Feasible Learnability of Formal Grammars and The Theory of Natural Language Acquisition

Naoki ABE
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104-6389

Abstract

We propose to apply a complexity theoretic notion of feasible learnability called "polynomial learnability" to the evaluation of grammatical formalisms for linguistic description. Polynomial learnability was originally defined by Valiant in the context of boolean concept learning and subsequently generalized by Blumer et al. to infinitary domains. We give a clear, intuitive exposition of this notion of learnability and what characteristics of a collection of languages may or many not help feasible learnability under this paradigm. In particular, we present a novel, nontrivial constraint on the degree of "locality" of grammars which allows a rich class of mildly context sensitive languages to be feasibly learnable. We discuss possible implications of this observation to the theory of natural language acquisition.

1 Introduction

A central issue of linguistic theory is the "projection problem", which was originally proposed by Noam Chomsky [?] and subsequently led to much of the development in modern linguistics. This problem poses the question: "How is it possible for human infants to acquire their native language on the basis of casual exposure to limited data in a short amount of time?" The proposed solution is that the human infant in effect "knows" what the natural language that it is trying to learn could possibly be. Another way to look at it is that there is a relatively small set of possible grammars that it would be able to learn, and its learning stratergy, implicitly or explicitly, takes advantage of this apriori knowledge. The goal of linguistic theory, then, is to characterize this set of possible grammars, by specifiying the constraints, often called the "Universal Grammar". The theory of inductive inference offers a precise solution to this problem, by characterizing exactly what collections of (or its dual "constraints on") languages satisfy the requirement for being the set of possible grammars, i.e. are learnable? A theory of "feasible" inference is particularly interesting because the language acquisition process of a human infant *is* feasible, not to mention its relevance to the technological counterpart of such a problem.

In this paper, we investigate the learnability of formal grammars for linguistic description with respect to a complexity theoretic notion of feasible learnability called 'polynomial learnability'. Polynomial learnability was originally developed by Valiant [?], [?] in the context of learning boolean concept from examples, and subsequently generalized by Blumer et al. for arbitrary concepts [?]. We apply this criterion of feasible learnability to subclasses of formal grammars that are of considerable linguistic interest. Specifically, we present a novel, nontrivial constraint on grammars called "k-locality", which enables a rich class of mildly context sensitive grammars called Ranked Node Rewriting Grammars (RNRG) to be feasibly learnable. We discuss possible implications of this result to the theory of natural language acquisition.

2 Polynomial Learnability

2.1 Formal Modeling of Learning

What constitutes a good model of the learning behavior? Below we list five basic elements that any formal model of learning must contain. (c.f. [13])

1. Objects to be learned: Let us call them 'knacks' for full generality. The question of learnability is asked of a *collection* of knacks.
2. Environment: The way in which 'data' are available to the learner.
3. Hypotheses: Descriptions for 'knacks', usually expressed in a certain language.
4. Learners: In general functions from data to hypotheses.
5. Criterion of Learning: Defines precisely what is meant by the question; When is a learner said to 'learn' a given collection of 'knacks' on the basis of data obtained through the environment ?

In most cases 'knacks' can be thought of as subsets of some universe (set) of objects, from which examples are drawn.[1] (Such a set is often called the 'domain' of the learning problem.) The obvious example is the definition of what a language is in the theory of natural language syntax. Syntactically, the English language is nothing but the set of all grammatical sentences, although this is subject to much philosophical controversy. The corresponding mathematical notion of a formal language is one that is *free* of such a controversy. A formal language *is* a subset of the set of all strings in Σ^* for some alphabet Σ. Clearly Σ^* is the domain. The characterization of a knack as a subset of a universe is in fact a very general one. For example, a boolean concept of n variables is a subset of the set of all assignments to those n variables, often written 2^n. Positive examples in this case are assignments to the n variables which 'satisfy' the concept in question.

When the 'knacks' under consideration can in fact be thought of as subsets of some domain, the overall picture of a learning model looks like the one given in Figure 1.

2.2 Polynomial Learnability

Polynomial learnability departs from the classic paradigm of language learning, 'identification in the limit',[2] in at least two important aspects. It enforces a higher demand on the time

[1] First order structures are an example in which languages are more than just subsets of some set [14].

[2] Identification in the limit was originally proposed and studied by Gold [8], and has subsequently been generalized in many different ways. See for example [13] for a comprehensive treatment of this and related paradigms.

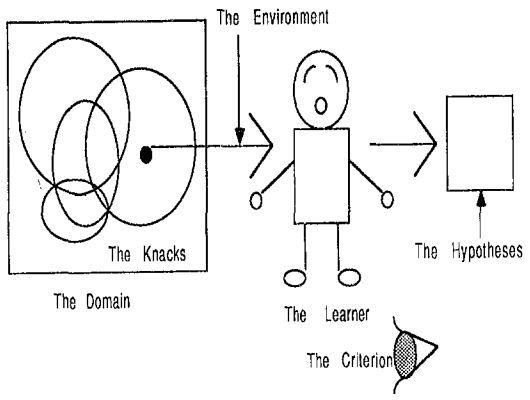

Figure 1: A Learning Model

complexity by requiring that the learner converge in time polynomial, but on the other hand relaxes the criterion of what constitutes a 'correct' grammar by employing an approximate, and probabilistic notion of correctness, or *accracy* to be precise. Furthermore, this notion of correctness is intricately tied to both the time complexity requirement and the way in which the environment presents examples to the learner. Specifically, the environment is assumed to present to the learner examples from the domain with respect to an unknown (to the learner) but fixed probability distribution, and the accuracy of a hypothesis is measured with respect to that same probability distribution. This way, the learner is, so to speak, *protected* from 'bad' presentations of a knack. We now make these ideas precise by specifying the five essential parameters of this learning paradigm.

1. Objects to be learned are languages or subsets of Σ^* for some fixed alphabet Σ. Although we do not specify *apriori* the language in which to express these grammars[3], for each collection of languages $\mathcal{L}$ of which we ask the learnability, we fix a class of grammars $\mathcal{G}$ (such that $L(\mathcal{G}) = \mathcal{L}$ where we write $L(\mathcal{G})$ to mean $\{L(G) \mid G \in \mathcal{G}\}$) with respect to which we will define the notion of 'complexity' or 'size' of a language. We take the number of bits it takes to write down a grammar under a reasonable[4], fixed encoding scheme to be the size of the grammar. The size of a language is then defined as the size of a minimal grammar for it. (For a language L, we write $size(L)$ for its size.)

2. The environment produces a string in Σ^* with a time-invariant probability distribution unknown to the learner and pairs it with either 0 or 1 depending on whether the string is in the language in question or not, gives it to the learner. It repeats this process indefinitely.

3. The hypotheses are expressed as grammars. The class of grammars allowed as hypotheses, say $\mathcal{H}$, is not necessarily required to generate exactly the class $\mathcal{L}$ of languages to be learned. In general, when a collection $\mathcal{L}$ can be learned by a learner which only outputs hypotheses from a class $\mathcal{H}$, we say that $\mathcal{L}$ is learnable *by* $\mathcal{H}$, and in particular, when $\mathcal{L} = L(\mathcal{G})$ is learnable by $\mathcal{G}$, the class of representations $\mathcal{G}$ is said to be *properly* learnable. (See [6].)

4. Learners passively receive an infinite sequence of positive and negative examples in the manner described above, and at each initial (finite) segment of such a sequence, output a hypothesis. In other words, they are functions from finite sequences of positive and negative examples[5] to grammars.

5. A learning function is said to *polynomially learn* a collection of languages just in case it is computable in time polynomial in the length of the input sample, and for an arbitrary degrees of accuracy ϵ and confidence δ, its output on a sample produced by the environment by the manner described above for any language L in that collection, will be an ϵ-approximation of the unknown language L with confidence probability at least $1 - \delta$, no matter what the unknown distribution is, *as long as* the number of strings in the sample exceeds $p(\epsilon^{-1}, \delta^{-1}, \text{size}(L))$ for some *fixed* plynomial p. Here, grammar G is an ϵ-approximation of language L, if the probability distribution over the *symmetric difference*[6] of L and $L(G)$ is at most ϵ.

2.3 Occam Algorithm

Blumer et al. [5] have shown an extremely interesting result revealing a connection between reliable data compression and polynomial learnability. *Occam's Razor* is a principle in the philosophy of science which stipulates that a shorter theory is to be preferred as long as it remains adequate. Blumer et al. define a precise version of such a notion in the present context of learning which they call Occam Algorithm, and establishes a relation between the existence of such an algorithm and polynomial learnability: If there exists a polynomial time algorithm which reliably "compresses" any sample of any language in a given collection to a provably small consistent grammar for it, then such an alogorithm polynomially learns that collection in the limit. We state this theorem in a slightly weaker form.

Definition 2.1 *Let $\mathcal{L}$ be a language collection with associated represenation $\mathcal{H}$ with size function "size". (Define a sequence of subclasses of $\mathcal{H}$ by $\mathcal{H}_n = \{G \in \mathcal{H} \mid size(G) \leq n\}$.) Then $\mathcal{A}$ is an Occam algorithm for $\mathcal{L}$ with range size $f(m,n)$ if and only if:*[7]

$\forall\, L \in \mathcal{L}$
  $\forall S \subset graph(L)$
    if $size(L) = n$ *and* $\mid S \mid = m$ *then*
    $\mathcal{A}(S)$ *is consistent with* S
    and $\mathcal{A}(S)) \in \mathcal{H}_{f(n,m)}$
    and $\mathcal{A}$ *runs in time polynomial in the length of* S.

Theorem 2.1 (Blumer et al.) *If $\mathcal{A}$ is an Occam algorithm for $\mathcal{L}$ with range size $f(n,m) = O(n^k m^\alpha)$ for some $k \geq 1$ $0 \leq \alpha < 1$ then $\mathcal{A}$ polynomially learns $\mathcal{L}$ in the limit.*

We give below an intuitive explication of why an Occam Algorithm polynomially learns in the limit. Suppose $\mathcal{A}$ is an Occam Algorithm for $\mathcal{L}$, and let $L \in \mathcal{L}$ be the language to be learned, and n its size. Then for an arbitrary sample for L of an arbitrary size, a minimal consistent language for it will never have size larger than $size(L)$ itself. Hence $\mathcal{A}$'s output on a sample of size m will always be one of the hypotheses in $\mathcal{H}_{f(m,n)}$, whose cardinality is at most $2^{f(m,n)}$. As the sample size m grows, its effect on the probability that any consistent hypothesis in $\mathcal{H}_{f(m,n)}$ is accurate will (polynomially) soon dominate that of the growth of the cardinality of the hypothesis class, which is less than linear in the sample size.

[3] Potentially any Turning program could be a hypothesis

[4] By a *reasonble* encoding, we mean one which can represent n different grammars using $O(\log n)$ bits.

[5] In the sequel, we shall call them 'labeled samples'

[6] The symmetric difference between two sets A and B is $(A-B) \cup (B-A)$.

[7] For any langugage L, $graph(L) = \{\langle x, 0\rangle \mid x \in L\} \cup \{\langle x, 1\rangle \mid x \notin L\}$.

3 Ranked Node Rewriting Grammars

In this section, we define the class of mildly context sensitive grammars under consideration, or Ranked Node Rewriting Grammars (RNRG's). RNRG's are based on the underlying ideas of Tree Adjoining Grammars (TAG's) [8], and are also a special case of context free tree grammars [15] in which unrestricted use of variables for moving, copying and deleting, is not permitted. In other words each rewriting in this system replaces a "ranked" nonterminal node of say rank j with an "incomplete" tree containing exactly j edges that have no descendants. If we define a hierarchy of languages generated by subclasses of RNRG's having nodes and rules with bounded rank j (RNRL_j), then $\mathrm{RNRL}_0 = \mathrm{CFL}$, and $\mathrm{RNRL}_1 = \mathrm{TAL}$.[9] We formally define these grammars below.

Definition 3.1 (Preliminaries) *The following definitions are necessary for the sequel.*
(i) The set of labeled directed trees over an alphabet Σ is denoted T_Σ.
(ii) The rank of an "incomplete" tree is the number of outgoing edges with no descendents.
(iii) The rank of a node is the number of outgoing edges.
(iv) The rank of a symbol is defined if the rank of any node labeled by it is always the same, and equals that rank.
(v) A ranked alphabet is one in which every symbol has a rank.
(vi) We write $rank(x)$ for the rank of anything x, if it is defined.

Definition 3.2 (Ranked Node Rewriting Grammars) *A ranked node rewriting grammar G is a quintuple $\langle \Sigma_N, \Sigma_T, \sharp, I_G, R_G \rangle$ where:*
(i) Σ_N is a ranked nonterminal alphabet.
(ii) Σ_T is a terminal alphabet disjoint from Σ_N. We let $\Sigma = \Sigma_N \cup \Sigma_T$.
(iii) $\sharp$ is a distinguished symbol distinct from any member of Σ, indicating "an outgoing edge with no descendent".[10]
(iv) I_G is a finite set of labeled trees over Σ. We refer to I_G as the "initial trees" of the grammar.
(v) R_G is a finite set of rewriting rules: $R_G \subset \{\langle A, \alpha \rangle \mid A \in \Sigma_N \ \& \ \alpha \in T_{\Sigma \cup \{\sharp\}} \ \& \ rank(A) = rank(\alpha)\}$. (In the sequel, we write $A \rightarrow \alpha$ for rewriting rule $\langle A, \alpha \rangle$.)
(vi) $rank(G) = max\ \{rank(A) \mid A \in \Sigma_N\}$.

We emphasize that the nonterminal vs. terminal distinction above does not coincide with the internal node vs. frontier node distinction. (See examples 2.1 - 2.3.) Having defined the notions of 'rewriting' and 'derivation' in the obvious manner, the tree language of a grammar is then defined as the set of trees *over the terminal alphabet*, which can be derived from the grammar.[11] This is analogous to the way the string language of a rewriting grammar in the Chomsky hierarchy is defined.

Definition 3.3 (Tree Languages and String Languages) *The tree language and string language of a RNRG G, denoted $T(G)$ and $L(G)$ repectively, are defined as follows;*
$T(G) = \{\beta \in T_{\Sigma_T} \mid \exists \alpha \in I_G \text{ such that } \alpha \vdash_G^* \beta\}$
$L(G) = \{yield(\beta) \mid \beta \in T(G)\}$.

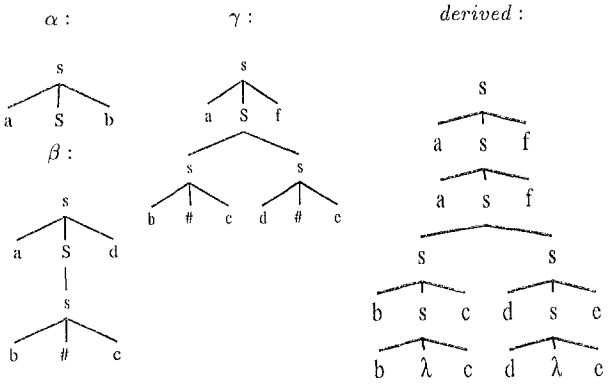

Figure 2: α, β, γ and deriving '$aabbccddeeff$' by G_3

If we now define a hierarchy of languages generated by subclasses of RNRG's with bounded ranks, context free languages (CFL) and tree adjoining languages (TAL) constitute the first two members of the hierarchy.

Definition 3.4 *For each $j \in N$ $RNRG_j = \{G \mid G \in RNRG \ \& \ rank(G) \leq j\}$. For each $j \in N$, $RNRL_j = \{L(G) \mid G \in RNRG_j\}$*

Theorem 3.1 *$RNRL_0 = CFL$ and $RNRL_1 = TAL$.*

We now give some examples of grammars in this hierarchy,[12] which also illustrate the way in which the weak generative capacity of different levels of this hierarchy increases progressively.[13]

Example 3.1 *$L_1 = \{a^n b^n \mid n \in N\} \in CFL$ is generated by the following $RNRG_0$ grammar, where α is shown in Figure 2.*
$G_1 = \langle \{S\}, \{s, a, b\}, \sharp, \{S\}, \{S \rightarrow \alpha, S \rightarrow s(\lambda)\} \rangle$

Example 3.2 *$L_2 = \{a^n b^n c^n d^n \mid n \in N\} \in TAL$ is generated by the following $RNRG_1$ grammar, where β is shown in Figure 2.*
$G_2 = \langle \{S\}, \{s, a, b, c, d\}, \sharp, \{(S(\lambda))\}, \{S \rightarrow \beta, S \rightarrow s(\sharp)\} \rangle$

Example 3.3 *$L_3 = \{a^n b^n c^n d^n e^n f^n \mid n \in N\} \notin TAL$ is generated by the following $RNRG_2$ grammar, where γ is shown in Figure 2.*
$G_3 = \langle \{S\}, \{s, a, b, c, d, e, f\}, \sharp, \{(S(\lambda, \lambda))\}, \{S \rightarrow \gamma, S \rightarrow s(\sharp, \sharp)\} \rangle$

4 K-Local Grammars

The notion of 'locality' of a grammar we define in this paper is a measure of how much global dependency there is within the grammar. By global dependency within a grammar, we mean the interactions that exist between different rules and nonterminals in the grammar. As it is intuitively clear, allowing unbounded amont of global interaction is a major, though not only, cause of a combinatorial explosion in a search for a right grammar. K-locality limits the amount of such interaction, by

[8] Tree adjoining grammars were introduced as a formalism for linguistic description by Joshi et al. [10], [9]. Various formal and computational properties of TAG's were studied in [17]. Its linguistic relevance was demonstrated in [12].

[9] This hierarchy is different from the hierarchy of "meta-TAL's" invented and studied extensively by Weir in [20].

[10] In context free tree grammars in [15], variables are used in place of $\sharp$. These variables can then be used in rewriting rules to move, copy, or erase subtrees. It is this restriction of avoiding such use of variables that keeps RNRG's within the class of efficiently recognizable rewriting systems called "Linear context free rewriting systems" ([18]).

[11] This is how an "obligatory adjunction constraint" in the tree adjoining grmar formalism can be simulated.

[12] Simpler trees are represented as term structures, whereas more involved trees are shown in the figure. Also note that we use uppercase letters for nonterminals and lowercase for terminals.

[13] Some linguistic motivations of this extension of TAG's are argugued for by the author in [1].

bounding the number of different rules that can participate in any single derivation.

Formally, the notion of "k-locality" of a grammar is defined with respect to a formulation of derivations due originally to Vijay-Shankar, Weir, and Joshi ([19]), which is a generalization of the notion of parse trees for CFG's. In their formulation, a derivation is a tree recording the history of rewritings. The root of a derivation tree is labeled with an initial tree, and the rest of the nodes with rewriting rules. Each edge corresponds to a rewriting; the edge from a rule (host rule) to another rule (applied rule) is labeled with the address of the node in the host tree at which the rewriting takes place.

The degree of locality of a derivation is the number of distinct kinds of rewritings that appear in it. In terms of a derivation tree, the degree of locality is the number of different kinds of edges in it, where two edges are equivalent just in case the two end nodes are labeled by the same rules, and the edges themselves are labeled by the same node address.

Definition 4.1 *Let $\mathcal{D}(G)$ denote the set of all derivation trees of G, and let $\tau \in \mathcal{D}(G)$. Then, the degree of locality of τ, written locality(τ), is defined as follows. locality(τ) = $card\{\langle p, q, \eta\rangle \mid$ there is an edge in τ from a node labeled with p to another labeled with q, and is itself labeled with $\eta\}$*

The degree of locality of a grammar is the maximum of those of all its derivations.

Definition 4.2 *A RNRG G is called k-local if $max\{locality(\tau) \mid \tau \in \mathcal{D}(G)\} \leq k$.*
We write k-Local-RNRG = $\{G \mid G \in$ RNRG and G is k-Local$\}$ and k-Local-RNRL = $\{L(G) \mid G \in$ k-Local-RNRG $\}$, etc..

Example 4.1 *$L_1 = \{a^n b^n a^m b^m \mid n, m \in N\} \in 4$-Local-RNRL$_0$ since all the derivations of $G_1 = (\{S\}, \{s, a, b\}, \sharp, \{s(S, S)\}, \{S \rightarrow s(a, S, b), S \rightarrow \lambda\})$ generating L_1 have degree of locality at most 4. For example, the derivation for the string a^3b^3ab has degree of locality 4 as shown in Figure 3.*

Because locality of a derivation is the number of distinct kinds of rewritings, *inclusive* of the positions at which they take place, k-locality also puts a bound on the number of nonterminal occurrences in any rule. In fact, had we defined the notion of k-locality by the two conditins: (i) at most k rules take part in any derivation, (ii) each rule is k-bounded.[14], the analogous learnability result would follow essentially by the same argument. So, k-locality in effect forces a grammar to be an unbounded union of boundedly simple grammar, with bounded number of rules each boundedly small, with a bounded number of nonterminals. This fact is captured formally by the existence of the following normal form with only a polynomial expansion factor.

Lemma 4.1 (K-Local Normal Form) *For every k-Local-RNRG$_j$ G, if we let $n = size(G)$, then there is a RNRG$_j$ G' such that*

1. *$L(G') = L(G)$.*
2. *G' is in k-local normal form, i.e. $G' = \cup\{H_i \mid i \in I_{G'}\}$ such that:*
   - *(a) each H_i has a nonterminal set that is: disjoint from any other H_j.*
   - *(b) each H_i is k-simple, that is*
     - *i. each H_i contains exactly 1 initial tree.*

[14] 'K-bounded' here means k nonterminal occurrences in each rule, [4]. For instance, a context free grammar in Chomsky Normal Form has only 2-bounded rules.

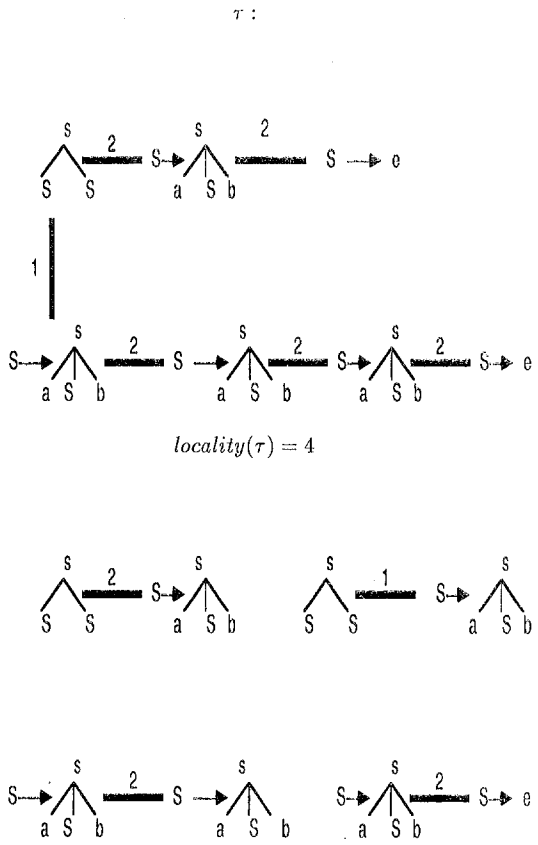

Figure 3: Degree of locality of a derivation of a^3b^3ab by G_1

ii. each H_i contains at most k rules.

iii. each H_i contains at most k nonterminal occurrences.

3. $size(G') = O(n^{k+1})$.

Crucially, the constraint of k-locality on RNRG's is an interesting one because not only each k-local subclass is an exponential class containing infinitely many infinite languages, but also k-local subclasses of the RNRG hierarchy become progressively more complex as we go higher in the hierarchy. In particular, for each j, RNRG_j can "count up to" $2(j+1)$ and for each $k \geq 2$, k-local-RNRG_j can also count up to $2(j+1)$.[15] We summarize these properties of k-local-RNRL's below.

Theorem 4.1 *For every $k \in N$,*

1. *$\forall j \in N \ \cup_{k \in N}$ k-local-$RNRL_j$ = $RNRL_j$.*
2. *$\forall j \in N \ \forall k \geq 3$ k-local-$RNRL_{j+1}$ is incomparable with $RNRL_j$.*
3. *$\forall j, k \in N$ k-local-$RNRL_j$ is a proper subset of (k+1)-local-$RNRL_j$.*
4. *$\forall j \ \forall k \geq 2 \in N$ k-local-$RNRL_j$ contains infinitely many infinite languages.*

Informal Proof:

1 is obvious because for each grammar in RNRL_j, the degree of locality of the grammar is finite.

As for *2*, we note that the sequence of the languages (for the first three of which we gave example grammars) $L_i = \{a_1^n a_2^n ... a_{2i}^n \mid n \in N\}$ are each in 3-local-RNRL_{i-1} but not in RNRL_{i-2}.

To verify *3*, we give the following sequence of languages $L_{j,k}$ such that for each j and k, $L_{j,k}$ is in k-local-RNRL_j but not in (k-1)-local-RNRL_j. Intuitively this is because k-local-languages can have at most $O(k)$ mutually independent dependencies in a single sentence.

Example 4.2 *For each $j, k \in N$, let* $L_{j,k} = \{ a_1^{n_1} ... a_{2(j+1)}^{n_1} \, a_1^{2n_2} ... a_{2(j+1)}^{2n_2} \ ... \ a_1^{kn_k} ... a_{2(j+1)}^{kn_k} \mid n_1, n_2, ..., n_k \in N\}$.

4 is obvious because $\mathcal{L}_\infty = \cup_{w \in \Sigma^*} L_w$ where $L_w = \{w^n \mid n \in N\}$ are a subset of 2-local-RNRL_0, and hence is a subset of k-local-RNRL_j for every j and $k \geq 2$. $\mathcal{L}_\infty$ clearly contains inifinitely many infinite languages. □

5 K-Local Languages Are Learnable

It turns out that *each* k-local subclass of *each* RNRL_j is polynomially learnable.

Theorem 5.1 *For each j and k, k-local-$RNRL_j$ is polynomially learnable.*

This theorem can be proved by exhibiting an Occam Algorithm (c.f. Subsection 2.3), for this class with a range size which is logarithmic in the sample size, and polynomial in the size of a minimal consistent grammar. We ommit a detailed proof and give an informal outline of the proof.

1. By the Normal Form Lemma, for any k-local-RNRG G, there is a language equivalent k-local-RNRG H in k-local normal form whose size is only polynomially larger than the size of G.
2. The number of k-simple grammars with is apriori infinite, but for a given positive sample, the number of such grammars that are 'relevant' to that sample (i.e. which could have been used to derive any of the examples) is polynomially bounded in the length of the sample. This follows essentially by the non-erasure and non-copying properties of RNRG's. (See [3] for detail.)
3. Out of the set of k-simple grammars in the normal form thus obtained, the ones that are inconsistent with the negative sample are eliminated. Such a filtering can be seen to be performable in polynomial time, appealing to the result of Vijay-Shankar, Weir and Joshi [18] that Linear Context Free Rewriting Systems (LCFRS's) are polynomial time recognizable. That RNRG's are indeed LCFRS's follow also from the non-erasure and non-copying properties.
4. What we have at this stage is a polynomially bounded set of k-simple grammars of *varying sizes* which are all consistent with the input sample. The 'relevant' part[16] of a minimal consistent grammar in k-local normal form is guaranteed to be a subset of this set of grammars. What an Occam algorithm needs to do, then, is to find some subset of this set of k-simple grammars that "covers" all the points in the positive sample, and has a total size that is provably only polynomially larger than the minimal total size of a subset that covers the positive sample and is less than linear in the sample size.
5. We formalize this as a variant of "Set Cover" problem which we call "Weighted Set Cover"(WSC), and prove (in [2]) the existence of an approximation algorithm with a performance guarantee which suffices to ensure that the output of $\mathcal{A}$ will be a basis set consistent with the sample which is provably only polynomially larger than a minimal one, and is less than linear in the sample size. The algorithm runs in time polynomial in the size of a minimal consistent grammar and the sample length.

6 Discussion: Possible Implications to the Theory of Natural Language Acquisition

We have shown that a single, nontrivial constraint of 'k-locality' allows a rich class of mildly context sensitive languages, which are argued by some [9] to be an upperbound of weak generative capacity that may be needed by a linguistic formalism, to be learnable. Let us recall that k-locality puts a bound on the amount of global interactions between different parts (rules) of a grammar. Although the most concise discription of natrual language might require almost unbounded amount of such interactions, it is conceivable that the actual grammar that is acquired by humans have a bounded degree of interactions, and thus in some cases may involve some inefficiency and redundancy. To illustrate the nature of inefficiecy introduced by 'forcing' a grammar to be k-local, consider the following. The syntactic category of a noun phrase seems to be essentially context independent in the sense that a noun phrase in a subject position and a noun phrase in an object position are more or less syntactically equivalent. Such a 'generalization' contributes to the 'global' interaction in a grammar. Thus, for a k-local grammar (for some relatively small k) to account for it, it may have to repeat the same set of noun phrase rules for different constructions.

[15] A class of grammars $\mathcal{G}$ is said to be able to "count up to" j, just in case $\{a_1^n a_2^n ... a_j^n \mid n \in N\} \in \{L(G) \mid G \in \mathcal{G}\}$ but $\{a_1^n a_2^n ... a_{j+1}^n \mid n \in N\} \notin \{L(G) \mid G \in \mathcal{G}\}$.

[16] This notion is to be made precise.

As is stated in Section 4, for each fixed k, there are clearly a lot of languages (in a given class) which could not be generated by a k-local grammar. However, it is also the case that many languages, for which the most concise grammar is not a k-local grammar, can be generated by a less concise (and thus perhaps less *explanatory*) grammar, which *is* k-local. In some sense, this is similar to the well-known distinction of 'competence' and 'performance'. It is conceivable that *performance grammars* which are actually acquired by humans are in some sense much less efficient and less explanatory than a competence grammar for the same language. After all when the 'projection problem' asks: 'How is it possible for human infants to acquire their native languages...', it does not seem necessary that it be asking the question with respect to 'competence grammars', for what we know is that the set of 'performance grammars' is feasibly learnable. The possibility that we are suggesting here is that 'k-locality' is not *visible* in competence grammars, however, it is *implicitly* there so that the *languages* generated by the class of competence grammars, which are not necessarily k-local, are indeed all k-local languages for some fixed 'k'.

7 Conclusions

We have investigated the use of complexity theory to the evaluation of grammatical systems as linguistic formalisms from the point of view of feasible learnability. In particular, we have demonstrated that a single, natural and non-trivial constraint of "locality" on the grammars allows a rich class of mildly context sensitive languages to be feasibly learnable, in a well-defined complexity theoretic sense. Our work differs from recent works on efficient learning of formal languages, for example by Angluin ([4]), in that it uses only examples and no other powerful oracles. We hope to have demonstrated that learning formal grammars need not be doomed to be necessarily computationally intractable, and the investigation of alternative formulations of this problem is a worthwhile endeavour.

8 Acknowledgment

The research reported here in was in part supported by an IBM graduate fellowship awarded to the author. The author gratefully acknowledges his advisor, Scott Weinstein, for his guidance and encouragement throughout this research. He has also benefitted from valuable discussions with Aravind Joshi and David Weir. Finally he wishes to thank Haim Levkowitz and Ethel Schuster for their kind help in formatting this paper.

References

[1] Naoki Abe. Generalization of tree adjunction as ranked node rewriting. 1987. Unpublished manuscript.

[2] Naoki Abe. Polynomial learnability and locality of formal grammars. In *26th Meeting of A.C.L.*, June 1988.

[3] Naoki Abe. Polynomially learnable subclasses of mildy context sensitive languages. 1987. Unpublished manuscript.

[4] Dana Angluin. *Learing k-bounded context-free grammars.* Technical Report YALEU/DCS/TR-557, Yale University, August 1987.

[5] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Classifying learnable geometric concepts with the vapnik-chervonenkis dimension. In *Proc. 18th ACM Symp. on Theory of Com putation*, pages 243 - 282, 1986.

[6] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. *Learnability and the Vapnik-Chervonenkis Dimension.* Technical Report UCSC CRL-87-20, University of California at Santa Cruz, Novermber 1987.

[7] Noam Chomsky. *Aspects of the Theory of Syntax.* The MIT Press, 1965.

[8] E. Mark Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.

[9] A. K. Joshi. How much context-sensitivity is necessary for characterizing structural description - tree adjoining grammars. In D. Dowty, L. Karttunen, and A. Zwicky, editors, *Natural Language Processing - Theoretical, Computational, and Psychological Perspectives*, Cambridege University Press, 1983.

[10] Aravind K. Joshi, Leon Levy, and Masako Takahashi. Tree adjunct grammars. *Journal of Computer and System Sciences*, 10:136–163, 1975.

[11] M. Kearns, M. Li, L. Pitt, and L. Valiant. On the learnability of boolean formulae. In *Proc. 19th ACM Symp. on Theory of Comoputation*, pages 285 - 295, 1987.

[12] A. Kroch and A. K. Joshi. Linguistic relevance of tree adjoining grammars. 1989. To appear in Linguistics and Philosophy.

[13] Daniel N. Osherson, Michael Stob, and Scott Weinstein. *Systems That Learn.* The MIT Press, 1986.

[14] Daniel N. Osherson and Scott Weinstein. Identification in the limit of first order structures. *Journal of Philosophical Logic*, 15:55 - 81, 1986.

[15] William C. Rounds. Context-free grammars on trees. In *ACM Symposium on Theory of Computing*, pages 143–148, 1969.

[16] Leslie G. Valiant. A theory of the learnable. *Communications of A.C.M.*, 27:1134–1142, 1984.

[17] K. Vijay-Shanker and A. K. Joshi. Some computational properties of tree adjoining grammars. In *23rd Meeting of A.C.L.*, 1985.

[18] K. Vijay-Shanker, D. J. Weir, and A. K. Joshi. Characterizing structural descriptions produced by various grammatical formalisms. In *25th Meeting of A.C.L.*, 1987.

[19] K. Vijay-Shanker, D. J. Weir, and A. K. Joshi. On the progression from context-free to tree adjoining languages. In A. Manaster-Ramer, editor, *Mathematics of Language*, John Benjamins, 1986.

[20] David J. Weir. *From Context-Free Grammars to Tree Adjoining Grammars and Beyond - A dissertation proposal.* Technical Report MS-CIS-87-42, University of Pennsylvania, 1987.