MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY
and
CENTER FOR BIOLOGICAL COMPUTATIONAL LEARNING
WHITAKER COLLEGE

A.I. Memo No. 1449 November, 1993
C.B.C.L. Paper No. 86

Formalizing Triggers: A Learning Model for Finite Spaces

Partha Niyogi and Robert C. Berwick

Abstract

In a recent seminal paper, Gibson and Wexler ([1], GW) take important steps to formalizing the notion of language learning in a (finite) space whose grammars are characterized by a finite number of *parameters*. One of their aims is to characterize the complexity of learning in such spaces. For example, they demonstrate that even in finite spaces, convergence may be a problem since it is possible under some single-st gradient ascent methods to remain at a local maximum. From the standpoint of learning theory, however, GW leave open several questions that can be addressed by a more precise formalization in terms of Markov structures (a possible formalization suggested but left unpursued in a footnote of GW). In this paper we explicitly formalize learning in a finite parameter space as a Markov structure whose states ar parameter settings. Several important results that follow directly from this characterization, include corrected version of GW's central convergence proof; (2) an explicit formula for calculating the transit probabilities between hypotheses and the existence of "problem states" in addition to local maxima; (3 an explicit calculation of the time needed to converge, in terms of number of (positive) examples; (the convergence and comparison of several variants of the GW learning procedure, e.g., random walk; (5) batch- and PAC-style learning bounds for the model.

This report describes research done within the Center for Biological and Computational Learning in the Department of Brain and Cognitive Sciences, and at the Artificial Intelligence Laboratory. This research is supported by NSF grant 9217041-ASC and ARPA under the HPCC program. Correspondence by e-mail could be directed to pn@ai.mit.edu or berwick@ai.mit.edu.

1 Introduction: The Triggering Model as a Markov structure

Recently, Gibson and Wexler ([1], GW) have begun to formalize the notion of language learning in a (finite) space whose grammars (and languages) are characterized by a finite number of parameters or 1-dimensional Boolean-valued arrays, n long. 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). One of Gibson and Wexler's aims is to establish that under some simple hill-climbing learning regimes, namely, single-step gradient ascent, some linguistically natural, finite, spaces are unlearnable, in the sense that positive-only examples lead to *local maxima*—incorrect hypotheses from which a learner can never escape. More broadly, they wish to 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.

In this paper, we choose as a convenient starting point their Triggering Learning Algorithm (TLA) to focus our investigation of parameter learning. Our central result is that the performance of this algorithm is completely modeled by a Markov chain. The remainder of the current paper is devoted to exploring the basic consequences of this fact.

Let us first review the GW model and the TLA. Following Gold [2] the basic framework is that of identification in the limit. The learner (child) starts out in an arbitrary *state* = some setting of the n parameter values. The learner (child) receives a (countably infinite) sequence of positive example sentences drawn from some target language, L_t. After each presentation, the learner can either (i) stay in the same state; or (ii) move to a new hypothesis state, using the algorithm given below. If after some finite number of examples the learner converges to the correct target language (= parameter settings) and never changes state, then it has correctly identified the target language; otherwise, it does not converge.

In addition, in the GW model the language learner obeys two fundamental constraints: (1) the *single-value constraint*—the learner can change only 1 parameter value at a time; and (2) the *greediness constraint*—if, the learner is given a positive example it cannot recognize (accept), and if the learner changes one parameter value and finds that it can accept the example, then the learner retains that new parameter value. Finally, we also recall GW's definition of a *local trigger* (minor notational changes aside): given values for all parameters but one, a *local trigger* for value v of parameter p_i is a sentence s from the target grammar such that s is grammatical iff $p_i(v) = v$. GW then state their TLA as follows:

- [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 at time t_i (examples drawn from the language of a single target grammar, $L(G_t)$), from a uniform distribution on the language (we shall be able to 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 gradient-ascent] Select a single parameter at random, uniformly with probability $1/n$, 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*; otherwise go to Step 2;

Of course, this algorithm never halts in the usual sense. GW aim to show under what conditions this algorithm converges "in the limit"—that is, after some number, n, of steps, where n is unknown, the correct target parameter settings will be selected and never be changed. Their central claim is stated as their Theorem 1 (p. 7 in their manuscript).[1]

Theorem 1 *As long as the probability is always greater than a lower bound b ($b > 0$) that the learner will 1) encounter a local trigger for some incorrectly-set parameter P, and 2) then reset P accordingly to the target value, it turns out that the target grammar can always be learned using the Triggering Learning Algorithm*

1.1 The Markov formulation

From the standpoint of learning theory, however, GW leave open several questions that can be addressed by a more precise formalization of this model in terms of Markov chains (a possible formalization suggested but left unpursued in footnote 9 of GW). We can picture the hypothesis space, of size 2^n, as a set of points, each corresponding to one particular array 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 Σ as a subset of Σ^*. One of them is the target language (grammar). We arbitrarily place the (single) target grammar 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 this is the so-called *Hamming distance*. We may further place *weights* on the transitions from state i to state j corresponding to the nonzero b's mentioned in the theorem above; these correspond to the probabilities that the learner will move from hypothesis state i to state j. In fact, as we shall show below, given a distribution over $L(G)$, we can further carry out the calculation of the actual b's themselves. Thus, we

[1] Note that the notion of "trigger" does not enter into the statement of the TLA or the constraints the TLA employs, but only into the statement of the theorem.

can picture the TLA learning space as a directed, labeled graph V with 2^n vertices.[2] More precisely, we can make the following remarks about the TLA system GW describe.

Remark. The TLA system is *memoryless*, that is, given a sequence s of sentences up to time t_i, the selection of hypothesis h depends only on sentence s_i and not (directly) on previous sentences, i.e.,

$$p\{h(s_i) \le x_i | x(t),\ t \le t_1\} = P\{x(t_i) \le x_i | x(t_{i-1})\}$$

In other words, the TLA system is a classical *discrete stochastic process*, in particular, a discrete *Markov process* or Markov chain. We can now use the theory of Markov chains to describe TLA parameter spaces [3]. For example, as is well known, we can convert the graphical representation of an n-dimensional Markov chain M to an $n \times n$ matrix T, where each matrix entry (i, j) represents the transition probability from state i to state j. A single step of the Markov process is computed via the matrix multiplication $T \times T$; n steps is given by T^n. A "1" entry in any cell (i, j) means that the system will converge with probability 1 to state j, given that it starts in state i.

As mentioned, not all these transitions will be possible in general. For example, by the single value hypothesis, the system can only move 1 Hamming 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 (nonzero b) from j to i, and there must be one from i to j. Further, by assumption and the TLA, it is clear that once we reach the target grammar there is nothing that can move the learner from this state, since all remaining positive evidence will not cause the learner to change its hypothesis. Thus, there must be a loop from the target state to itself, with some positive label b and no exit arcs. In the Markov chain literature, this is known as an *Absorbing State* (AS). Obviously, a state that only leads to an AS will also drive the learner to that AS. Finally, if a state corresponds to a grammar that generates some sentences of the target there is always a loop from any state to itself, that has some nonzero probability. Clearly, one can conclude at once the following learnability result:

Theorem 2 *Given a Markov chain C corresponding to a GW TLA learner,* $\exists$ exactly *1 AS (corresponding to the target grammar/language) iff C is learnable.*

Proof. $\Leftarrow$. By assumption, C is learnable. Now assume for sake of contradiction that there is not exactly one AS. Then there must be either 0 AS or >1 AS. In the first case, by the definition of an absorbing state, there is no hypothesis in which the learner will remain forever. Therefore C is not learnable, a contradiction. In the second case, without loss of generality, assume there are exactly two absorbing states, the first S corresponding to the target parameter setting, and the second S' corresponding to some other setting. By the definition of an absorbing state, in the limit C will with some nonzero probability enter S', and never exit S'. Then C is not learnable, a contradiction. Hence our assumption that there is not exactly 1 AS must be false.

$\Rightarrow$. Assume that there exists exactly 1 AS i in the Markov chain M. Then, by the definition of an absorbing state, after some number of steps n, no matter what the starting state, M will end up in state i, corresponding to the target grammar. ∎

Note that this approach avoids a crucial flaw in the proof given in GW (pp. 7-8 in manuscript):

> That is, if the learner never goes through the same state twice, then she is bound to end up in the target state at some point, because the parameter space is finite in size. Thus the probability of avoiding the target state forever is equivalent to the probability of cycling forever through some ordered set of states (a cycle).
>
> We can divide the parameter space into a finite set of minimal cycles, where each minimal cycle contains no cycles as a subpart. Because the parameter space is finite, the set of minimal cycles in the parameter space is also finite. For each minimal cycle, we can now calculate the probability that the learner remains in that cycle forever... the probability of staying in the [minimal pm/rcb] cycle in the limit (forever) is zero. The same is true for all of the finitely-many minimal cycles, so that the probability of staying in any of these cycles in the limit is also zero. *Thus the probability of ending up at the target state in the limit is one.*

In brief, GW attempt to show that the probability of the learner avoiding the target forever is zero by showing that the fact that some minimal cycle occurs infinitely often makes the probability of the infinite sequence zero. In other words every way in which the learner avoids the target has probability zero. Thus they conclude that probability of the event

$$\text{Event} = \text{Learner avoids target forever}$$

is zero, more precisely, they claim,

$$Pr[\cup W_\alpha] = 0$$

where each W_α is a path avoiding the target and $\cup W_\alpha$ is set of all such paths. However, as is well known, this union computation is true iff it is taken over a *countable* number of elements. In the example at hand, the crucial omission in the argument is that the there are an *uncountable* number of ways in which the learner can avoid the target. This is because there are an uncountable number of sequences of numbers between 1 and $M-1$. The base $M-1$ expansion of any real number in the

[2] GW construct an identical transition diagram in the description of their computer program for calculating local maxima. However, this diagram is not explicitly presented as a Markov structure; it does not include transition probabilities. Of course, topologically both structures must be identical.

interval [0, 1) would yield such a sequence (e.g., consider an irrational expansion such as the square root of 2).

Since there are an uncountable number of ways in which the event of avoiding the target forever can be realized, the fact that each such way has probability zero *does not* imply that the total event has probability zero as well. To see this consider a random variable X with a uniform distribution on [0, 1]. Now consider the event:

$$\text{Event: } X < 1/2$$

There are many ways in which this event could occur e.g. $X=1/4$, $X=1/3$, $X=0.234$ etc. Each of these ways has probability zero i.e., $P[X=1/4]=0$, $P[X=1/3]=0$... and so on. However we know that the probability of the event $X<1/2$ is 1/2 not zero. This is because there are an *uncountable* number of ways in which the event $X<1/2$ could take place. Thus the proof as given in [1] is incorrect. One correct way to formulate the proof is by resorting to an explicit Markov formulation, as suggested but not executed in GW's footnote 9, and as we established above. A similar conceptual difficulty seemingly leads to their failure to note that there may be other states *besides* local maxima, for which convergence may not occur.

Corollary 1 *Given a Markov chain corresponding to a (finite) family of grammars in a GW learning system, if there exist 2 or more AS, then that family is not learnable.*

Example.
Consider the GW 3-parameter system. Its binary parameters are: (1) Spec(ifier) first (0) or last (1); (2) Comp(lement) first (0) or last (1); and Verb Second (V2) does not exist (0) or does exist (1). By *Specifier* GW follow the standard linguistic convention of whether there is part of a phrase that "specifies" that phrase, roughly like *the old* in *the old book*; by *Complement* GW roughly mean 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, Object, Direct Object, Indirect Object, Adverb, and Adjective. 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 GW for discussion). Note that the "surface strings" of these languages are actually *phrases* such as Subject, Verb, and Object. Figure (3) of GW 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 GW's figure (3), the other possible arrangements of surface strings corresponding to this parameter setting include SV; SV O1 O2 (two objects, as in *give John an ice-cream*); S Aux V (as in *John is a nice guy*; 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.

Suppose SOV (setting #5=[0 1 0]) is the target grammar (language). With the GW 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 the figure above. The shaded rings represent increasing Hamming distances from the target. Each labeled circle is a Markov state, a possible array of parameter settings or grammar, hence extensionally specifies a possible target language. Each state is exactly 1 binary digit away from its possible transition neighbors. Each directed arc between the points is a possible (nonzero) transition from state i to state j; we shall show how to compute this immediately below. We assume that the target grammar, a double circle, lies at the center. This corresponds to the (English) SOV language. Surrounding the bulls-eye target are the 3 other parameter arrays that differ from [0 1 0] by one binary digit each; we picture these as a ring 1 Hamming bit away from the target: [0, 1, 1], corresponding to GW's parameter setting #6 in their figure 3 (Spec-first, Comp-final, +V2, basic order SVO+V2); [0 0 0], corresponding to GW's setting #7 (Spec-first, Comp-first, –V2), basic order SOV; and [1 1 0], GW's setting #1 (Spec-final, Comp-final, $-V2$], basic order VOS.

Around this inner ring lie 3 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 GW figure 3). Note that by the Single Value hypothesis that the learner can only move one grey ring towards or away from the target at any one step. 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 2 absorbing states in this Markov chain, that is, states that have no exit arcs. One AS is the target grammar (by definition). The other AS is state 2. Finally, state 4 is also a sink (a so-called "closed state" in the Markov terminology) that leads only to state 4 or state 2. These two states correspond to the local maxima at the head of GW's figure 4. Hence this system 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 can never reach the target.

2 Derivation of Transition Probabilities for the Markov TLA Structure

The computation of the transition probabilities from the language family can be computed by a direct extension of the procedure given in GW. Let the target language L_t consist of the strings $s_1, s_2, s_3 \ldots$, i.e.,

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

Let there be a probability distribution P on these strings. Suppose the learner is in a state corresponding to the language L_s. Suppose it now receives the string s_j. It will do so with probability $P(s_j)$. There are two cases to examine depending upon whether or not the string s_j is analyzable by the grammar corresponding to the current parameter setting.

Case I. Suppose the learner can syntactically analyze the received string s_j. By the TLA, it will not change its parameter values. In the Markov chain formulation, the learner remains in the same state. Remember that this state corresponds to the language L_s. Also note that this situation arises only when s_j is in the language L_s. Therefore the probability of the learner remaining in the state s is $P(s_j)$.

Case II. Suppose the learner cannot syntactically analyze the string. Then $s_j \notin L_s$. By the TLA, the learner chooses a parameter at random, flips it, and if the new parameter setting makes s_j analyzable, it retains this value and moves to the corresponding state; otherwise it remains in its original state s. Let us examine this situation using the Markov chain formulation. The learner is in state s. It has n neighboring states each at a Hamming distance of 1 from itself. The learner picks one of these uniformly at random. Imagine that n_j of these neighboring states correspond to languages which contain s_j. If the learner picks any one of these states (which of course it does with probability n_j/n), it would stay in that state. If the learner picks any of the other states (with probability $(n-n_j)/n$) then it remains in state s. Note that n_j of course could be 0 which means that none of the neighboring states would allow the string to be analyzed. The maximum value n_j could take is n. Thus we see that the probability that the learner remains in state s is $P(s_j)((n-n_j)/n)$. The probability that it moves to each of the other n_j states is $P(s_j)(1/n)$.

Clearly this allows us to compute the probability that the learner will remain in its original state s as the sum of the probabilities of the above two cases, namely the following expression:

$$\sum_{s_j \in L_s} P(s_j) + \sum_{s_j \notin L_s} (1 - n_j/n) P(s_j)$$

The above expression is still a little untidy because it has the n_j's in it. We would like to clean it up a little. To do this consider the way we would compute the transition probability of state s to some other neighboring state say k in the chain. From the above analysis, we see that such a transition will occur with probability $1/n$ for all the strings s_j that are in the language L_k but not in the language L_s. The strings themselves occur with probability $P(s_j)$ each and so the transition probability is:

$$P[s \to k] = \sum_{s_j \in L_t, s_j \notin L_s, s_j \in L_k} (1/n) P(s_j)$$

Note that the above summation is done over all strings $s_j \in (L_t \cap L_k) \setminus L_s$ where $\setminus$ is the set difference symbol. It is easy to see that

$$s_j \in (L_t \cap L_k) \setminus L_s \Leftrightarrow s_j \in (L_t \cap L_k) \setminus (L_t \cap L_s).$$

Thus we can rewrite the transition probability as

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

Since we have shown this in generality where for any given target, we can compute the transition probabilities between any two states in the Markov chain formulation of the parameter space, the self-transition probability can now be given as,

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

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 the GW procedure for finding local maxima, with the addition of a probability measure on the language family.

- (Assign distribution) First 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.) Now for any two states i and k, if they are more than 1 Hamming distance apart, then the transition $P[i \to k] = 0$. If they are 1 Hamming distance apart then $P[i \to k] = P(L'_k \setminus L'_i)$.

This model captures the dynamics of the TLA completely.

Example.

Consider again the 3-parameter system in the previous figure with target language 5. We can calculate the following set differences to build the Markov figure straightforwardly.

1. $L_1 \cap L_5 = \emptyset$ (no strings in common between L_1 and target L_5).
2. $L_2 \cap L_5 = \{$S V, S V O, S V O1 O2, S Aux V, S Aux V O, S Aux V O1 O2 $\}$.
3. $L_3 \cap L_5 = \emptyset$.
4. $L_4 \cap L_5 = \{$S V, S V O, S Aux V$\}$.
5. $L_5 \cap L_5 = L_5$.
6. $L_6 \cap L_5 = \{$S V, S V O, S V O1 O2, S Aux V, S Aux V O, S Aux V O1 O2$\}$
7. $L_7 \cap L_5 = \{$S V, Adv S V$\}$.
8. $L_8 \cap L_5 = \{$S V, S V O, S Aux V$\}$.

From these values alone, we can draw the figure illustrated, and find the local maxima. For example, since the normalized state set for state 1 is the emptyset, the set difference between states 1 and 5 gives all of the 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.

Many additional properties of the triggering learning system now become evident once the mathematical formalization has been given. It is easy to imagine other alternatives to the TLA that will avoid the local maxima problem. For example, as it stands the learner only changes a parameter setting if that change allows the learner to analyze the sentence it could not analyze before. If we relax this condition so that in this situation the learner picks a parameter at random to change, then the problem with local maxima disappears, because there can be only 1 Absorbing State, namely the target grammar. All other states have exit arcs. Thus, by our main theorem, such a system *is* learnable.

Or consider for example the possibility of noise—that is, occasionally the learner gets strings that are not in the target language. GW state (fn. 4, p. 5) 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. Unless some kind of memory or frequency-counting device is added, the learner cannot know whether the example it receives is noise or not. This being so, then there is always some finite probability, however small, of escaping a local maximum. It appears that the identification in the limit framework as given is simply incompatible with the notion of noise, unless a memory window of some kind is added.

We may now proceed to ask the following questions about the TLA more precisely:

1. Does it converge?
2. How fast does it converge? How does this vary with distributional assumptions on the input examples?
3. Can we now compute the dynamics for other "natural" parameter systems, like the 10-parameter system for the acquisition of stress in languages developed by [4]?
4. Variants of TLA would correspond to other Markov structures. Do they converge? If so, how fast?
5. How does the convergence time scale up with the number of parameters?
6. What is the computational complexity of learning parametrized language families?
7. What happens if we move from on-line to batch learning? Can we get PAC-style bounds [6]?
8. What does it mean to have non-stationary (nonergodic) Markov structures? How does this relate to assumptions about parameter ordering and maturation?
9. What other parametrizations can we consider?

In the remainder of this paper we shall consider these and other questions. We turn first to the question of convergence and convergence times.

3 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 TLA and see how these can also be formally analyzed within the Markov formulation. 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 attack the feasibility question. It also allows us to clarify the assumptions about the behavior of data and learner inherent in such an attack. We begin by considering a few ways in which one could formulate the question of convergence times.

3.1 Some Transition Matrices and Their Convergence Curves

Let us begin by following the procedure detailed in the previous section to actually obtain a few transition matrices. Consider the example which we looked at informally in the previous section. Here the target grammar was grammar 5 and the L languages have already been obtained. For simplicity, let us first assume a uniform distribution on the 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:

	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8
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 the previous figure as well) that state 4 only exits to either itself or to state 2, hence is also a local maximum. More precisely, 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. This is the probability that the learner moves from state i to state j in m steps. 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^m should contain all 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:

	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8
L_1		$\frac{1}{3}$			$\frac{2}{3}$			
L_2		1						
L_3		$\frac{1}{3}$			$\frac{2}{3}$			
L_4		1						
L_5					1			
L_6					1			
L_7					1			
L_8					1			

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. If the learner starts in either of these states, it will reach the target grammar with probability 2/3 and reach state 2, the other absorbing state with probability 1/3. Thus we see that local maxima are *not* the only problem for learnability. GW (p. 26 in manuscript) focuses exclusively on local maxima, and indirectly implies that these are the only difficult states: "most of the source grammars have local triggers that enable the learner to get to the target. . . however, there exist pairs of source and target grammars from the parameter space given in the table in Figure 3, such that no data from the target grammar will ever shift the learner out of the source grammar. . . There are six such pairs of source local maximum and target grammars" They then go on to list in their figure 4, *two* such local maxima for the target grammar 5, corresponding to states 2 and 4.

While this statement is strictly true, it does not *exhaust* the set of source states that never lead to the target grammar. As we see from the transition matrix, while it is true that states 2 and 4 will, with probability 1, not converge to the target grammar, it is *also* true that states 1 and 3 will not converge to the target. Thus, the number of "bad" initial hypotheses is significantly larger than that presented in Figure 4 of GW. This difference is again due to the new probabilistic framework introduced in the current paper, and in fact is related to the difficulty found earlier with the central convergence proof, not looking just at minimal paths and cycles in fact misses some possible learning paths. In the appendix of this paper, we provide a complete list of all starting states which might result in non-learnability. While the implications of the existence of additional non-learnable starting states is not clear, presumably the issue of learnability even in the 3-parameter case deserves re-examination in light of this possibility.

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 obtained when the target language is L_1. Again we assume a uniform distribution on strings of the target.

	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8
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

$$\begin{bmatrix} p_1(m) \\ p_2(m) \\ p_3(m) \\ p_4(m) \\ p_5(m) \\ p_6(m) \\ p_7(m) \\ p_8(m) \end{bmatrix}$$

Here p_i denotes the probability of being in state 1 at the end of m examples in the case where the learner started in state i. Naturally we want

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

and for this example this is indeed the case. The next figure shows a plot of the following quantity as a function of m, the number of examples.

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

The quantity $p(m)$ is easy to interpret. Thus $p(m) = 0.95$ 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_3) 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 "actual" distributions of examples, and we are currently engaged in this effort.)

As one example of the power of this approach, we can compare the convergence time of TLA to other algorithms. Perhaps the simplest is random walk: start the learner at a random point in the 3-parameter space, and then, if an input sentence cannot be analyzed, move randomly from state to state. Note that this regime does not suffer from the local maxima problem, since there is always some finite probability of exiting a non-target state.

To satisfy the reader's curiosity, we provide the convergence curves for a random walk algorithm (RWA) on the 8 state space. We find that the convergence times are actually faster than for the TLA; see figure 2. Since the RWA is also superior in that it does not suffer from the same local maxima problems as TLA, the conceptual support for the TLA is by no means clear. 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.), but this evidence is lacking, as far as we know.

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.

3.2 Distributional Assumptions

In the earlier section we assumed that the data was uniformly distributed. 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 which 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 parametrized by the variables a, b, c, d where

$$\begin{aligned} a &= P(A = \{\text{Adv VS}\}) \\ b &= P(B = \{\text{Adv VOS, Adv Aux VS}\}) \\ c &= P(C = \{\text{Adv VO1 O2 S, Adv Aux VOS,} \\ &\quad\ \text{Adv Aux VO1 O2 S}\}) \\ d &= P(D = \{\text{VS}\}) \end{aligned}$$

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 $1 - (a + b + c + d)$. 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 $AdvVOS$ is $b/2$, the probability of $AdvAuxVOS$ is $c/3$, that of VOS 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 1.

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 as before. 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.

3.3 Absorption Times

In the previous sections, we computed the transition matrix for a variety of distributions and showed the rate of convergence. 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 [3])

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

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 symmetric about the mean, as may be seen from the previous curves.)

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

3.4 Eigenvalue Rates of Convergence

In classical Markov chain theory, there are also well-known convergence theorems derived from a consideration of the eigenvalues of the transition matrix. We state without proof a convergence result for transition matrices stated in terms of its eigenvalues.

Theorem 3 *Let T be an $n \times n$ transition matrix with n linearly independent left eigenvectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ corresponding to eigenvalues $\lambda_1, \ldots, \lambda_n$. Let $\mathbf{x}_0$ (an n-dimensional vector) represent the starting probability of being in each state of the chain and π be the limiting probability of being in each state. Then after k transitions, the probability of being in each state $\mathbf{x}_0 T^k$ can be described by*

$$\| \mathbf{x}_0 T^k - \pi \| = \| \sum_{i=1}^{n} \lambda_i^k \mathbf{x}_0 \mathbf{y}_i \mathbf{x}_i \| \leq \max_{2 \leq i \leq n} |\lambda_i|^k \sum_{i=2}^{n} \| \mathbf{x}_0 \mathbf{y}_i \mathbf{x}_i \|$$

where the $\mathbf{y}_i$'s are the right eigenvectors of T.

This theorem thus bounds the rate of convergence to the limiting distribution π (in cases where there is only one absorption state, π will have a 1 corresponding to that state and 0 everywhere else). Using this result we can now bound the rates of convergence (in terms of number k of samples) by:

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

This theorem also helps us to see the connection between the number of examples and the number of parameters since a chain with n states (corresponding to an $n \times n$ transition matrix) represents a language family with $\log_2(n)$ parameters.

4 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^*.

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

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 4 *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 the 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 5 *If the learner draws more than $M = \frac{1}{\ln(1/(1-b_t))} \ln(1/\delta)$ samples, then it will identify the target with confidence greater than $1 - \delta$. (Here $b_t = P(L_t \setminus \cup_{j \neq t} L_j)$).*

Proof. Consider the set $L = L_t \setminus \cup_{j \neq t} L_j$. Any element of this set is present in the target language L_t but not in any other language. Consequently upon receiving such a string, the learner will be able to instantly identify the target. After $m > M$ samples, the probability that the learner has not received any member of this set is $(1-P(L))^m = (1-b_t)^m < (1-b_t)^M = \delta$. Hence the probability of seeing some member of L in those m samples is greater than $1-\delta$. But seeing such a member enables the learner to identify the target so the probability that the learner is able to identify the target is greater than $1-\delta$ if it draws more than M samples. ∎

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 [6] formulation. However there is a crucial difference. In the PAC formulation, one is interested in any 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 choice of unfavorable distributions. There are some interesting directions one could follow within this batch learning framework. One could try to get true PAC-style distribution-free bounds for various kinds of language families. Alternatively one could use the exact identification results here for linguistically plausible language families with "reasonable" probability distributions on the data. It might be an interesting exercise to recompute the bounds for cases where the learner receives both positive and negative data. Finally the bounds obtained here could be sharpened further. We intend to look into some of these questions in the future.

5 Variants of the Learning Model

We have so far focused on the TLA scheme for learning. TLA observes the single value and greediness constraints. There could be several variants of this learning algorithm and many of these are captured completely by our Markov formulation. 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. 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. 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 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. Further they do not suffer from local maxima problems. It should be pointed out, however, that the differences from TLA are marginal and this convergence has been shown only for L_1 as the target language. Ideally the convergence rates have to be computed for each target language and then either a worst case or average case rate should be decided upon to characterize the convergence times for the algorithm on the language family as a whole.

6 Conclusion, Open Questions, and Future Directions

As the number of parameters n increases, the size of the corresponding Markov matrix grows as 2^n. Thus in the case of a 10 parameter system as found in models of English stress ([4]) the corresponding Markov structure will be a 1024×1024 matrix. We are currently conducting an analysis of this larger system to find its local maxima, analyze its convergence times, and see if its convergence times correspond to what one might find in practice with real stress systems.

Additional questions remain to be answered. One issue has to do with the "smoothness" relation between the parameter settings and the resulting surface strings. In principles-and-parameters theory, it has often been suggested that a small parameter change could lead to a large deductive change in the grammar, hence a large change in the surface language generated. In all the examples considered so far there is a smooth relation between surface sentences and parameters, in that switching from a V2 to a non-V2 system, for instance, leads us to a Markov state that is not too far away from the previous one. If this is not so, it is not so clear that the TLA will work as before. In fact, the whole question of how to formulate the notion of "smoothness" in a language–grammar framework is unclear. We know in the case of continuous functions, for example, that if the learner is allowed to choose examples (which can be simulated by selective attention), then such an "active" learner can approximate such functions much more quickly than a "passive" learner, like the one presented in GW. Is there an analog to this in the discrete, digital domain of language? How can one approximate a language? Here too mathematics may play a helpful role. Recall that there is an analog to a functional analysis of languages—namely, the algebraic approach advanced by Chomsky and Schutzenberger ([5]). In this model, a language is described by an (infinite) polynomial generating function, where the coefficients on the polynomial term x gives the number of ways of deriving the string x. A (weak, string) approximation to a language can then be defined in terms of an approximation to the generating function. If this method can be deployed, then one might be able to carry over the results of functional analysis and approximation for active vs. passive learners into the "digital" domain of language. If this is possible, we would then have a very powerful set of previously underutilized mathematical tools to analyze language learnability.

7 Acknowledgements

We would like to thank Ken Wexler and Ted Gibson, for valuable discussions that led to this work; all residual errors are ours. This research is supported by NSF grant 9217041-ASC and ARPA under the HPCC program

Appendix

A Learnable Grammars: The Full Story

A.1 Problem States

We provide in Table 2 a complete list of problem states. In other words we list all the initial starting grammar, target grammar pairs for which the learner is not guaranteed to converge to the target with probability 1. In fact, assuming a uniform distribution on the strings for the target grammar, it is possible to compute the probability of not converging to the target for each of these pairs. Note that this probability is non-zero for the pairs listed.

A.2 Remarks

1. We have provided a complete list of initial starting grammars from which some target is not learnable (i.e. learnable with probability 1). We notice that there are three kinds of such problem starting states. Some states correspond to sinks in the Markov Structure with respect to some target grammar. Here the learner gets stuck, never leaves it and correspondingly never converges to the target. Then there are states which are not sinks (OVS+V2 when the target is SVO-V2) but which can only move to some non-target sink, and so never converge to the target. These two kinds of problem states (starred in our table) have been listed by Gibson and Wexler in Fig. 4 (pg. 27 of manuscript). Finally there are states which are not sinks, but which can with a non zero probability converge to some non-target sink. They can also with a non-zero probability converge to the target and in this respect are distinguished from problem states of type 2.

2. We would like to observe that of the 56 possible initial grammar-target grammar combinations possible, 12 result in non-learnable situations in the 3-parameter system investigated here. This is a fairly high density of unfavourable initial configurations. It would be interesting to see how this changes with other lingual subsystems with a larger number of parameters.

3. We also did an analysis of convergence times under uniform distribution for the each target grammar. We find that the results are similar to the results displayed in the paper for the case when the target grammar is (VOS-V2). For cases when the target is learnable, the learner converges to the target in 100-200 samples with high (greater than 0.99) probability. Further, the variants of the TLA all outperform the TLA in terms of convergence times.

References

[1] E. Gibson and K. Wexler, Triggers. Linguistic Inquiry, 1993, to appear.

[2] E. Gold, Language Identification in the Limit. Information and Control 10 (1967) 447–474.

[3] D. Isaacson and J. Masden, Markov Chains, John Wiley, New York, 1976.

[4] B. E. Dresher and J. Kaye, A computational learning model for metrical phonology. Cognition, 1990, 137-195.

[5] N. Chomsky and M. Schutzenberger, The Algebraic Theory of Context-free Languages. Computer Programming and Formal Systems, North Holland, Amsterdam, 1963, 53–77.

[6] L. G. Valiant, A theory of the Learnable. Proc. of the 1984 STOC, 1984, 436-445

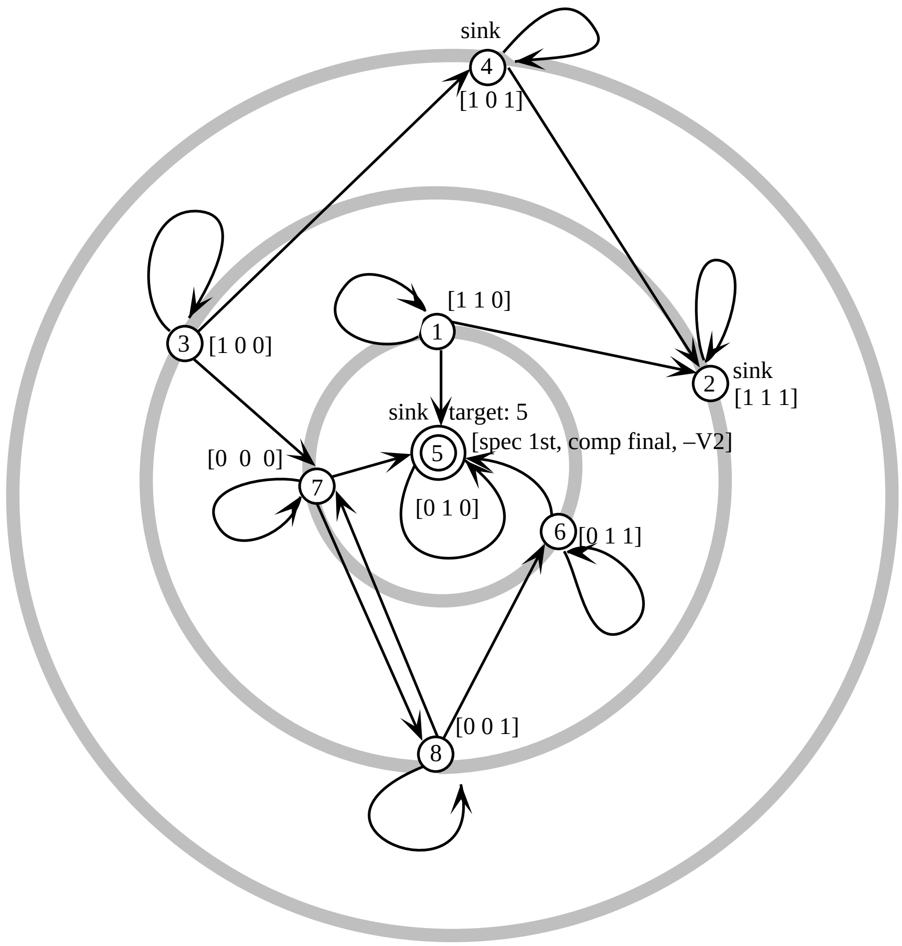

Figure 1: The 8 parameter settings in the GW example, shown as a Markov structure, with transition probabili
omitted. (Without transition probabilities, this diagram corresponds exactly to that in GW's appendix, as me
above.) Directed arrows between circles (states) represent possible nonzero (possible learner) transitions
grammar (in this case, number 5, setting [0 1 0]), lies at dead center. Around it are the three settings th
from the target by exactly one binary digit; surrounding those are the 3 hypotheses two binary digits away fr
target; the third ring out contains the single hypothesis that differs from the target by 3 binary digits.
the learner can either cycle or step in or out one ring (binary digit) at a time, according to the single-ste
hypothesis; but some transitions are not possible because there is no data to drive the learner from one sta
other under the TLA.

	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 1: Transition matrix corresponding to a parametrized choice for the distribution on the target strings
case the target is L_1 and the distribution is parametrized according to Section 3.2.

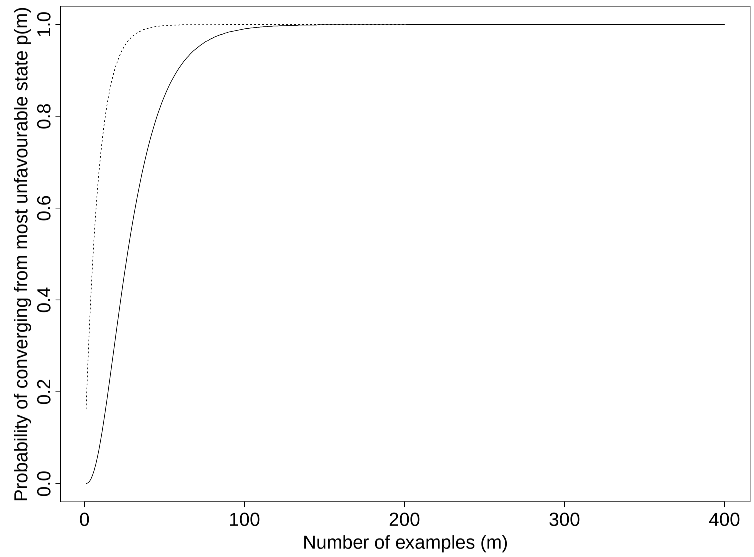

Figure 2: Convergence as function of number of examples. The horizontal axis denotes the number of examp
received and the vertical axis represents the probability of converging to the target state. The data from t
is assumed to be distributed uniformly over degree-0 sentences. The solid line represents TLA convergence
and the dotted line is a random walk learning algorithm (RWA). Note that random walk actually converges *fast*
than the TLA in this case.

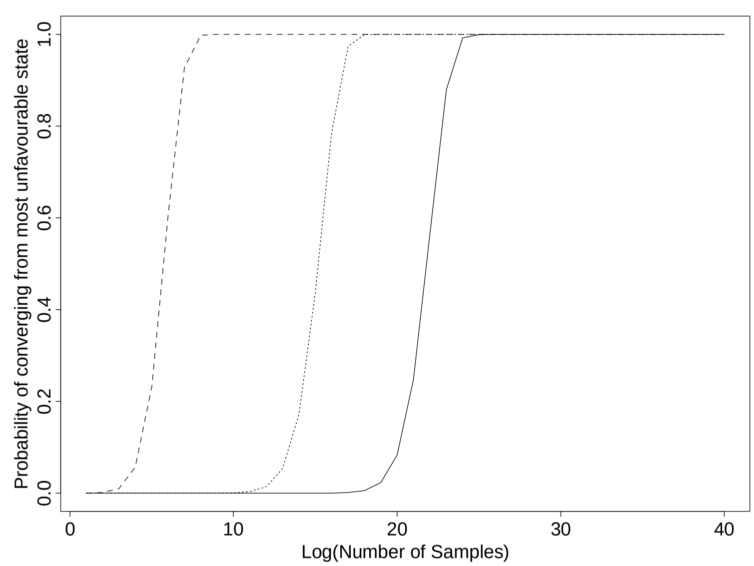

Figure 3: Rates of convergence for TLA with L_1 as the target language for different distributions. The y-axis plots t
probability of converging to the target after m samples and the x-axis is on a log scale, i.e., it shows $\log(m)$ a
The solid line denotes the choice of an "unfavorable" distribution characterized by $a = 0.9999$; $b = c = d = 0.0$
The dotted line denotes the choice of $a = 0.99$; $b = c = d = 0.0001$ and the dashed line is the convergence curve
a uniform distribution, the same curve as plotted in the earlier figure.

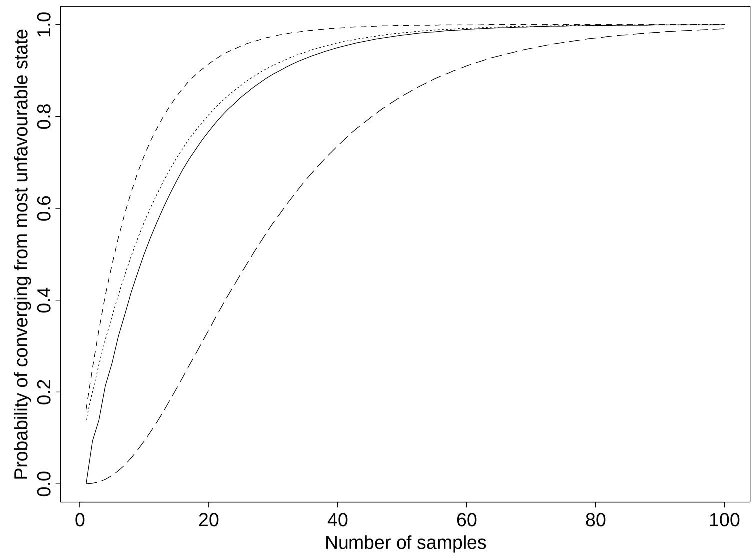

Figure 4: Convergence rates for different learning algorithms when L_1 is the target language. The curve with the slowest rate (large dashes) represents the TLA. The curve with the fastest rate (small dashes) is the Randor (RWA) with no greediness or single value constraints. Random walks with exactly one of the greediness and si value constraints have performances in between these two and are very close to each other.

Initial Grammar	Target Grammar	State of Initial Grammar (Markov Structure)	Probability of Not Converging to Target
(SVO-V2)	(OVS-V2)	Not Sink	0.5
(SVO+V2)*	(OVS-V2)	Sink	1.0
(SOV-V2)	(OVS-V2)	Not Sink	0.15
(SOV+V2)*	(OVS-V2)	Sink	1.0
(VOS-V2)	(SVO-V2)	Not Sink	0.33
(VOS+V2)*	(SVO-V2)	Sink	1.0
(OVS-V2)	(SVO-V2)	Not Sink	0.33
(OVS+V2)*	(SVO-V2)	Not Sink	1.0
(VOS-V2)	(SOV-V2)	Not Sink	0.33
(VOS+V2)*	(SOV-V2)	Sink	1.0
(OVS-V2)	(SOV-V2)	Not Sink	0.08
(OVS+V2)*	(SOV-V2)	Sink	1.0

Table 2: Complete list of problem states, i.e., all combinations of starting grammar and target grammar whicl in non-learnability of the target. The items marked with an asterisk are those listed in the original paper b and Wexler [1].