{st}(n) = 0$$

where $p_{st}(n)$ denotes the probability of going from state s to state t in exactly n steps.

Proof: (*Sketch*) Proposition 2.6.3 (page 88) of Resnick (1992) states that

$$\sum_{n=1}^{\infty} p_{st}(n) < \infty$$

Therefore, $\sum p_{st}(n)$ is a convergent series. Thus $p_{st}(n)_{n\to\infty} \to 0$. ■

3.7.2 Canonical Decomposition

A particular Markov chain might have many closed states (see Definition 12 earlier in text), and these need not be disjoint; they might also be subsets of each other. However, even though there can be many closed states in a particular Markov chain, the following standard result shows that there is a canonical decomposition of the chain (Lemma 2) that will be useful to us in proving the learnability theorem.

Lemma 2 *Given a Markov chain M, we may decompose M into disjoint sets of states as follows:*

$$M = T \cup C_1 \cup C_2 \ldots$$

where (i) T is a collection of transient states and (ii) the C_i's are closed, equivalence classes of recurrent states.

Proof: This is a standard Markov chain result; see Corollary 2.10.2 of page 99 of Resnick (1992). ■

We can now proceed to a proof of the main learnability theorem.

3.7.3 Proof of Main Theorem

$\Rightarrow$. We need to show that if the target grammar is learnable, then every closed set in the chain must contain the target state. By assumption, target grammar g_f is learnable. Now assume for sake of contradiction that there is some closed set C that does not include the target state associated with the

target grammar. If the learner starts in some $s \in C$, by the definition of a closed set of states, it can never reach the target state. This contradicts the assumption that g_f was learnable.

$\Leftarrow$ Assume that every closed set of the Markov chain associated with the learning system includes the target state. We now need to show that the target grammar is Gold-learnable. First, we make use of some properties of the target state in conjunction with the canonical decomposition of Lemma 2 to show that every non-target state must be transient. Then we make use of Lemma 1 about transient states to show that the learner must converge to the target grammar in the limit with probability 1.

First, note the following properties of the target state:

(i) by construction, the target state is an absorbing state, i.e., no other state is reachable from the target state;

(ii) therefore, no other state can be in an equivalence relation with the target state and the target state is in an equivalence class by itself;

(iii) the target state is recurrent since the chain returns to it with probability 1 in one step (the target state is an absorbing state).

These facts about the target state show that the target state consititutes a closed class (say C_i) in the canonical decomposition of M. However, there cannot be any *other* closed class $C_j, j \neq i$ in the canonical decomposition of M. This is because by the definition of the canonical decomposition any other such C_j must be disjoint from C_i, and by the hypothesis of the theorem, such C_j must contain the target state, leading to a contradiction. Therefore, by the canonical decomposition lemma, every *other* state in M must belong to T, and must therefore be a transient state.

Now denote the target state by s_f. The canonical decomposition of M must therefore be in the form:

$$T \cup \{s_f\}.$$

Without loss of generality, let the learner start at some arbitrary state s. After any integer number n of positive examples, we know that,

$$\sum_{t \in M} p_{st}(n) = 1$$

because the learner has to be in *one* of the states of the chain M after n examples with probability 1. But by the decomposition lemma and our

previous arguments $M = T \cup s_f$. Therefore we can rewrite this sum as two parts, one corresponding to the transient states and the other corresponding to the final state:

$$\sum_{t \in T} p_{st}(n) + p_{ss_f}(n) = 1$$

Now take the limit as n goes to infinity. By the transient state lemma, every $p_{st}(n)$ goes to zero for $t \in T$. There are only a finite (known) number of states in T. Therefore, $\sum_{t \in T} p_{st}(n)$ goes to zero. Consequently, p_{ss_f} goes to 1. But that means that the learner converges to the target state in the limit (with probability 1). Since this is true irrespective of the starting state of the learner, the learner converges to the target with probability 1, and the associated target grammar g_f is Gold-learnable.

■

Language	Spec	Comp	V2	Degree-0 unembedded sentences
L_1	1	1	0	"V S" "V O S" "V O1 O2 S" "AUX V S" "AUX V O S" "AUX V O1 O2 S" "ADV V S" "ADV V O S" "ADV V O1 O2 S" "ADV AUX V S" "ADV AUX V O S" "ADV AUX V O1 O2 S"
L_2	1	1	1	"S V" "S V O" "O V S" "S V O1 O2" "O1 V O2 S" "O2 V O1 S" "S AUX V" "S AUX V O" "O AUX V S" "S AUX V O1 O2" "O1 AUX V O2 S" "O2 AUX V O1 S" "ADV V S" "ADV V O S" "ADV V O1 O2 S" "ADV AUX V S" "ADV AUX V O S" "ADV AUX V O1 O2 S"
L_3	1	0	0	"V S" "O V S" "O2 O1 V S" "V AUX S" "O V AUX S" "O2 O1 V AUX S" "ADV V S" "ADV O V S" "ADV O2 O1 V S" "ADV V AUX S" "ADV O V AUX S" "ADV O2 O1 V AUX S"
L_4	1	0	1	"S V" "O V S" "S V O" "S V O2 O1" "O1 V O2 S" "O2 V O1 S" "S AUX V" "S AUX O V" "O AUX V S" "S AUX O2 O1 V" "O1 AUX O2 V S" "O2 AUX O1 V S" "ADV V S" "ADV V O S" "ADV V O2 O1 S" "ADV AUX V S" "ADV AUX O V S" "ADV AUX O2 O1 V S"
L_5 (English, French)	0	1	0	"S V" "S V O" "S V O1 O2" "S AUX V" "S AUX V O" "S AUX V O1 O2" "ADV S V" "ADV S V O" "ADV S V O1 O2" "ADV S AUX V" "ADV S AUX V O" "ADV S AUX V O1 O2"
L_6	0	1	1	"S V" "S V O" "O V S" "S V O1 O2" "O1 V S O2" "O2 V S O1" "S AUX V" "S AUX V O" "O AUX S V" "S AUX V O1 O2" "O1 AUX S V O2" "O2 AUX S V O1" "ADV V S" "ADV V S O" "ADV V S O1 O2" "ADV AUX S V" "ADV AUX S V O" "ADV AUX S V O1 O2"
L_7 (Bengali, Hindi)	0	0	0	"S V" "S O V" "S O2 O1 V" "S V AUX" "S O2 O1 V AUX" "ADV S V" "ADV S O V" "ADV S O2 O1 V" "ADV S V AUX" "ADV S O V AUX" "ADV S O2 O1 V AUX"
L_8 (German, Dutch)	0	0	1	"S V" "S V O" "O V S" "S V O2 O1" "O1 V S O2" "O2 V S O1" "S AUX V" "S AUX O V" "O AUX S V" "O1 AUX S O2 V" "O2 AUX S O1 V" "ADV V S" "ADV V S O" "ADV V S O2 O1" "ADV AUX S V" "ADV AUX S O V" "S AUX O2 O1 V" "S O V AUX" "ADV AUX S O2 O1 V"

Figure 3.6: A list of all the degree-0 (unembedded) expressions for each of eight different grammatical types. The eight grammatical types (denoted by L_1 through L_8) are obtained by setting the *spec*, *comp*, and $V2$ parameters respectively. Expressions are not lexicalized but are denoted as strings over grammatical categories. These categories are S (Subject), O (Object), V (Verb), ADV (Adverb), AUX (Auxiliary verb), $O1$ (Direct Object), $O2$ (Indirect Object) for the sentence types shown.

Chapter 4

Language Acquisition – More Computational Details

In this chapter, we continue our exploration of the Markov chain framework for the analysis of learning algorithms for language acquisition. There are two main themes to this exploration. First, we see how the framework allows us to get a theoretical handle on the important question of *rates of learnability*. Not only must the learner converge to the target in the limit, it must do so at psychologically plausible rates. We develop this point in the next few sections. Second, we see in later parts of this chapter, that *all* learning algorithms may be characterized by *inhomogeneous* Markov chains. More significantly, however, the cognitively interesting class of memory limited learning algorithms may be ultimately characterized by first order Markov chains. We explore this in subsequent parts of this chapter. We conclude finally with a brief overview of computational work in language acquisition that engages the research traditions in linguistics and psychology at varying levels of detail.

4.1 Characterizing Convergence Times for the Markov Chain Model

The Markov chain formulation gives us some distinct advantages in theoretically characterizing the language acquisition problem. First, we have already seen how given a Markov Chain one could investigate whether or not it has exactly one absorbing state corresponding to the target grammar. This is equivalent to the question of whether any local maxima exist. One could

also look at other issues (like stationarity or ergodicity assumptions) that might potentially affect convergence. Later we will consider several variants to the TLA and analyze them formally within the Markov framework. We will also see that these variants do not suffer from the local maxima problem associated with GW's TLA.

Perhaps the significant advantage of the Markov chain formulation is that it allows us to also analyze convergence times. Given the transition matrix of a Markov chain, the problem of how long it takes to converge has been well studied. This question is of crucial importance in learnability. Following GW, we believe that it is not enough to show that the learning problem is *consistent* i.e., that the learner will converge to the target in the limit. We also need to show, as GW point out, that the learning problem is *feasible*, i.e., the learner will converge in "reasonable" time. This is particularly true in the case of finite parameter spaces where consistency might not be as much of a problem as feasibility. The Markov formulation allows us to address the feasibility question. It also allows us to clarify the assumptions about the behavior of data and learner inherent in such an approach. We begin by considering a few ways in which one could formulate the question of convergence times.

4.1.1 Some Transition Matrices and Their Convergence Curves

Let us begin by following the procedure detailed in the previous chapter to explicitly calculate a few transition matrices. Consider the three parameter example that was informally considered before. The target grammar was grammar 5 (according to our numbering of the languages in Fig. 3.6). For simplicity, let us first assume a uniform distribution on the degree-0 strings in L_5, i.e., the probability the learner sees a particular string s_j in L_5 is $1/12$ because there are 12 (degree-0) strings in L_5. We can now compute the transition matrix as the following, where 0's occupy matrix entries if not otherwise specified:

		To							
		L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8
From	L_1	$\frac{1}{2}$	$\frac{1}{6}$			$\frac{1}{3}$			
	L_2		1						
	L_3			$\frac{3}{4}$	$\frac{1}{12}$			$\frac{1}{6}$	
	L_4		$\frac{1}{12}$		$\frac{11}{12}$				
	L_5					1			
	L_6					$\frac{1}{6}$	$\frac{5}{6}$		
	L_7					$\frac{5}{18}$		$\frac{2}{3}$	$\frac{1}{18}$
	L_8						$\frac{1}{12}$	$\frac{1}{36}$	$\frac{8}{9}$

Notice that both 2 and 5 correspond to absorbing states; thus this chain suffers from the local maxima problem. Note also (following Fig. 3.5 as well) that state 4 exits either to itself or to state 2 and is also a problematic initial state. For a given transition matrix T, it is possible to compute[1]

$$T_\infty = \lim_{m \to \infty} T^m.$$

If T is the transition probability matrix of a chain, then T_{ij}, i.e. the element of T in the ith row and jth column is the probability that the learner moves from state i to state j in one step. It is a well-known fact that if one considers the corresponding 'i, j element of T^m then this is the probability that the learner moves from state i to state j in *exactly* m steps. Correspondingly, the i, jth element of T_∞ is the probability of going from initial state i to state j "in the limit" as the number of examples goes to infinity. For learnability to hold irrespective of which state the learner starts in, the probability that the learner reaches state 5 should tend to 1 as m goes to infinity. This means that column 5 of T_∞ should consist of 1's, and the matrix should contain 0's everywhere else. Actually we find that T^m converges to the following matrix as m goes to infinity:

[1]The limiting matrix is not always guaranteed to exist. The existence of a limiting distribution is equivalent to the chain being ergodic. For precise conditions on ergodicity, see Isaacson and Madsen, 1976 or any standard text on Markov chains. For our discussion, we will assume that a limit exists, and barring pathological conditions, it does for the applications we consider.

$$T_\infty = \begin{array}{c} \\ \textbf{From} \end{array} \begin{array}{c} \textbf{To} \\ \begin{array}{cccccccc} L_1 & L_2 & L_3 & L_4 & L_5 & L_6 & L_7 & L_8 \end{array} \\ \begin{bmatrix} 0 & \frac{2}{5} & 0 & 0 & \frac{3}{5} & 0 & 0 & 0 \\ & 1 & & & & & & \\ & \frac{2}{5} & & & \frac{3}{5} & & & \\ & 1 & & & & & & \\ & & & & 1 & & & \\ & & & & 1 & & & \\ & & & & 1 & & & \\ & & & & 1 & & & \end{bmatrix} \end{array} \begin{array}{c} \\ L_1 \\ L_2 \\ L_3 \\ L_4 \\ L_5 \\ L_6 \\ L_7 \\ L_8 \end{array}$$

Examining this matrix we see that if the learner starts out in states 2 or 4, it will certainly end up in state 2 in the limit. These two states correspond to local maxima grammars in the GW framework. If the learner starts in either of these two states, it will never reach the target. From the matrix we also see that if the learner starts in states 5 through 8, it will certainly converge in the limit to the target grammar.

The situation regarding states 1 and 3 is more interesting, and not covered in Gibson and Wexler (1994). If the learner starts in either of these states, it will reach the target grammar with probability 3/5 and reach state 2 (the other absorbing state) with probability 2/5. Thus we see that local maxima (states unconnected to the target) are *not* the only problem for learnability. As a consequence of our stochastic formulation, we see that there are initial hypotheses from which triggered paths exist to the target, however the learner will not take these paths with probability one. In our case, because of the uniform distribution assumption, we see that the path to the target will only be taken with probability 3/5. By making the distribution more favorable, this probability can be made larger, but it can never be made one.

This analysis considerably increases the number of problematic initial states from that presented in Gibson and Wexler (1994). While the broader implications of this are not clear, it certainly renders moot some of the linguistic[2] implications of GW's analysis.

[2]For example, GW rely on "connectedness" to obtain their list of local maxima. From this (incorrect) list, noticing that all local maxima were +Verb Second (+V2), they argued for ordered parameter acquisition or "maturation". In other words, they claimed that the V2 parameter was more crucial, and had to be set earlier in the child's language acquisition

Obviously one can examine other details of this particular system. However, let us now look at a case where there is no local maxima problem. This is the case when the target languages have verb-second (V2) movement in GW's 3-parameter case. Consider the transition matrix (shown below) obtained when the target language is L_1. Again we assume a uniform distribution on strings of the target.

		To							
		L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8
From	L_1	1							
	L_2	$\frac{1}{6}$	$\frac{5}{6}$						
	L_3	$\frac{5}{18}$		$\frac{2}{3}$	$\frac{1}{18}$				
	L_4		$\frac{3}{36}$	$\frac{1}{36}$	$\frac{8}{9}$				
	L_5	$\frac{1}{3}$				$\frac{23}{36}$	$\frac{1}{36}$		
	L_6		$\frac{5}{36}$				$\frac{31}{36}$		
	L_7			$\frac{1}{18}$				$\frac{11}{12}$	$\frac{1}{36}$
	L_8				$\frac{1}{18}$				$\frac{17}{18}$

Here we find that T^m does indeed converge to a matrix with 1's in the first column and 0's elsewhere. Consider the first column of T^m. It is of the form:

$$(p_1(m), p_2(m), p_3(m), p_4(m), p_5(m), p_6(m), p_7(m), p_8(m))'$$

Here $p_i(m)$ denotes the probability of being in state 1 at the end of m examples for the case in which the learner started in state i. For learnability, we naturally want

$$\lim_{m \to \infty} p_i(m) = 1$$

and for the example at hand this is indeed the case. Fig. 4.1 shows a plot of the following quantity as a function of m, the number of examples.

$$p(m) = \min_i \{p_i(m)\}$$

The quantity $p(m)$ is easy to interpret. For example, $p(m) = 0.95$ means that for every initial state of the learner the probability that it is in the target state after m examples is at least 0.95. Further, there is one initial state (the worst initial state with respect to the target, which in our example is L_8)

process. Our analysis shows that this is incorrect, an example of how computational analysis can aid the search for adequate linguistic theories.

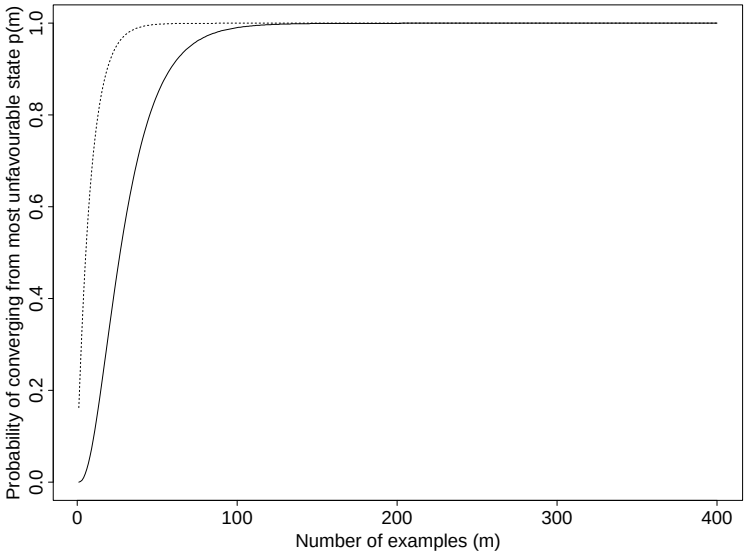

Figure 4.1: Convergence as a function of number of examples. The horizontal axis denotes the number of examples received and the vertical axis represents the probability of converging to the target state. The data from the target is assumed to be distributed uniformly over degree-0 sentences. The solid line represents TLA convergence times and the dotted line is a random walk learning algorithm (RWA). Note that random walk actually converges *faster* than the TLA in this case.

for which this probability is exactly 0.95. We find on looking at the curve that the learner converges with high probability within 100 to 200 (degree-0) example sentences, a psychologically plausible number. One can now of course proceed to examine actual transcripts of child input to calculate convergence times for more realistic distributions of examples.

Now that we have made a first attempt to quantify the convergence time, several other questions can be raised. How does convergence time depend upon the distribution of the data? How does it compare with other kinds of Markov structures with the same number of states? How will the convergence time be affected if the number of states increases, i.e the number of parameters increases? How does it depend upon the way in which the parameters relate to the surface strings? Are there other ways to characterize convergence times? We now proceed to answer some of these questions.

4.1.2 Absorption Times

In the previous section, we computed the transition matrix for a fixed (in principle, this could be arbitrary) distribution and characterized the rate of convergence in a certain way. In particular, we plotted $p(m)$, (the probability of converging from the most unfavorable initial state) against m (the number of samples). However, this is not the only way to characterize convergence times. Given an initial state, the time taken to reach the absorption state (known as the absorption time) is a random variable. One can compute the mean and variance of this random variable. For the case when the target language is L_1, we have seen that the transition matrix has the form:

$$T = \begin{pmatrix} 1 & 0 \\ R & Q \end{pmatrix}$$

Here Q is a 7-dimensional square matrix. The mean absorption times from states 2 through 8 is given by the vector (see Isaacson and Madsen (1976))

$$\mu = (I - Q)^{-1}\mathbf{1}$$

where $\mathbf{1}$ is a 7-dimensional column vector of ones. The vector of second moments is given by

$$\mu' = (I - Q)^{-1}(2\mu - \mathbf{1}).$$

Using this result, we can now compute the mean and standard deviation of the absorption time from the most unfavorable initial state of the learner. (We note that the second moment is fairly skewed in such cases and so is not

Learning scenario	Mean abs. time	St. Dev. of abs. time
TLA (uniform)	34.8	22.3
TLA ($a = 0.99$)	45000	33000
TLA ($a = 0.9999$)	4.5×10^6	3.3×10^6
RW	9.6	10.1

Table 4.1: Mean (col. 1) and Standard Deviation (col. 2) of absorption times to the target state for TLA with different distributions and the Random Walk Algorithm. See text for more explanation.

symmetric about the mean, as may be seen from the previous curves.) The four learning scenarios considered are the TLA with uniform, and increasingly malicious distributions (discussed later), and the random walk (also discussed later).

4.1.3 Eigenvalue Rates of Convergence

We have shown how to characterize learnability by Markov chains. Recall that Markov chains corresponding to memoryless learning algorithms have an associated transition matrix T. We saw that T^k was the transition matrix after k examples, and in the limiting case,

$$\lim_{k \to \infty} T^k = T_\infty.$$

In general, the structure of T_∞, as discussed earlier, determined whether the target grammar was learnable with probability 1. The rate at which T converges to T_∞ determines the rate at which the learner converges to the target "in the limit". This rate allows us to bound the sample complexity in a formal sense, i.e., it allows us to bound the number of examples needed before the learner will be at the target with high confidence. In this section, we develop some formal machinery borrowed from classical Markov chain theory that is useful to bound the rate of convergence of the learner to the target grammar for learnable target grammars. We first develop the notion of an eigenvalue of a transition matrix and show how this can be used to construct an alternative representation of T^k. We then discuss the limiting distributions of Markov chains from various initial conditions, and finally combine all these notions to formally state some results for the rate at which the learner converges to the target.

Eigenvalues and Eigenvectors

Many properties of a transition matrix can be characterized by its eigenvalues and eigenvectors.

Definition 16 *A number λ is said to be an eigenvalue of a matrix T if there exists some nonzero vector $\mathbf{x}'$ satisfying*

$$\mathbf{x}'T = \lambda\mathbf{x}'.$$

Such a row vector $\mathbf{x}'$ is called a left eigenvector of T corresponding to the eigenvalue λ. Similarly, a nonzero column vector $\mathbf{y}$ satisfying $T\mathbf{y} = \lambda\mathbf{y}$ is called a right eigenvector of T.

It can be shown that the eigenvalues of a matrix T can be obtained by solving

$$|\lambda I - T| = 0 \tag{4.1}$$

where I is the identity matrix and $|M|$ denotes the determinant of the matrix M.

Example: Consider the matrix

$$T = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{2}{3} \end{bmatrix}$$

Such a matrix could, for example, be the transition matrix for a learner in a parametric space with two grammars, i.e., a space defined by *one* boolean valued parameter. In order to solve for the eigenvalues of the matrix, we need to solve

$$|\lambda I - T| = \left| \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} - \begin{bmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{2}{3} \end{bmatrix} \right| = 0$$

This reduces to the quadratic equation

$$(\lambda - \frac{2}{3})^2 = \frac{1}{9}$$

which can be solved to yield $\lambda = 1$ and $\lambda = \frac{1}{3}$ as its two solutions. It can be easily seen that the row vector, $\mathbf{x} = (1, 1)$ is a left eigenvector corresponding to the eigenvalue $\lambda = 1$. As a matter of fact, all multiples of $(1, 1)$ are

eigenvectors for this particular eigenvalue. Similarly, it can also be seen that $\mathbf{x} = (1, -1)$ is a left eigenvector for the eigenvalue $\lambda = \frac{1}{3}$. ■

In general, for an $m \times m$ matrix T, Eq. 4.1 is an mth order equation and can be solved to yield m solutions (complex-valued) for λ. Two other facts about eigenvalue solutions of such transition matrices are worth noting here.

1. For transition matrices corresponding to finite Markov chains, it is possible to show that $\lambda = 1$ is always an eigenvalue. Further, it is the largest eigenvalue in that any other eigenvalue, λ, is less than one in absolute value, i.e., $|\lambda| < 1$.

2. For transition matrices corresponding to finite Markov chains, the multiplicity of the eigenvalue $\lambda = 1$ is equal to the number of closed classes in the chain.

In our example above, we do see that $\lambda = 1$ is an eigenvalue. It has multiplicity of 1, indicating that there is only one closed class in the chain; in the example, the class consists of the two states of the chain.

Representation of T^k

The eigenvalues and associated eigenvectors can be used to represent T^k in a form that is convenient for bounding the rate of its convergence to T_∞. This representation is only true for matrices that are of full rank, i.e., $m \times m$ matrices that have m linearly independent left eigenvectors.

Let T be an $m \times m$ transition matrix. Let it have m linearly independent left eigenvectors $\mathbf{x}'_1, \ldots, \mathbf{x}'_m$ corresponding to eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_m$. One could then define the matrix L whose rows are the left eigenvectors of the matrix T. Thus

$$L = \begin{bmatrix} \mathbf{x}'_1 \\ \mathbf{x}'_2 \\ . \\ . \\ \mathbf{x}'_m \end{bmatrix}$$

Clearly, since the rows of L are linearly independent, its inverse, L^{-1} exists. It turns out that the columns of L^{-1} are the right eigenvectors of T. Let the ith column of L^{-1} be $\mathbf{y}_i$; i.e.,

$$L^{-1} = \begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_m \end{bmatrix}$$

Now we can represent T^k in a convenient form stated in the following lemma:

Lemma 3 *Let T be an $m \times m$ transition matrix having m linearly independent left eigenvectors, $\mathbf{x}'_1, \ldots, \mathbf{x}'_m$ corresponding to eigenvalues $\lambda_1, \ldots, \lambda_m$. Further let L be the matrix whose rows are the left eigenvectors and let the columns of L^{-1} be $\mathbf{y}_i$'s. Then*

$$T^k = \sum_{i=1}^{m} \lambda_i^k \mathbf{y}_i \mathbf{x}'_i$$

Thus, according to the lemma above, T^k can be represented as the linear combination of m fixed matrices $(\mathbf{y}_i \mathbf{x}'_i)$. The coefficients of this linear combination are λ_i^k. Clearly, we see that the rate of convergence of T^k is now bounded by the rate of convergence of terms like λ_i^k.

Example (contd.) Continuing our previous example, we can construct the matrices, L and L^{-1} out of the left eigenvectors. In fact using our solutions from before, we see that

$$L = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad \text{and } L^{-1} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \end{bmatrix}$$

The rows of L are the $\mathbf{x}_i$'s and the columns of L^{-1} are the $\mathbf{y}_i$'s. ■

Initial Conditions and Limiting Distributions

Recall that the learner could start in any initial state. One could quantify the initial condition of the learner by putting a distribution on the states of the Markov chain according to which the learner picks its initial state. Let this be denoted by the row vector $\Pi_0 = (\pi_1(0), \pi_2(0), \ldots, \pi_m(0))$. Thus, $\pi_i(0)$ is the probability with which the learner picks the ith state as the initial state. For example, if the learner were equally likely to start in any state, then $\pi_i(0) = \frac{1}{m}$ for all i.

The above characterizes the probability with which the learner is in each of the states before having seen any examples. The learner would then move from state to state according to the transition matrix T. After k examples, the probability with which the learner would be in each of the states is given by:

$$\Pi_k = \Pi_0 T^k$$

Finally, one could characterize the limiting distribution as

$$\Pi = \lim_{k\to\infty} \Pi_k = \Pi_0 T_\infty \tag{4.2}$$

Clearly, Π characterizes the probability with which the learner is in each of the states "in the limit". Suppose the target were L_1, and it were Gold-learnable; then the first element of the vector Π would be 1 and all other elements would be 0. In other words, the probability that the learner is at the target in the limit is 1 and the probability that the learner is at some other state (non-target) in the limit is correspondingly 0.

Rate of Convergence

We are interested in bounding the rate at which Π_k converges to Π. We see that this rate depends on the rate at which T^k converges to T_∞ (Eq. 4.2) which in turn depends upon the rates at which the λ_i^k's converge to 0 by Lemma 3 (for $i > 1$). As we have discussed, $\lambda_1 = 1$. Consequently, we can bound the rate of convergence by the rate at which the second largest eigenvalue converges to 0. Thus we can state the following theorem.

Theorem 11 *Let the transition matrix characterizing the behavior of the memoryless learner be T. Further, let T have the eigenvalues, $\lambda_1, \ldots, \lambda_m$, and m linearly independent left eigenvectors, $\mathbf{x}'_1, \ldots, \mathbf{x}'_m$ and m right eigenvectors $\mathbf{y}_1, \ldots, \mathbf{y}_m; \lambda_1 = 1$ Then, the distance between the learner's state after k examples and its state in the limit is given by:*

$$\| \Pi_k - \Pi \| = \| \sum_{i=2}^{n} \lambda_i^k \Pi_0 \mathbf{y}_i \mathbf{x}'_i \| \leq \max_{2\leq i\leq n} \{|\lambda_i|^k\} \sum_{j=2}^{n} \| \Pi_0 \mathbf{y}_j \mathbf{x}'_j \|$$

Let us first apply this theorem to the illustrative example of this section.
Example (contd.) We have already solved for the eigenvalues of T and constructed the matrices L and L^{-1}. The rows of L are the row vectors $\mathbf{x}'_i$ and the columns of L^{-1} are the column vectors $\mathbf{y}_i$. Assuming that the learner is three times as likely to start in state 1 as compared to state 2, i.e., $\Pi_0 = (\frac{3}{4}, \frac{1}{4})$, we can show that :

$$\| \Pi_k - \Pi \| \leq \left(\frac{1}{3}\right)^k (\frac{1}{2})$$

Thus the rate at which the learner converges to the limiting distribution over the state space is of the order of $(\frac{1}{3})^k$. Note that $\frac{1}{3}$ is the second largest eigenvalue of the transition matrix. ■

Transition Matrix Recipes:

The above discussion allows us to see how one could extract useful learnability properties of the memoryless learner from the transition matrix characterizing the behavior of that learner on the finite parameter space. As a matter of fact, we can now outline a procedure whereby one could check for the learnability and sample complexity of learning in such parameter spaces.

1. Construct the transition matrix T for the memoryless learner according to the arguments developed earlier. Such a matrix has 2^n states if there are n boolean valued parameters in the grammatical theory.

2. Compute the eigenvalues of the matrix T.

3. If the multiplicity of the eigenvalue $\lambda = 1$ is more than one, then there are additional closed classes and by the learnability theorem, the target grammar is not Gold-learnable.

4. If the target is Gold-learnable, and the eigenvectors are linearly independent, then use Theorem 11 to bound the rate of convergence. If the eigenvectors are not linearly independent, then one will need to project into the appropriate subspace of lower dimension and compute the rates in the subspace. See Isaacson and Madsen (1976) for general details and Rivin and Komarova (2003) for specific calculations pertaining to these kinds of learning algorithms.

Using such a procedure, we can bound the rate of convergence of each of the following learning scenarios for the three parameter syntactic subsystem we have examined in some detail in previous examples. In each case, the target is the language L_1. The learning algorithm is the TLA with different sentence distributions (parameterized by a with b, c, d chosen to make sentences outside of A equally likely; see next section). We also considered the Random Walk Algorithm (no greediness, no single value; see next section) with a uniform sentence distribution. The rate of convergence is denoted as a function of the number of examples.

4.2 Exploring Other Points

We have developed, by now, a complete set of tools to characterize learnability and sample complexity of memoryless algorithms working on finite

Learning scenario	Rate of Convergence
TLA (uniform)	$O(0.94^k)$
TLA($a = 0.99$)	$O((1 - 10^{-4})^k)$
TLA($a = 0.9999$)	$O((1 - 10^{-6})^k)$
Random Walk	$O(0.89^k)$

Table 4.2: Bounds on the rate of convergence to the target for TLA under different distributional assumptions and the Random Walk Algorithm. k is the number of examples. We see how the second eigenvalue changes for each of these cases.

parameter spaces. We applied these tools to a specific learning problem which corresponded to a single point in our 5-dimensional space — a point previously investigated by Gibson and Wexler. We also provided an account of how our new analysis revised some of their conclusions and had possible applications to linguistic theory. Here we now explore some other points in the space. In the next section, we consider varying the learning algorithm, while keeping other assumptions about the learning problem identical to that before. Later, we vary the distribution of the data.

4.2.1 Changing the Algorithm

As one example of the power of this approach, we can compare the convergence time of TLA to other algorithms. TLA observes the single value and greediness constraints. We consider the following three simple variants by dropping either or both of the Single Value and Greediness constraints:

Random walk with neither greediness nor single value constraints: We have already seen this example before. Suppose the learner is in a particular state. Upon receiving a new sentence, it remains in that state if the sentence is analyzable. If not, the learner moves uniformly at random to any of the other states and stays there waiting for the next sentence. This is done without regard to whether the new state allows the sentence to be analyzed.

Random walk with no greediness but with single value constraint: The learner remains in its original state if the new sentence is analyzable. Otherwise, the learner chooses one of the parameters uniformly at random and flips it thereby moving to an adjacent state in the Markov structure.

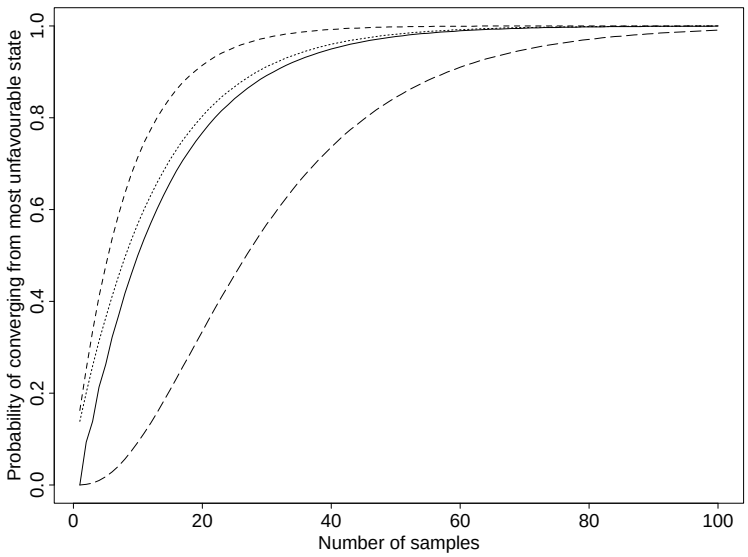

Figure 4.2: Convergence rates for different learning algorithms when L_1 is the target language. The curve with the slowest rate (large dashes) represents the TLA. The curve with the fastest rate (small dashes) is the Random Walk (RWA) with no greediness or single value constraints. Random walks with exactly one of the greediness and single value constraints have performances in between these two and are very close to each other.

Again this is done without regard to whether the new state allows the sentence to be analyzed. However since only one parameter is changed at a time, the learner can only move to neighboring states at any given time.

Random walk with no single value constraint but with greediness: The learner remains in its original state if the new sentence is analyzable. Otherwise the learner moves uniformly at random to any of the other states and stays there iff the sentence can be analyzed. If the sentence cannot be analyzed in the new state the learner remains in its original state.

Fig. 4.2 shows the convergence times for these three algorithms when L_1 is the target language. Interestingly, all three perform better than the TLA for this task (learning the language L_1). More generally, it is found

that the variants converge faster than the TLA for every target language. Further, they do not suffer from local maxima problems. In other words, the class of languages is not learnable by the TLA, but is by its variants. This is another striking consequence of our analysis. The TLA seems to be the "most preferred algorithm" by psychologists. The failure of the TLA to learn the 3-parameter space was used to argue for maturational theories, alternate parameterizations, and parameter ordering.

In view of the fact that the failure of the TLA can be corrected by fairly simple alterations[3], one should examine the conceptual support (from psychologists) for the TLA more closely before drawing any serious linguistic implications. This remains yet another example of how the computational perspective can allow us to rethink cognitive assumptions. Of course, it may be that the TLA has empirical support, in the sense of independent evidence that children do use this procedure (given by the pattern of their errors, etc.).

4.2.2 Distributional Assumptions

In an earlier section we assumed that the example data was generated according to a uniform distribution on the sentences of the target language. We computed the transition matrix for a particular target language and showed that convergence times were of the order of 100-200 samples. In this section we show that the convergence times depend crucially upon the distribution. In particular we can choose a distribution that will make the convergence time as large as we want. Thus the distribution-free convergence time for the 3-parameter system is infinite.

As before, we consider the situation where the target language is L_1. There are no local maxima problems for this choice. We begin by letting the distribution be parameterized by the variables a, b, c, d where

$$\begin{aligned} a &= P(A = \{\text{Adv V S}\}) \\ b &= P(B = \{\text{Adv V O S, Adv Aux V S}\}) \\ c &= P(C = \{\text{Adv V O1 O2 S, Adv Aux V O S,} \\ &\quad \text{Adv Aux V O1 O2 S}\}) \\ d &= P(D = \{\text{V S}\}) \end{aligned}$$

Thus each of the sets A, B, C and D contain different degree-0 sentences of L_1. Clearly the probability of the set $L_1 \setminus \{A \cup B \cup C \cup D\}$ is $1-(a+b+c+d)$.

[3]Note that we have barely scraped the tip of the iceberg as far as exploring the space of possible algorithms is concerned.

	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8
L_1	1							
L_2	$\frac{1-a-b-c}{3}$	$\frac{2+a+b+c}{3}$						
L_3	$\frac{1-a-d}{3}$		$\frac{2+a+d-b}{3}$	$\frac{b}{3}$				
L_4		$\frac{c}{3}$	$\frac{d}{3}$	$\frac{3-c-d}{3}$				
L_5	$\frac{1}{3}$				$\frac{2-a}{3}$	$\frac{a}{3}$		
L_6		$\frac{b+c}{3}$				$\frac{3-b-c}{3}$		
L_7			$\frac{a+d}{3}$				$\frac{3-2a-d}{3}$	$\frac{a}{3}$
L_8				$\frac{b}{3}$				$\frac{3-b}{3}$

Table 4.3: Transition matrix corresponding to a parameterized choice for the distribution on the target strings. In this case the target is L_1 and the distribution is parameterized according to Section 4.7.2

The elements of each defined subset of L_1 are equally likely with respect to each other. Setting positive values for a, b, c, d such that $a+b+c+d < 1$ now defines a unique probability for each degree(0) sentence in L_1. For example, the probability of (Adv V O S) is $b/2$, the probability of (Adv Aux V O S) is $c/3$, that of (V O S) is $(1-(a+b+c+d))/6$ and so on.

We can now obtain the transition matrix corresponding to this distribution. This is shown in Table 4.3.

Compare this matrix with that obtained with a uniform distribution on the sentences of L_1 in the earlier section. This matrix has non-zero elements (transition probabilities) exactly where the earlier matrix had non-zero elements. However, the value of each transition probability now depends upon a, b, c, and d. In particular if we choose $a = 1/12, b = 2/12, c = 3/12, d = 1/12$ (this is equivalent to assuming a uniform distribution) we obtain the appropriate transition matrix corresponding to a uniform distribution. Looking more closely at the general transition matrix, we see that the transition probability from state 2 to state 1 is $(1-(a+b+c))/3$. Clearly if we make a arbitrarily close to 1, then this transition probability is arbitrarily close to 0 so that the number of samples needed to converge can be made arbitrarily large. Thus choosing large values for a and small values for b will result in large convergence times.

This means that the sample complexity cannot be bounded in a distribution-free sense, because by choosing a highly unfavorable distribution the sample complexity can be made as high as possible. For example, we now give the

convergence curves calculated for different choices of a, b, c, d. We see that for a uniform distribution the convergence occurs within 200 samples. By choosing a distribution with $a = 0.9999$ and $b = c = d = 0.000001$, the convergence time can be pushed up to as much as 50 million samples. (Of course, this distribution is presumably not psychologically realistic.) For $a = 0.99, b = c = d = 0.0001$, the sample complexity is on the order of $100,000$ positive examples.

4.2.3 Natural Distributions–CHILDES CORPUS

Given the distribution of the sample complexity upon distributional assumptions, it is of interest to examine the fidelity of the model using real language distributions. For this purpose we carried out some preliminary experiments using the CHILDES database (Macwhinney, 1991). We have carried out preliminary direct experiments using the CHILDES caretaker English input to "Nina" and German input to "Katrin"; these consist of 43,612 and 632 sentences each, respectively. We note, following well-known results by psycholinguists, that both corpora contain a much higher percentage of **aux**-inversion and **wh**-questions than "ordinary" text (e.g., the LOB): 25,890 questions, and 11, 775 wh-questions; 201 and 99 in the German corpus; but only 2,506 questions or 3.7% out of 53,495 LOB sentences.

To test convergence, an implemented system using a newer version of de Marcken's partial parser (see de Marcken, 1990) analyzed each degree-0 or degree-1 sentence as falling into one of the input patterns SVO, S Aux V, etc., as appropriate for the target language. Sentences not parsable into these patterns were discarded (presumably "too complex" in some sense following a tradition established by many other researchers; see Wexler and Culicover (1980) for details). Some examples of caretaker inputs follow:

this is a book ? what do you see in the book ?

how many rabbits ?

what is the rabbit doing ? (...)

is he hopping ? oh . and what is he playing with ?

red mir doch nicht alles nach !

ja , die schwätzen auch immer alles nach (...)

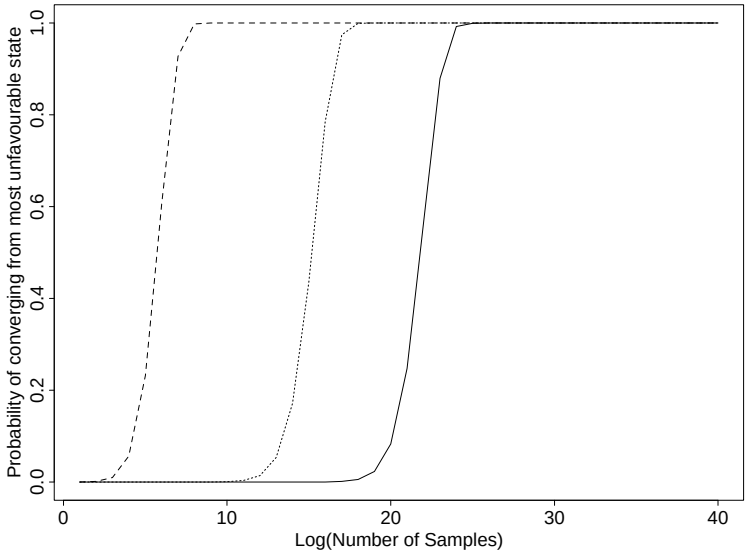

Figure 4.3: Rates of convergence for TLA with L_1 as the target language for different distributions. The y-axis plots the probability of converging to the target after m samples and the x-axis is on a log scale, i.e., it shows $\log(m)$ as m varies. The solid line denotes the choice of an "unfavorable" distribution characterized by $a = 0.9999; b = c = d = 0.000001$. The dotted line denotes the choice of $a = 0.99; b = c = d = 0.0001$ and the dashed line is the convergence curve for a uniform distribution, the same curve as plotted in the earlier figure.

When run through the TLA, we discover that convergence falls roughly along the TLA convergence time displayed in Fig. 4.1–roughly 100 examples to asymptote. Thus, the feasibility of the basic model is confirmed by actual caretaker input, at least in this simple case, for both English and German. One may explore this model with other languages and distributional assumptions. However, there is one very important new complication that must be taken into account: we have found that one must (obviously) add patterns to cover the predominance of auxiliary inversions and wh-questions. However, that largely begs the question of whether the language is verb-second or not. Thus, as far as we can tell, we have not yet arrived at a satisfactory parameter-setting account for V2 acquisition.

4.3 Batch Learning Upper and Lower Bounds: An Aside

So far we have discussed a memoryless learner moving from state to state in parameter space and hopefully converging to the correct target in finite time. As we saw this was well-modeled by our Markov formulation. In this section however we step back and consider upper and lower bounds for learning finite language families if the learner was allowed to remember all the strings encountered and optimize over them. Needless to say this might not be a psychologically plausible assumption, but it can shed light on the information-theoretic complexity of the learning problem.

Consider a situation where there are n languages $L_1, L_2, \ldots L_n$ over an alphabet Σ. Each language can be represented as a subset of Σ^* i.e.

$$L_i = \{\omega_{i1}, \omega_{i2}, \ldots\}; \omega_{ij} \in \Sigma^*$$

The learner is provided with positive data (strings that belong to the language) drawn according to distribution P on the strings of a particular target language. The learner is to identify the target. It is quite possible that the learner receives strings that are in more than one language. In such a case the learner will not be able to uniquely identify the target. However, as more and more data becomes available, the probability of having received only ambigious strings becomes smaller and smaller and eventually the learner will be able to identify the target uniquely. An interesting question to ask then is how many samples does the learner need to see so that with high confidence it is able to identify the target, i.e. the probability that after seeing that

many samples, the learner is still ambigious about the target is less than δ. The following theorem provides a lower bound.

Theorem 12 *The learner needs to draw at least* $M = \max_{j \neq t} \frac{1}{\ln(1/p_j)} \ln(1/\delta)$ *samples (where* $p_j = P(L_t \cap L_j)$*) in order to be able to identify the target with confidence greater than* $1 - \delta$.

Proof: Suppose the learner draws m (less than M) samples. Let $k = \arg\max_{j \neq t} p_j$. This means (1) $M = \frac{1}{\ln(1/p_k)} \ln(1/\delta)$ and (2) that with probability p_k the learner receives a string which is in both L_k and L_t. Hence it will be unable to discriminate between the target and the kth language. After drawing m samples, the probability that all of them belong to the set $L_t \cap L_k$ is $(p_k)^m$. In such a case even after seeing m samples, the learner will be in an ambiguous state. Now $(p_k)^m > (p_k)^M$ since $m < M$ and $p_k < 1$. Finally since $M \ln(1/p_k) = \ln((1/p_k)^M) = \ln(1/\delta)$, we see that $(p_k)^m > \delta$. Thus the probability of being ambiguous after m examples is greater than δ which means that the confidence of being able to identify the target is less than $1 - \delta$. ■

This simple result allows us to assess the number of samples we need to draw in order to be confident of correctly identifying the target. Note that if the distribution of the data is very unfavorable, that is, the probability of receiving ambiguous strings is quite high, then the number of samples needed can actually be quite large. While the previous theorem provides the number of samples *necessary* to identify the target, the following theorem provides an upper bound for the number of samples that are *sufficient* to guarantee identification with high confidence.

Theorem 13 *If the learner draws more than* $M = \frac{1}{\ln(1/b_t)} \ln((N-1)/\delta)$ *samples, then it will identify the target with confidence greater than* $1 - \delta$. *(Here* $b_t = \max_{j \neq t} P(L_t \cap L_j)$*) and* N *is the total number of languages in the family.)*

Proof: Let the target be L_t. We can define A_i to be the event that L_t and L_i are not distinguishable after n events. The probability of event A_i (denoted by $P(A_i)$) is p_i^n where $p_i = P(L_t \cap L_i)$. Thus A_i occurs if all n example sentences belong to both L_t and L_i. Now, the probability that at least one of the events A_i occurs is given by $P(\cup_{j \neq t} A_j)$. Using the union bound, we have $P(\cup_{j \neq t} A_j) \leq \sum_{j \neq t} P(A_j) \leq (N-1) b_t^n$. For this to be smaller than δ, we need $(1/b_t)^n > (N-1)/\delta$ or $n > M = \frac{1}{\ln(1/b_t)} \ln((N-1)/\delta)$. Thus, if more

than M examples are drawn, the probability of being unable to distinguish the target language from any one of the other languages is made small. ■

To summarize, this section provides a simple upper and lower bound on the sample complexity of exact identification of the target language from positive data. The δ parameter that measures the confidence of the learner of being able to identify the target is *suggestive* of a PAC (Valiant, 1984) formulation. However there are two crucial differences. In the PAC formulation, one is interested in an ϵ-approximation to the target language with at least $1 - \delta$ confidence. In our case, this is not so. Since we are not allowed to approximate the target, the sample complexity shoots up with the choice of unfavorable distributions. Second, the learner has to make do with only positive data. In the classical PAC setting, the learner has access to both positive and negative examples. Recalling our discussion of the PAC framework from an earlier chapter, it is worthwhile to note that any finite family of languages is PAC learnable and upper and lower bounds on the sample complexity for learning such families are easily derived following the usual analysis (Vapnik, 1998). We do not explore these sorts of questions any further in the rest of the book.

4.4 Generalizations and Variations

The previous sections introduced and analyzed the Markov chain framework for analyzing the learnability of grammars in the Principles and Parameters (P&P) tradition. We now show that this framework has general applicability well beyond the scope of P&P and the Triggering Learning Algorithm.

4.4.1 Markov Chains and Learning Algorithms

Consider a learning algorithm $\mathcal{A}$ specified by a mapping from

$$\mathcal{D} \rightarrow \mathcal{H}$$

where as before $\mathcal{D}$ is the set of all finite length data streams and $\mathcal{H}$ is a class of hypothesis grammars (languages). The learning algorithm is conceptualized as an on line procedure that develops grammatical hypotheses after each new example sentence. Suppose the learner has an initial hypothesis h_0. After each new sentence has been received, it updates its hypothesis. Let us denote its hypothesis after n examples as h_n. One can then reasonably ask — what is the probability with which the following event happens:

$$\text{Event: } h_n = g; h_{n+1} = f$$

The probability of this event is simply given by the measure of the set

$$A_{n,g} \cap A_{n+1,f} = \{t \in T | \mathcal{A}(t_n) = g\} \cap \{t \in T | \mathcal{A}(t_{n+1}) = f\}$$

Here T is the set of all texts and t is a generic element of this set, i.e., it is a particular text. In accordance with the notation introduced in the earlier chapter, t_n refers to the first n sentences in the text t and therefore $t_n \in \mathcal{D}$. We discussed the natural measure μ_∞ on T that exists by the Kolmogorov extension theorem and thus both sets $A_{n,g}$ and $A_{n+1,f}$ are measurable. Therefore, one may define

$$\mathbb{P}[h_{n+1} = f | h_n = g] = \frac{\mu_\infty(A_{n+1,f} \cap A_{n,g})}{\mu_\infty(A_{n,g})}$$

provided that $\mu_\infty(A_{n,g}) > 0$. If $\mu_\infty(A_{n,g}) = 0$, this means that after n examples have been received, the probability with which the learner will conjecture g at this stage is exactly 0.

We can now naturally define an *inhomogeneous* Markov Chain. The state space corresponds to the set of possible grammars in $\mathcal{H}$ – for each grammar $g \in \mathcal{H}$ we have a state in the chain. At each point in time, the learner has a particular grammatical hypothesis and therefore the chain is in the state corresponding to that grammar. At the nth time step (after n examples have been received), the transition matrix of the chain is given by

$$T_n(g, f) = \mathbb{P}[h_{n+1} = f | h_n = g] = \mathbb{P}[g \to f] = \frac{\mu(A_{n+1,f} \cap A_{n,g})}{\mu_\infty(A_{n,g})} \tag{4.3}$$

Since $T_n(g, f)$ as defined by Eq. 4.3 can be evaluated only for those g for which $\mu_\infty(A_{n,g}) > 0$, we need to specify the values of T_n for other choices of g. To do this it is enough to choose a set of positive numbers α_g such that $\sum_{g \in \mathcal{H}} \alpha_g = 1$. Therefore, we can define

$$T_n(g, f) = \alpha_f \Leftrightarrow \mu_\infty(A_{n,g}) = 0 \tag{4.4}$$

It is easy to check that

$$\forall g, \sum_{f \in \mathcal{H}} T_n(g, f) = 1$$

It is similarly easy to also check that

$$\mathbb{P}[h_{n+1} = f] = \mu_\infty(A_{n+1,f}) = \sum_{g \in \mathcal{H}} \mathbb{P}[h_n = g] T_n(g, f)$$

The transition matrix T_n is time dependent and characterizes the evolution of the learner's hypothesis from example sentence to example sentence.

Thus we make the following observations:

1. Let $\mathcal{H}$ be a class of possible target grammars.
2. Let $g \in \mathcal{H}$ be the target grammar.
3. Let μ be a probability measure on $L_g \subset \Sigma^*$ according to which sentences are presented to the learner.
4. By the Kolmogorov extension theorem, a unique measure μ_∞ exists on the set of all possible texts T as discussed.
5. Any arbitrary learner $\mathcal{A} : \mathcal{D} \to \mathcal{H}$ may be exactly characterized by an inhomogeneous Markov chain with as many states as there are grammars in $\mathcal{H}$ and whose transition matrix T_n (after n steps) is given by Eq. 4.3 and 4.4 respectively.

Wee have thus proved

Theorem 14 *Any deterministic learning algorithm may be characterized by an inhomogeneous Markov chain. The behavior of the chain depends upon the learning algorithm $\mathcal{A}$, the target grammar $g \in \mathcal{H}$ and the probability measure μ on $L_g \in \Sigma^*$. The target grammar g is learnable (with measure 1) if and only if the chain settles in the state corresponding to the target grammar.*

Thus learnability of grammars is related to the convergence of non-stationary Markov chains. In general, such an inhomogeneous chain converges to its limiting distribution if the chain is ergodic. Conditions for the ergodicity of inhomogeneous chains may be expressed in a variety of ways, most notably utilizing the notion of the ergodic coefficient. This, in turn, might allow us to obtain learnability conditions expressed in the language of Markov chains rather than recursion theory. Thus, for example, if the learning algorithm $\mathcal{A}$ is such that (i) the associated Markov chain is ergodic (ii) for each n, T_n is such that all closed sets contain the target state, then the hypothesis generated by $\mathcal{A}$ will converge to the target grammar in the limit. It is worthwhile to note, however, that ergodicity is not necessary for learnability since the chain need not converge to the target state from all initial distributions. A more involved discussion of the relationship between the learnability of grammars and the convergence of the corresponding inhomogeneous chains is beyond the scope of this book.

We turn now to the consideration of the class of memoryless learning algorithms which are characterized by stationary chains. The conditions for the convergence of such chains are obtained from the analysis of the TLA provided earlier.

4.4.2 Memoryless Learners

Memoryless algorithms may be regarded as those that have no recollection of previous data, or previous inferences made about the target function. At any point in time, the only information upon which such an algorithm acts is the current data, and the current hypothesis (state). A memoryless algorithm may then be characterized as an effective procedure mapping this information to a new hypothesis. In general, given a particular hypothesis state (h in $\mathcal{H}$, the hypothesis space), and a new datum (sentence, s in Σ^*), such a memoryless algorithm will map onto a new hypothesis ($g \in \mathcal{H}$). Of course, g could be the same as h or it could be different depending upon the specifics of the algorithm and the datum.

Formally, therefore, the algorithm $\mathcal{A}$ must be such that for all n and for all texts $t \in T$, we have

$$\mathcal{A}(t_{n+1}) = a(\mathcal{A}(t_n), t(n+1))$$

where a is a mapping from $\mathcal{H} \times \Sigma^*$ to $\mathcal{H}$.

Following our previous discussion, the behavior of such an algorithm is also characterized by a Markov chain. It is easy to see that

$$T_n(g, f) = Prob[h_{n+1} = f | h_n = g] = \mu(\{s \in \Sigma^* | a(g, s) = f\})$$

where as before we have assumed that the text is generated by sampling in i.i.d. fashion according to a probability measure μ on Σ^*. Clearly T_n is independent of n and the resulting Markov chain is a stationary one.

4.4.3 The Power of Memoryless Learners

Pure memoryless learners belong to the more general class of *memory limited* learners. Memory limited learners develop grammatical hypothesis based on a finite memory of sentences they have heard over their lifetime. The following definition provides a useful formalization of the notion:

Definition 17 (Wexler and Culicover, 1980) *For any finite data stream $u \in \mathcal{D}$, let u^- be the data stream consisting of all but the last sentence of u.*

Thus if $n = lh(u)$ and $u = s_1, s_2, \ldots, s_n$, then $u^- = s_1, s_2, \ldots, s_{n-1}$. Let u^-k be the data stream consisting of the last k elements of u. Thus if $n > k$ then $u^-k \in \mathcal{D}$ is such that $u^-k = s_{n-(k-1)}, s_{n-(k-2)}, \ldots, s_n$. A learning algorithm $\mathcal{A}$ is said to be k-memory limited if for all $u, v \in \mathcal{D}$, such that (i) $u^-k = v^-k$, and (ii) $\mathcal{A}(u^-) = \mathcal{A}(v^-)$, we have $\mathcal{A}(u) = \mathcal{A}(v)$.

To put it differently, $\mathcal{A}(u)$ depends only upon the previous grammatical hypothesis ($\mathcal{A}(u^-)$) and the last k sentences heard (u^-k). Using this, it is easy to develop the notion of a memory limited learner. This is given by

Definition 18 *A learning algorithm $\mathcal{A}$ is memory limited if there exists some integer m such that $\mathcal{A}$ is m-memory limited.*

Thus, in general, a memory limited learner is required only to have a finite memory. No bound is set on the size of the memory it is required to have. From a cognitive point of view, such memory limited learners have great appeal since it seems like a natural way to characterize the fact that learning children are unlikely to have arbitrary unbounded memory of their data.

One might think that the class of languages learnable by memory limited learners is larger than that learnable by memoryless learners. This, however, turns out not to be the case.

Theorem 15 *If a class of grammars $\mathcal{H}$ is learnable in the limit by a k-memory limited learning algorithm $\mathcal{A}_k$, then there exists some memoryless learning algorithm that is also able to learn it.*

Proof: Omitted. ■

In the last two chapters, we have implicitly adopted a probabilistic model of learning where the learner is required to converge to the target with probability one. It is worthwhile to make the following additional observation

Theorem 16 *If a class of grammars $\mathcal{H}$ is learnable in the limit (in the classical Gold sense), then it is learnable with measure one by some memoryless learner.*

Proof: The proof has been relegated to the appendix for continuity of ideas. ■

Thus, the class of memoryless learners is quite general. Consequently, the characterization of memoryless learners by first order Markov chains takes on a general significance that far exceeds the original context of the Triggering Learning Algorithm.

4.5 Other Kinds of Learning Algorithms

We have examined in some detail the problem of language acquisition in the P&P framework with particular attention to models surrounding the Triggering Learning Algorithm of Gibson and Wexler (1994). By this time, it should be clear, however, that the basic framework that has been applied for a more penetrating analysis of the TLA is considerably more general in its scope. It is therefore timely for us to note that there has been significant computational activity in the area of language acquisition in a variety of linguistic and cognitive traditions. Let us consider a few different organizing strands for this kind of research.

1. In Optimality Theory (Prince and Smolensky, 1993), grammatical variation is treated via constraints rather than rules. Surface expressions, be they phonological forms or syntactic forms are deemed acceptable if they violate the least number of constraints. In many instantiations of this theory, one begins with a finite number of constraints $C_1, \ldots, C_n$. In the grammar of a particular language, these constraints are ordered in importance and determine the ranking and relative importance of constraint violations for candidate surface forms. Thus there are in principle $n!$ different grammars possible. The task of the learning child is conceptualized as determining the appropriate ordering for the target grammar given example sentences they hear. An extensive treatment of the learning algorithms appropriate for this framework is provided in Tesar and Smolensky (2000). Iterative strategies utilizing *error-driven constraint demotion* are online and memoryless and may be exactly characterized by a random walk on a state space of $n!$ possible grammars.

2. Algorithms for learning grammars in different linguistic traditions have been considered by Briscoe (2000) in an LFG framework, Yang (2000) in a GB framework, Stabler in a minimalist framework with movement (1998), Fodor (1994,1998), Sakas (2000), Bertolo (2000), Clark and Roberts (1993) in P&P frameworks, Neumann and Flickinger (1999) in an HPSG framework. They present interesting variations, consider subtleties involved in learning complex grammar families, and investigate issues when one models natural languages as composed of multiple grammars.

3. An important thread in computational studies of language acquisition

attempts to clarify the manner in which semantic considerations enter the process of acquiring a language. Learning procedures rely on semantic feedback to acquire formal structures in a more functionalist perspective on language acquisition. See Feldman etal (1996), Regier (1996), or Siskind (1992) for computational exploration of these themes.

4. Other probabilistic accounts of language acquisition attempt to encode prior knowledge of language to varying degrees of specificity in a minimum description length framework as in Brent (1999) , Goldsmith (2001), DeMarcken (1996).

5. Connectionist and other data driven approaches to language acquisition may also be characterized formally within the frameworks provided in this chapter and the previous one. Examples of such approaches are Daelemans (1996), Gupta and Touretzky (1994), Charniak (1993), MacWhinney (1987, 2004) and so on.

Most of these approaches to language acquisition attempt to characterize in computational terms the procedures of language learning in a variety of cognitive settings with varying degrees of preconceived notions. All of these are ultimately analyzable within the general computational framework considered over the last three chapters.

4.6 Conclusions

In this chapter we have continued our investigation of language acquisition within the P&P framework with central attention to the kinds of models inspired by the TLA. The problem of learning parameterized families of grammars has several different dimensions as we have emphasized earlier. One needs to investigate the learnability for a variety of algorithms, distributional assumptions, parameterizations, and so on. In this chapter, we have emphasized that it is not enough to merely check for learnability in the limit (as previous research within an inductive inference Gold framework has tended to do; see, for example, Jain et al, 1998); one also needs to quantify the sample complexity of the learning problem, i.e., how many examples does the learning algorithm need to see in order to be able to identify the target grammar with high confidence.

In order to get a handle on this question, we take our Markov analysis to the next logical stage — that of characterizing convergence times. A

rich literature exists on characterizing the invariant distributions of Markov chains and the rate at which the chain converges to it. We have provided in this chapter a brief survey of some of the important techniques that are involved in this analysis. We saw the dependence of the convergence rates upon the probability distribution according to which example sentences were presented to the learner. We considered pathological distributions that significantly increased convergence times as well as more natural distributions obtained from the CHILDES corpus.

Although much of the analysis was inspired by the TLA, it is important to recognize that the general framework is considerably broader in scope. Any learning algorithm on any enumerable class of grammars may be characterized as an inhomogeneous Markov chain. Any memory-limited learning algorithm (as biological learning algorithms must be) is ultimately a first order Markov chain. Much of the cognitively motivated computational work on language acquisition — reviewed briefly in Sec. 4.5 — may then be analyzed satisfactorily within this framework.

How a child acquires its native language presents one of the deepest scientific problems in cognitive science today. While we are still quite far from a complete understanding of the process, much research in linguistics, psychology, and artificial intelligence has been conducted with this problem in mind.

Because a natural language with its phonetic distinctions, morphological patterns, and syntactic forms has a certain kind of formal structure, computational modeling has played an important role in helping us reason through various explanatory possibilities for how such a formal system may be acquired. Chapters 2 through 4 of this book present many variations of the basic computational framework and an overview of the central insights that must inform us as we search for a solution.

Throughout these past few chapters, language acquisition is framed in a conventional setting as an idealized parent child interaction with a single homogeneous target grammar (language) that must be attained over the course of this interaction. We use this as a building block for the more natural setting of learners immersed in linguistically heterogeneous populations. By doing so, the problems of language change and evolution on the one hand, and language acquisition on the other, become irrevocably linked. We elaborate on this theme in the rest of the book.

4.7 Appendix: Proofs for Memoryless Algorithms

The power of memoryless algorithms comes from the fact that it is possible for an algorithm to code the data set it has received so far into its current conjecture. This is a consequence of two important and well known facts that we state without proof.

Proposition 1 *There is a mapping $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ that is one-to-one, onto, and therefore invertible.*

Therefore, any pair of natural numbers i, j can be coded as $k = f(i, j)$ such that from knowing the value of k, one is able to decode it by $f^{-1}(k) = (i, j)$. By applying this recursively, one may code any finite number of natural numbers. For example, if one were to code three numbers i, j, k then one may do this by $f(f(i, j), k)$. Applying f^{-1} twice to this number would recover the three original numbers. To make this idea work in general, one will also need to code the total number of numbers being coded. This will indicate to the receiver how many times the f^{-1} operation needs to be applied to the coded number to recover the original numbers. Thus the true code for the numbers i, j, k would be given by $l = f(3, f(f(i, j), k))$. Upon applying f^{-1} once to l, one recovers 3 (the total number of natural numbers being coded). This indicates that f^{-1} needs to be applied two more times to recover the three original numbers. The extension to coding a finite number of natural numbers is clear.

This means that the current data set may be coded as a natural number. Enumerate the elements of Σ^* as $s_1, s_2, s_3, \ldots,$. Let t be a text presented to the learner. Let $i_1, i_2, \ldots, i_n$ be the indices of the first n sentences in the text. Thus $s_{i_1} = t(1), s_{i_2} = t(2), \ldots$, and so on. Then at stage n, the learner will have encountered $t_n = s_{i_1} s_{i_2} \ldots s_{i_n}$. The learner may encode this data by encoding the n integers $i_1, \ldots, i_n$. Let us denote this coding procedure as $l = code(t_n)$.

A second fact is a consequence of the $s - m - n$ theorem. Let $g_1, g_2, \ldots$ be an enumeration of phrase structure grammars (equivalent to the *r.e.* sets) in an acceptable programming system. Let $L_1, L_2, \ldots$ correspond to their respective languages. As is well known, for any *r.e.* set L, there are an infinite number of indices $j_1, j_2, \ldots$, such that $L_{j_k} = L$. Further, an infinite number of such indices may be enumerated by the padding function as the following theorem indicates.

Theorem 17 *There exists a one-to-one, onto, computable function*

$$pad : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$$

such that

$$L_{pad(i,j)} = L_i$$

for all i, j and $pad(i, j)$ is an increasing function of j for each i.

Now recall the basic learning-theoretic setting. Consider an acceptable enumeration of grammars. Then grammars may be specified by specifying their index in this enumeration. Consider $A \subseteq \mathbb{N}$. Then this specifies $\mathcal{G} = \{g_i | i \in A\}$ and $\mathcal{L} = \{L_i | i \in A\}$. Any learning algorithm is a map from data to grammars.

Consider learning according to a prespecified $0-1$ valued metric d such that $d(g_i, g_j) = 0 \Leftrightarrow L_i = L_j$. This is the same as requiring extensional (behavioral) convergence.

Theorem 18 *If a family of grammars $\mathcal{G}$ (correspondingly a family of languages $\mathcal{L}$) is identifiable (on all texts and with an extensional norm) in the limit by an algorithm $\mathcal{A}$, then it is identifiable (on all texts and with an extensional) in the limit by some memoryless algorithm.*

Proof: The memoryless algorithm $\mathcal{A}_{memoryless}$ works by coding the data, calling $\mathcal{A}$ as a subroutine on this data and padding the output of $\mathcal{A}$.

More formally, consider text $t = s_{i_1} s_{i_2} \ldots$ as input to the learning algorithm. The initial guess of the learning algorithm before seeing any data is g_1. After seeing the first data point s_{i_1} the learner calls $\mathcal{A}$ to obtain $g_m = \mathcal{A}(s_{i1})$. It codes the data as $k = code(t_1)$. It uses the padding function to obtain $p = pad(m, k)$. The learner outputs g_p.

At stage n, let the learner's hypothesis be g_j. Let $(i, k) = pad^{-1}(j)$. Then $L_i = L_j$ and $uncode(k) = s_{i_1} \ldots s_{i_n}$. On input $s_{i_{n+1}}$ the learner recovers t_{n+1} by appending it to $uncode(k)$. Thus it has effectively created t_{n+1}.

It calls $\mathcal{A}$ to obtain $g_m = \mathcal{A}(t_{n+1})$. Let $l = code(t_{n+1})$. Finally let $p = pad(m, l)$. The learner outputs g_p.

It is clear that at each stage n, if $g_{m_n} = \mathcal{A}_{memoryless}(t_n)$ and $g_{l_n} = \mathcal{A}(t_n)$ then $L_{m_n} = L_{l_n}$. Therefore if the target language is grammar is g (language L)

$$\lim_{n\to\infty} d(g, g_{m_n}) = \lim_{n\to\infty} d(g, g_{l_n}) = 0$$

■

In the above theorem, we see that while the memoryless learner converges extensionally to the right set, it need not stabilize on any grammar. One may be interested in a stronger notion of convergence in which on each text t the learner stabilizes on a grammar g_t such that $d(g_t, g) = 0$ (here g is the target grammar). An algorithm $\mathcal{A}$ is said to stabilize on a grammar g_t on text t if there exists a point in time n such that for all $m > n$, we have that $\mathcal{A}(t_m) = g_t$, i.e., after seeing n sentences, the learner's grammatical hypothesis is always g_t.

It turns out that while it is not possible to construct a memoryless algorithm that converges in this strong sense on every single text, it is possible to construct one that converges on almost all texts. In other words, the following theorem is true.

Theorem 19 *If a family of grammars $\mathcal{G}$ (correspondingly languages $\mathcal{L}$) is identifiable in the limit (by stabilizing on an appropriate grammar on all texts) by some learning algorithm, it is identifiable with measure one (by stabilizing to an appropriate grammar on almost all texts) by some memoryless learning algorithm.*

Proof: Let $L \in \mathcal{L}$ be the target language and let j be the least index for it, i.e., j is the least index such that $L_j = L$.

Let $t = s_{i_1} s_{i_2} \ldots$ be a text from the target obtained by sampling according to μ in i.i.d. fashion. Note that μ has support on L. We first begin by making the following two observations.

(1) Suppose the text t is such that each element of L occurs infinitely often in t. We call such a text a rich text. Then by the laws of probability, it is possible to show that (here μ_∞ is the product measure on all texts by Kolmogorov Extension Theorem in the standard way).

$$\mu_\infty(\{t | t \text{ is a rich text of } L\}) = 1$$

We will now construct a memoryless learning algorithm that can identify L on all rich texts.

(2) Since $\mathcal{L}$ is identifiable in the limit, by the necessary and sufficient conditions for identifiablity discussed in Chapter 2, we know that for every $L \in \mathcal{L}$, there exists a D_L such that if any L' contains D_L then $L' \not\subset L$.

The description of the memoryless learning algorithm follows. The learner's hypothesis after n examples have been processed will be denoted by the number $h(n)$. $h(n)$ is the index of the grammar ($g_{h(n)}$) or language ($L_{h(n)}$) that is hypothesized after n examples.

The learner will maintain a small evidentiary set S composed of example sentences and a counter c that counts how many times it enters cases 3 and 4 of the algorithm below. Note that the evidentiary set S may be coded as a natural number.

Now consider the following learning algorithm. At stage 0, i.e., after 0 examples have been seen, the algorithm is initialized by having $S = \emptyset, c = 0$. The learner's initial guess is $h(0) = pad(M, f(code(S), c))$ where M is the smallest index k such that $L_k \in \mathcal{L}$.

At stage n, i.e., after n examples have been seen, the learner updates its hypothesis in the following way. Note that from $h(n)$ it can recover S and c uniquely.

if $(D_{L_{h(n)}} \subseteq S \cup \{s_{i_{n+1}}\} \subseteq L_{h(n)})$
  then
  if $(c \leq i_{n+1})$
    then [*Case 1*]
    $h(n+1) = h(n)$
    else [*Case 2*]
    $S = S \cup \{s_{i_{n+1}}\}$
    $p = f(code(S), c)$
    if (S and p don't change)
      then $h(n+1) = h(n)$
      else $h(n+1) = pad(h(n), p)$
    endif
  endif
  else
  $c = c + 1$
  $S = S \cup \{s_{i_{n+1}}\}$
  $p = f(code(S), c)$
  if ($\exists$ smallest $l \leq n$ s.t. $L_l \in \mathcal{L}$ and $D_{L_l} \subseteq S \subseteq L_l$)
    then [*Case 3*]
    h(n+1) = pad(l,p)
    else [*Case 4*]
    h(n+1) = pad(M,p)
  endif
endif

First, we must agree that this is memoryless. To see this, simply notice that the new hypothesis depends only upon the previous hypothesis and

the current data. This is because from the previous hypothesis, both the hypothesized language and the evidentiary set S may be recovered. Second, notice that at all stages, the algorithm outputs a hypothesis that is in the family $\mathcal{L}$.

Next, we prove that if the algorithm stabilizes on a grammar, it stabilizes on one that generates the target language L. Assume that the algorithm converges on index k, i.e., $h(n) = k$ for all n large enough. Since cases 3 and 4 result in change of hypotheses, it must be in case 1 or 2 after a finite number of examples have been seen. From this stage on, it is clear that $D_{L_k} \subseteq S \cup \{t(n)\} \subseteq L_k$ for all n. Since every element of L occurs infinitely often in t, we see that every element of L must be contained in L_k, i.e., $L \subseteq L_k$. On the other hand, $D_{L_k} \subseteq S \cup \{t(n)\} \subseteq L$. Therefore by the definition of D_{L_k} and the learnability of $\mathcal{L}$ we have that L cannot be a proper subset of L_k. Therefore it must be that $L_k = L$.

Finally, we show that it must converge (stabilize). The proof is by contradiction. Suppose not. This means that it changes its hypothesis infinitely often. Therefore cases 2,3,4 must occur infinitely often.

Let us argue that cases 3,4 can occur only a finite number of times. We will prove this by contradiction. Assume it occurs an infinite number of times. Therefore c increases without bound. Consider an arbitrary $s_i \in L$. We know that s_i occurs infinitely often in t. If on any of the instances it occurs, the algorithm enters case 3 or 4, then s_i will be included in S after that point. On the other hand, if on each occasion, it enters case 1 or 2, then the moment $c > i$ (and this moment must come since c increases without bound), the algorithm will enter case 2 at that stage and s_i will be included in S after that point. Thus every $s_i \in L$ will get included in S eventually. Now consider the elements of D_L. Since D_L is a finite set, there is a finite time (stage N) such that after N examples have been received, all elements of D_L have been included in S so that $D_L \subseteq S$.

Let $k = \max(N, j)$. Consider the case where the learner enters case 3 or 4 after stage k. Since $L_j = L$, we have that (i) $D_{L_j} \subseteq S \subseteq L_j$. Consider any $i < j$. We will now argue that the learning algorithm can hypothesize L_i only a finite number of times after this. Suppose not. Then it must be the case that (ii) $D_{L_i} \subseteq S \subseteq L_i$ an infinite number of times. Since every element of $L = L_j$ eventually gets included in S, this means that $L_j \subseteq L_i$. Yet, by the learnability of $\mathcal{L}$, if $D_{L_i} \subseteq S \subseteq L_j$, it cannot be that L_j is a proper subset of L_i. Therefore, it must be that $L_i = L_j$. Yet we know that j is the smallest index for $L_j = L$ leading to a contradiction.

Thus, every hypothesis L_i where $i < j$ will be eventually discarded when the learner enters case 3 or 4 after stage k. Therefore, the learner will eventually hypothesize L_j if it enters case 3 or 4 after stage k. Having done this, it is clear that it will never enter case 4 or case 3 ever after. This leads to a contradiction in our assumption that the algorithm enters case 3 or 4 an infinite number of times. Therefore cases 3 and 4 must occur only a finite number of times and there is a maximum number C which c achieves after which it never grows.

Thus eventually, the learner will only enter case 1 or 2. Now we will argue that the learner's grammatical hypothesis can change only a finite number of times after this. First we note that there are only a finite number of sentences $s_i \in L$ such that $i < C$. The algorithm can enter case 2 only when one of these s_i's occurs in the text. Each of these s_i's will eventually get included in S when the algorithm enters case 2. After this point whether the algorithm is in case 1 or 2, the set S does not change and $c = C$ does not change. Therefore, $h(n)$ does not change with n and the learner converges on a fixed index.

■