Bericht Nr. 84, Februar 1997

Arbeitspapiere des
Sonderforschungsbereichs 340

ISSN 0947-6954/97

Sprachtheoretische Grundlagen für die Computerlinguistik

James Rogers :

The Descriptive Complexity of Generalized Local Sets

Universität Stuttgart Universität Tübingen
IBM Deutschland Informationssysteme GmbH

Arbeitspapiere des SFB 340
Bericht Nr. 84, Februar 1997

THE DESCRIPTIVE COMPLEXITY OF GENERALIZED LOCAL SETS

JAMES ROGERS

ABSTRACT. Context-free grammars and tree automata, because they are required to be finite, are limited to defining sets of trees in which the branching is bounded by a finite constant. As a result they cannot capture accounts of syntactic phenomena in which no such *a priori* bound exists—in flat accounts of coordination, for instance. This mismatch led Langendoen in 1976 and Gazdar, et al., in 1985 (GPSG) to propose varieties of two level grammars, in the one case infinite grammars that are themselves generated by other grammars, in the other grammars that permit regular expressions on the right-hand side of rewrite rules. In earlier work, we have characterized the local sets (the sets of trees generated by CFGs) and the recognizable sets (those accepted by tree automata) by definability in the logical language $L^2_{K,P}$. In defining such sets of trees in $L^2_{K,P}$, however, one must explicitly bound the branching. In this paper we explore the consequences of relaxing these bounds. We show, for instance, that the GPSG account of coordination can be captured in $L^2_{K,P}$. Moreover, we show that the sets of finite trees definable in $L^2_{K,P}$ without the bound are exactly the sets of trees accepted by tree automata that are representable as regular sets, or, equivalently, projections of sets of trees generated by infinite CFGs that are themselves generated by regular grammars.

1. INTRODUCTION

Let us take, as a starting point, the following descriptive characterization of the strongly context-free sets:

Theorem 1. *A set of finite trees with bounded branching is definable in $L^2_{K,P}$ iff it is a projection of a local set.*

To appear in: Kolb, H.-P. & U. Mönnich (eds.), *The Mathematics of Syntactic Structure*.

ISSN 0947-6954/97

This is a variation of a result originally due to Doner [Don70]. It characterizes the sets of derivation trees generated by context-free grammars (the *local sets*) in terms of their definability as sets of models for a restricted logical language, in this case $L^2_{K,P}$, a monadic second-order language over trees, which is a linguistically oriented language equivalent in descriptive power to SωS, the monadic second-order theory of the complete ω-branching tree.[1] The content of the theorem is that if one restricts to finite trees in which there is a fixed finite bound on the branching (both definable restrictions in $L^2_{K,P}$), then the sets of such trees that satisfy collections of constraints stateable in $L^2_{K,P}$ are strongly context-free in the sense that it is possible to generate them with a context-free grammar, possibly by adding auxiliary labels that refine the categories labeling the trees into finitely many sub-categories. All of this will be made more precise shortly.

The qualifications to this theorem—the restriction to trees with bounded branching and the projection—have both formal and linguistic significance.[2] The effect of the projection is to permit the refinement of categories on the basis of local context that is typical of GPSG and HPSG (as in *slashed categories*). Exploring this in [Rog96c] we establish some formal consequences of the question of whether to require all categories to be empirically justified or whether to admit their arbitrary refinement in this way. In this paper we turn our attention to the requirement that there be a finite bound on the branching factor of the trees. Such a bound exists for the local sets simply because we require CFGs to be finite—since there can be only finitely many rewriting rules and since they are restricted to expanding non-terminals into finite strings of terminals and non-terminals such a bound must always exist. Furthermore, it has be argued (on learnability grounds, for instance, see [Kay84]) that such a bound should exist for human languages. Nonetheless, tree sets in which no finite bound on the branching exists are linguistically interesting figuring, for instance, in all *flat* accounts of coordination.[3] Such accounts of coordination also figure prominently in GPSG and HPSG. Strikingly, when we relax this restriction and allow trees with arbitrary finite branching, the sets of trees

[1] Doner's result was in terms of wSnS, a definable sub-theory of SωS.

[2] The requirement that the trees be finite is a simple consequence of our assumption that they can be generated by a finite process. This restriction is without great significance.

[3] Note that finiteness of the trees will still require all branching to be finite, branching of any finite degree, however, may be admitted.

definable in $L^2_{K,P}$ (or SωS) turn out to be a mild generalization of the sets of trees definable using GPSG-style grammars.

In the next two sections we introduce $L^2_{K,P}$ and present our formalizations of CFGs, tree automata, and the sets of trees they generate. We discuss the descriptive characterizations of these classes of sets of trees in Section 4. In Section 5 we look at the linguistic motivation for accounts of syntax without fixed bounds on branching. We then introduce generalizations of CFGs and tree automata that can accommodate sets of trees without such bounds and characterize the sets of trees definable in $L^2_{K,P}$ without explicitly bounded branching in those terms. We close, in Section 7, with some thoughts on the significance of the model-theoretic approach to syntax these results exemplify.

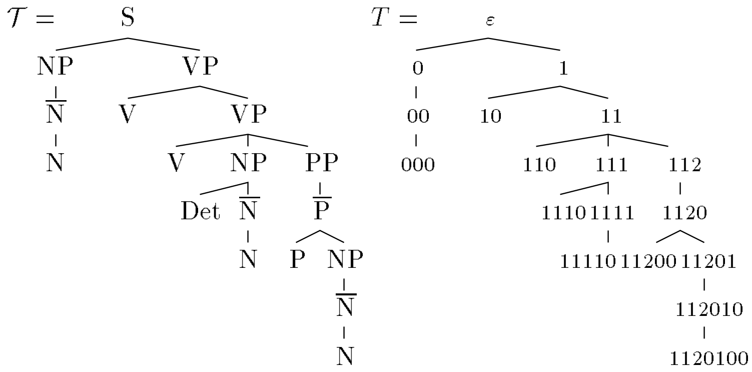
2. $L^2_{K,P}$

$L^2_{K,P}$ is a mathematical language for specifying properties of trees which can naturally express the types of constraints that typically comprise theories of natural language syntax. It includes the binary predicates:

$\triangleleft$: — immediate domination (constituency),
$\triangleleft^*$: — (reflexive) domination,
$\triangleleft^+$: — proper domination,
$\prec$: — linear precedence (left-of), and
$\approx$: — equality,

as well as a collection of labels $\boldsymbol{\mathcal{P}}$ (constants ranging over sets of nodes), a collection of individual constants $\boldsymbol{\mathcal{K}}$, and two sorts of variables—those ranging over individuals (for which we will use lower case), and those ranging over arbitrary sets (for which we will use upper case). Either sort of variable may be bound by quantifiers. We denote the fact that an individual a is a member of the set labeled A with the formula $A(a)$. Similarly, the formula $X(x)$ will be made true iff the individual assigned to the variable x is a member of the set assigned to the variable X.

We interpret these formulae over trees in the form of ordinary model-theoretic structures $\mathcal{T} = \langle T, P_T, D_T, L_T, I_{\mathcal{K}}, I_{\mathcal{P}} \rangle$. The domain T is a *tree domain* [Gor67]—a set of strings of natural numbers which are interpreted as node addresses: the root of the tree occurs at address ε and the children of a node at address w are at addresses $w0, w1, \ldots, wn$ in left-to right order. Tree domains are required to be both prefix closed ($uv \in T \Rightarrow u \in T$) and left closed ($ui \in T$ and $j < i \Rightarrow uj \in T$). The predicates $\triangleleft$, $\triangleleft^*$, and $\prec$ are interpreted by P_T, D_T, and L_T, respectively, which are fixed



$$
\begin{aligned}
\boldsymbol{K} &= \emptyset,\ \boldsymbol{P} = \{\mathrm{S}, \mathrm{NP}, \mathrm{VP}, \mathrm{PP}, \overline{\mathrm{N}}, \overline{\mathrm{P}}, \mathrm{N}, \mathrm{V}, \mathrm{Det}, \mathrm{P}\} \\
P_T &= \{\langle \varepsilon, 0\rangle, \langle \varepsilon, 1\rangle, \langle 0, 00\rangle, \langle 00, 000\rangle, \langle 1, 10\rangle, \ldots, \langle 112010, 1120100\rangle\} \\
D_T &= \{\langle \varepsilon, \varepsilon\rangle, \langle \varepsilon, 0\rangle, \langle \varepsilon, 00\rangle, \langle \varepsilon, 000\rangle, \langle \varepsilon, 1\rangle, \ldots, \langle 112010, 1120100\rangle, \\
&\qquad \langle 1120100, 1120100\rangle\} \\
L_T &= \{\langle 0, 1\rangle, \langle 0, 10\rangle, \langle 0, 11\rangle, \ldots, \langle 00, 1\rangle, \langle 00, 10\rangle, \langle 00, 11\rangle, \langle 00, 110\rangle, \\
&\qquad \ldots, \langle 11200, 1120100\rangle\} \\
I_{\mathcal{K}} &= \emptyset \\
I_{\mathcal{P}} &= \{\mathrm{S} \mapsto \{\varepsilon\}, \mathrm{NP} \mapsto \{0, 111, 11201\}, \mathrm{VP} \mapsto \{1, 11\}, \ldots, \mathrm{P} \mapsto \{11200\}\}
\end{aligned}
$$

FIGURE 1. A tree, e.g., *We should give the award to Alice*

by T:

$$
\begin{aligned}
P_T &\stackrel{\text{def}}{=} \{\langle u, ui\rangle \in T \times T \mid u \in \mathbb{N}^*, i \in \mathbb{N}\} \\
D_T &\stackrel{\text{def}}{=} \{\langle u, uv\rangle \in T \times T \mid u, v \in \mathbb{N}^*\} \\
L_T &\stackrel{\text{def}}{=} \{\langle uiv, ujw\rangle \in T \times T \mid u, v, w \in \mathbb{N}^*, i < j \in \mathbb{N}\}.
\end{aligned}
$$

Finally, the individual and set constants are interpreted by $I_{\mathcal{K}}$ and $I_{\mathcal{P}}$ which map $\boldsymbol{\mathcal{K}}$ to individuals in T and $\boldsymbol{\mathcal{P}}$ to subsets of T, respectively.

A formula $\varphi(\vec{x}, \vec{X})$ of $L^2_{K,P}$ with free variables in the sequences $\vec{x} = \langle x_1, \ldots, x_m\rangle$ and $\vec{X} = \langle X_1, \ldots, X_n\rangle$ is *satisfied* in a tree $\mathcal{T} = \langle T, P_T, D_T, L_T, I_{\mathcal{K}}, I_{\mathcal{P}}\rangle$ relative to an assignment s mapping individual variables in $\vec{x}$ to individuals in T and labels $\vec{X}$ to subsets of T. We extend this to a function $\hat{s}$ interpreting *terms*:

$$\hat{s}(t) = \begin{cases} s(t) & \text{if } t \in \vec{x} \text{ or } t \in \vec{X} \\ I_{\mathcal{K}}(t) & \text{if } t \in \boldsymbol{\mathcal{K}} \\ I_{\mathcal{P}}(t) & \text{if } t \in \boldsymbol{\mathcal{P}}. \end{cases}$$

Then $\mathcal{T}$ satisfies $\varphi(\vec{x}, \vec{X})$ with s in the standard way:

$$\begin{aligned} \mathcal{T} \models t \triangleleft u\ [s] \quad &\overset{\text{def}}{\iff} \quad \langle \hat{s}(t), \hat{s}(u) \rangle \in \mathcal{P}_T, \\ \mathcal{T} \models U(t)\ [s] \quad &\overset{\text{def}}{\iff} \quad \hat{s}(t) \in \hat{s}(U), \\ \mathcal{T} \models (\exists X)[\varphi(X)]\ [s] \quad &\overset{\text{def}}{\iff} \quad \mathcal{T} \models \varphi(X)\ [s[X \mapsto A]] \text{ for some } A \subseteq T, \\ \text{etc.} \end{aligned}$$

where $s[X \mapsto A]$ is the assignment s modified to map X to A. Proper domination ($\triangleleft^+$) is treated as a defined predicate—dominates and not equal to.

In this setting the individual constants $\boldsymbol{\mathcal{K}}$ are, in fact, redundant since we can take them to denote singleton sets. Thus, since the interpretations of the binary predicates are fixed in the domain, models depend only on the domain and the interpretation of $\boldsymbol{\mathcal{P}}$. If we invert $I_{\mathcal{P}}$ to get a mapping of nodes in T to labels we get the standard notion of a $\boldsymbol{\mathcal{P}}$*-labeled tree*: $\langle T, \tau \rangle$ where $\tau(w) = A \overset{\text{def}}{\iff} w \in I_{\mathcal{P}}(A)$.

3. Local and Recognizable Sets

The defining characteristic of the "context-freeness" of a syntactic theory is that it is (or can be) stated in terms of *local* relationships—relationships that hold between a category and its immediate constituents (i.e., between categories labeling a node and its children) or between phrase mates (categories labeling siblings). Such relationships can be expressed as local trees.

Definition 1 (Local trees)**.** *The* local tree *occurring at a node w in a* $\boldsymbol{P}$*-labeled tree $\mathcal{T} = \langle T, \tau \rangle$ is the pair*

$$L(\mathcal{T}, w) \overset{\text{def}}{=} \langle \tau(w), \tau(\langle wi \mid i \in \mathbb{N}, wi \in T \rangle) \rangle .$$

That is, the pair consisting of the label of that node and the sequence of labels of its children—here we extend τ to sequences of nodes in the usual way by taking $\tau(\langle w_0, \ldots, w_n \rangle)$ to be $\langle \tau(w_0), \ldots, \tau(w_n) \rangle$.

The set of local trees occurring in $\mathcal{T}$ is

$$L(\mathcal{T}) \overset{\text{def}}{=} \{L(\mathcal{T}, w) \mid w \in T\} .$$

There are specific sequences of labels occurring in a tree and, in particular, sets of such sequences (i.e., languages) that have distinguished roles in the theory.

Definition 2. *Suppose $\mathcal{T} = \langle T, \tau \rangle$ is a $\boldsymbol{P}$-labeled tree. Then:*

- *The set of* **leaves** *of $\mathcal{T}$ is the set of its atomic categories—those labeling nodes with no children:*
$$l(\mathcal{T}) \stackrel{\text{def}}{=} \{\sigma \in \boldsymbol{P} \mid \langle \sigma, \epsilon \rangle \in L(\mathcal{T})\}$$
- *The* **yield** *of $\mathcal{T}$ is the sequence of leaves of $\mathcal{T}$ in left-to-right order:*
$$\mathrm{Yd}(\mathcal{T}) \stackrel{\text{def}}{=} \tau(\langle w_i \mid w_i \in l(\mathcal{T}), i < j \Rightarrow w_i \prec w_j \rangle).$$
- *For all $\sigma \in \boldsymbol{P}$, the* **child language** *of σ in $\mathcal{T}$ is the set of strings labeling the children of nodes labeled σ:*
$$\mathrm{Ch}_\sigma(\mathcal{T}) \stackrel{\text{def}}{=} \{\alpha \in \boldsymbol{P}^* \mid \langle \sigma, \alpha \rangle \in L(\mathcal{T})\}$$

Again as usual, we extend these to a set of trees $\mathbb{T}$ by taking the union over each tree in the set, e.g.,
$$L(\mathbb{T}) \stackrel{\text{def}}{=} \bigcup_{\mathcal{T} \in \mathbb{T}} [L(\mathcal{T})].$$
In particular, the yield of a set of trees $\mathbb{T}$ is the string language associated with that set
$$\mathrm{Yd}(\mathbb{T}) \stackrel{\text{def}}{=} \{\mathrm{Yd}(\mathcal{T}) \mid \mathcal{T} \in \mathbb{T}\}.$$

3.1. **Local Sets.** While Context-free grammars are familiar objects, we present them in a somewhat unusual manner, one that is chosen to emphasize the relationships between CFGs and the local sets and between CFGs and tree automata.

Definition 3 (Context-free grammars)**.** *A (positive)* context-free grammar *(CFG) is a finite set of local trees over some alphabet Σ:*
$$G \subseteq \Sigma \times \Sigma^*, \text{ finite }.$$

For any $\Sigma_S \subseteq \Sigma$ the set of trees admitted *by G wrt Σ_S is the set of finite trees constructed from the local trees in G in which the root is labeled with some symbol in Σ_S:*
$$G(\Sigma_S) \stackrel{\text{def}}{=} \{\mathcal{T} = \langle T, \tau \rangle \mid T \text{ finite }, \ \tau(\epsilon) \in \Sigma_S, \ \text{and } L(\mathcal{T}) \subseteq G\}$$

The set of terminal symbols of G is:
$$\Lambda_G \stackrel{\text{def}}{=} \{\sigma \in \Sigma \mid \langle \sigma, \epsilon \rangle \in G\}.$$

This definition generalizes the standard notion of CFG in two ways: we allow a finite set of start symbols (rather than just one), and we allow productions for terminal symbols. Thus terminal symbols may label interior nodes of the tree, although, since leaves are local trees with empty sequences of children and must be licensed by G, only terminal symbols may label the leaves (i.e., $l(\mathcal{T}) \subseteq \Lambda(G)$). These generalizations are inconsequential for string languages, of course, but are significant when considering the set of trees admitted by a grammar, where the standard notion is unnaturally restrictive. Note that all grammars in this form are *positive*—they do not admit languages including the empty string. Since local trees with empty sequences of children are taken to be terminals, there is no way of erasing a non-terminal.

It should be clear that a tree is admitted by a CFG iff it is a derivation tree generated by that CFG in its more typical presentation.

Definition 4 (Local sets). *A set of trees is* local *iff it is the set of trees admitted by some CFG.*

The *context-free (string) languages* are exactly the yields of the local sets.

3.2. **Recognizable Sets.** As we have noted, constraints stated in $L^2_{K,P}$ are capable of defining sets of trees that are not local sets. We will show in the next section that the definable sets are, in fact *recognizable sets*. These are the sets of trees that are recognized by *tree automata* [GS84], which we can regard as CFGs with a set of auxiliary labels (states) that are employed in controlling the derivation but do not show up in the derived tree.

Definition 5 (Tree Automata). *A* tree automaton *over an alphabet* Σ *and a finite set of states* Q *is a finite set of triples:*

$$\mathcal{A} \subseteq \Sigma \times Q \times Q^*,\ \textit{finite}\ .$$

For any $Q_S \subseteq Q$ *the set of trees* recognized *by* $\mathcal{A}$ *(wrt* Q_S*) is*

$$\begin{aligned}
\mathcal{A}(Q_S) &\stackrel{\mathrm{def}}{=} \\
&\{\mathcal{T} = \langle T, \tau\rangle \mid T \textit{ finite and} \\
&\qquad \exists r : T \to Q \textit{ such that} \\
&\qquad\quad r(\epsilon) \in Q_S \textit{ and} \\
&\qquad\quad (\forall w \in T)[\langle \tau(w), r(w), r(\langle wi \in T\rangle)\rangle \in \mathcal{A}]\}.
\end{aligned}$$

Which is to say, $\langle T, \tau\rangle$ is accepted by $\mathcal{A}$ wrt Q_S iff it is possible to assign states in Q to the nodes in T, with the root being assigned a state

$$\begin{aligned}\mathcal{A}_{1\mathrm{B}} = \{ \; & \langle A, S_0, \langle S_0, S_1\rangle\rangle\,, \\ & \langle A, S_0, \langle S_1, S_0\rangle\rangle\,, \\ & \langle A, S_1, \langle S_1, S_1\rangle\rangle\,, \\ & \langle A, S_0, \epsilon\rangle\,, \\ & \langle A, S_1, \epsilon\rangle\,, \\ & \langle B, S_0, \langle S_1, S_1\rangle\rangle\,, \\ & \langle B, S_0, \epsilon\rangle \qquad\qquad \}\end{aligned}$$

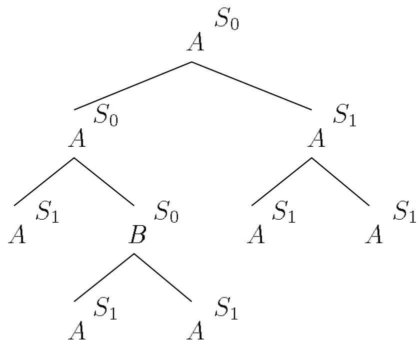


FIGURE 2. Separating the recognizable and local sets.

in Q_S, that is consistent with $\mathcal{A}$ in the sense that if a node labeled σ is assigned state q and its children are assigned the sequence of states $\langle q_0, \ldots, q_n\rangle$ (ε if there are no children) then $\langle \sigma, q, \langle q_0, \ldots, q_n\rangle\rangle$ is in $\mathcal{A}$. The assignment r of states to nodes in the domain of $\mathcal{T}$ is referred to a **run** of $\mathcal{A}$ (for Q_S) on $\mathcal{T}$.

Definition 6 (Recognizable Sets)**.** *A set of trees is* recognizable *iff it is $\mathcal{A}(Q_S)$ for some $\mathcal{A}$ and Q_S.*

Clearly every local set is recognizable—take Q to be Σ. The converse is not true, however; the inclusion is proper. For example, let $\mathcal{A}_{1B}$ be the tree automaton of Fig. 2. The set $\mathcal{A}_{1B}(\{S_0\})$ is the set of binary-branching trees in which at most one node is labeled B, all others are labeled A (a set we will refer to as "OneB"). To see that this is not a local set, note that if the local tree $\langle A, \langle A, B\rangle\rangle$, for instance, occurs in a tree admitted by a CFG then it must occur in that CFG. But then that CFG will license trees in which $\langle A, \langle A, B\rangle\rangle$ occurs any number of times.

The automaton of the example recognizes OneB by distinguishing nodes that may dominate a B from those that may not by assigning them distinct states (S_0 and S_1, respectively). If we make that state explicit by labeling the trees with pairs in $\Sigma \times Q$ then the set becomes a local set.

The CFG G_{1B} of Fig. 3 generates trees in which no more than one node is labeled $\langle B, S_0\rangle$, all nodes that properly dominate such a node are labeled $\langle A, S_0\rangle$, and those that don't are labeled $\langle A, S_1\rangle$. In effect we are making OneB local by refining the original categories ($\{A, B\}$) into finitely many sub-categories ($\{\langle A, S_0\rangle, \langle A, S_1\rangle, \langle B, S_0\rangle, \langle B, S_1\rangle\}$) which suffice to distinguish those nodes that may dominate Bs from those that

$$
\begin{array}{l}
G_{1B} = \\
\quad \{\; \langle A, S_0 \rangle \to \langle \langle A, S_0 \rangle, \langle A, S_1 \rangle \rangle, \\
\qquad \langle A, S_0 \rangle \to \langle \langle A, S_1 \rangle, \langle A, S_0 \rangle \rangle, \\
\qquad \langle A, S_0 \rangle \to \langle \langle B, S_0 \rangle, \langle A, S_1 \rangle \rangle, \\
\qquad \langle A, S_0 \rangle \to \langle \langle A, S_1 \rangle, \langle B, S_0 \rangle \rangle, \\
\qquad \langle A, S_0 \rangle \to \quad \varepsilon, \\
\qquad \langle A, S_1 \rangle \to \langle \langle A, S_1 \rangle, \langle A, S_1 \rangle \rangle, \\
\qquad \langle A, S_1 \rangle \to \quad \varepsilon, \\
\qquad \langle B, S_0 \rangle \to \langle \langle A, S_1 \rangle, \langle A, S_1 \rangle \rangle, \\
\qquad \langle B, S_0 \rangle \to \quad \varepsilon \qquad \}
\end{array}
$$

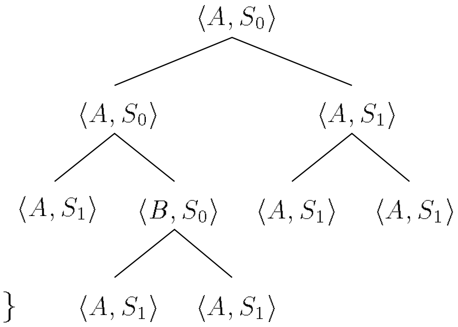


FIGURE 3. Making the state explicit.

may not. Note that the original trees are recoverable from the "refined" trees by taking the first projection of their labels. This approach works in general—whenever it is possible to refine the categories labeling a set of trees into finitely many sub-categories in such a way that the ways in which a given category may be expanded is determined entirely by its sub-category then the equivalent set of trees labeled with the sub-categories will be a local. This is always the case for recognizable sets.

Definition 7. *Suppose $\mathcal{T} = \langle T, \tau \rangle$ is a tree and the range of τ is a set of n-tuples. The i^{th}* -projection *of $\mathcal{T}$, for $0 < i \leq n$, is $\langle T, \pi_i \circ \tau \rangle$—$T$ with nodes labeled with the i^{th} component of their labels in $\mathcal{T}$.*

Lemma 1 (Thatcher [Tha67])**.** *Every recognizable set is the projection of a local set.*

Proof. The proof simply carries out the idea of the example. Suppose $\mathbb{T} = \mathcal{A}(Q_S)$ for some tree automaton $\mathcal{A}$ over Σ and Q and set of initial states Q_S. Let $\hat{\mathcal{A}}(Q_S)$ be the set of $\Sigma \times Q$-labeled trees:

$$\hat{\mathcal{A}}(Q_S) = \{\langle T, \tau \rangle \mid \mathcal{T} = \langle T, \pi_1 \circ \tau \rangle \in \mathcal{A}(Q_S) \text{ and } \pi_2 \circ \tau \text{ is a run of } \mathcal{A} \text{ on } \mathcal{T}\}.$$

Which is to say that $\langle T, \tau \rangle \in \hat{\mathcal{A}}(Q_S)$ iff the first projection of $\mathcal{T}$ is a tree in $\mathcal{A}(Q_S)$ and the second projection of τ encodes a run of $\mathcal{A}(Q_S)$ on that tree.

Thus $\mathcal{A}(Q_S)$ is a projection of $\hat{\mathcal{A}}(Q_S)$. It is not hard to show that $\hat{\mathcal{A}}(Q_S)$ is generated by a CFG and is, therefore, a local set. □

Corollary 1. *A language is a CFL iff it is the yield of a recognizable set.*

This follows from the fact that we might as well label the leaves in $\hat{\mathcal{A}}(Q_S)$ with Σ.

4. Descriptive Characterization

We are now in a position to establish the characterization we took as our starting point, which is, in essence, a characterization of the recognizable sets by definability in $L^2_{K,P}$. The result was originally established for SnS, the monadic second-order theory of n successor functions [Rab69]. As we will show shortly, $L^2_{K,P}$ and SnS are equivalent in their descriptive power; we work in $L^2_{K,P}$ because it has a relatively natural signature for reasoning about trees in the manner typical of theories of syntax. Thus, while encodings of linguistically significant relationships in SnS tend to be rather difficult to decipher, in $L^2_{K,P}$ they are reasonably transparent.

Definition 8 (SnS). *Let* $\mathsf{n}_n = \langle T_n, \varepsilon, \triangleleft^*, \preceq\!\!\triangleleft, r_i \rangle_{i<n}$, *where* T_n *is the complete* n*-branching tree-domain,* ε *is the root of* T_n, $\triangleleft^*$ *and* $\preceq\!\!\triangleleft$ *are domination and lexicographic order on* T_n, *respectively, and* r_i *takes elements of* T_n *to their* i^{th} *child.*

SnS *is the monadic second-order theory of* n_n*;* wSnS *is the* weak *monadic second-order theory of* n_n*, the theory in which second-order quantification is restricted to finite sets.*

Finiteness is a definable property of subsets in SnS, therefore wSnS is a definable fragment of SnS.

Theorem 2 (Büchi [Büc60]). *S1S is decidable.*

Theorem 3 (Rabin [Rab69]). *SnS is decidable for all* $n \leq \omega$.

Büchi's result is a special case of Rabin's. We will sketch Rabin's proof. Rabin first establishes this result for the $n = 2$ case. The proof is based on the idea that assignments making some formula $\varphi(\vec{X})$ over the signature of S2S true in n_2 are, in effect, $\boldsymbol{\mathcal{P}}(\vec{X})$-labeled trees with domain T_2 (where $\boldsymbol{\mathcal{P}}(\vec{X})$ is the powerset of the set of elements of $\vec{X}$). He gives an effective procedure for constructing, for any such $\varphi(\vec{X})$, a *Rabin tree-automaton*—an automaton over infinite trees which is, in essence, a tree automaton with an acceptance condition based on the sets of states that occur infinitely often along the paths of the tree—which accepts a $\boldsymbol{\mathcal{P}}(\vec{X})$-labeled tree iff it encodes a satisfying assignment for $\varphi(\vec{X})$. It follows that $(\exists \vec{X})[\varphi(\vec{X})] \in$ S2S iff the set of trees accepted by this automaton is non-empty. He then shows that the class of sets of labeled trees accepted by these automata (the class of *Rabin recognizable sets*) is closed under

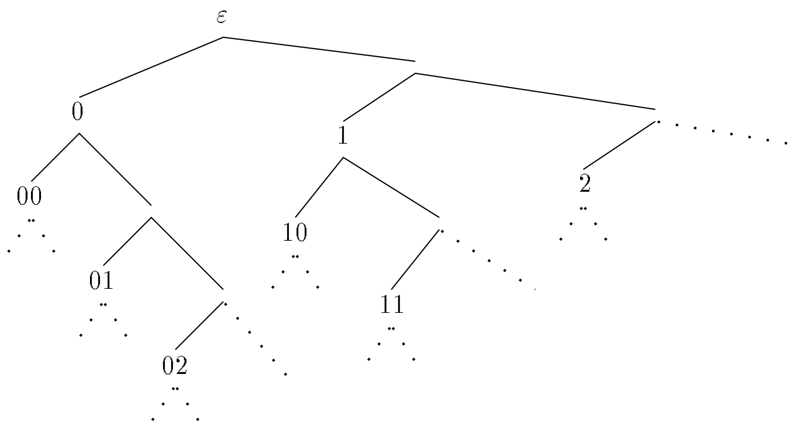


FIGURE 4. Embedding SωS in S2S.

Boolean operations and that emptiness is decidable for these sets. Thus, the result extends to arbitrary formulae over the signature of S2S and satisfiability in S2S is decidable.

Rabin generalizes this result to SnS for all $n \leq \omega$ by showing that n_n for any such n can be embedded in n_2 via the map $h : T_n \rightarrow (1^*0)^*$ where:

$$h(w) = \begin{cases} \varepsilon & \text{if } w = \varepsilon \\ h(u) \cdot 1^i \cdot 0 & \text{if } w = ui. \end{cases}$$

This map locates the i^{th} child of w at the left child of the i^{th} node on the right-most branch of the subtree rooted at w. This is pretty much the standard way of interpreting an arbitrarily branching tree as a binary branching tree. The contribution of the proof is the observation that the range of h and the relations and functions of n_n in terms of the range of h are all definable on n_2.

Definition 9. *A set $\mathbb{T}$ of $\boldsymbol{P}$-labeled trees, where $\boldsymbol{P} = \{\sigma_1, \ldots, \sigma_m\}$, is* **definable in** S$n$S *iff there is a formula $\varphi(X_U, X_1, \ldots, X_m)$ such that*

$$\mathsf{n}_n \models \varphi[X_U \mapsto T, X_i \mapsto \tau^{-1}(\sigma_i)] \Leftrightarrow \langle T, \tau \rangle \in \mathbb{T}.$$

In words, the trees in the set encode all and only the satisfying assignments for φ.

Theorem 4. *A set of trees is definable in $L^2_{K,P}$ iff it is definable in* SωS.

Proof (Sketch). The crux of the proof[4] is to show that immediate domination ($\triangleleft$) and linear precedence ($\prec$) are definable in SωS and that root (ε), lexicographic order ($\trianglelefteq$), and the successor functions (r_i) are definable in $L^2_{K,P}$. This is not hard. Let $\mathcal{T} = \langle T, \tau \rangle$ and $s = x \mapsto v, y \mapsto w$ for some $v, w \in T$. Then, first we have:

$$\begin{aligned} \mathcal{T} \models x \triangleleft y\,[s] \quad &\Leftrightarrow \quad \mathsf{n}_\omega \models x \triangleleft^* y \wedge (\forall z)[(z \triangleleft^* y \wedge \neg y \triangleleft^* z) \rightarrow z \triangleleft^* x]\,[s] \\ \mathcal{T} \models x \prec y\,[s] \quad &\Leftrightarrow \quad \mathsf{n}_\omega \models x \trianglelefteq y \wedge \neg x \triangleleft^* y\,[s]. \end{aligned}$$

Which simply says that (the point assigned to) x is the parent of (the point assigned to) y iff it dominates y and all points that properly dominate y (i.e., that dominate y but are not themselves dominated by y) dominate x. For linear precedence, it says that x is left of y iff it precedes y in lexicographic order but does not dominate it. For the other direction we have:

$$\begin{aligned} \mathsf{n}_\omega \models \varepsilon \approx x\,[s] \quad &\Leftrightarrow \quad \mathcal{T} \models (\forall z)[x \triangleleft^* z]\,[s] \\ \mathsf{n}_\omega \models x \trianglelefteq y\,[s] \quad &\Leftrightarrow \quad \mathcal{T} \models x \triangleleft^* y \vee x \prec y\,[s] \\ \mathsf{n}_\omega \models r_0(x) \approx y \quad &\Leftrightarrow \quad \mathcal{T} \models x \triangleleft y \wedge (\forall z)[x \triangleleft z \rightarrow z \not\prec y]\,[s] \\ \mathsf{n}_\omega \models r_{i+1}(x) \approx y \quad &\Leftrightarrow \quad \mathcal{T} \models x \triangleleft y \wedge (\forall z)[r_i(x) \prec z \rightarrow (y \triangleleft^* z \vee y \prec z)]\,[s] \end{aligned}$$

In words, x is the root iff it dominates all points in T, it lexicographically precedes (or is equal to) y iff it dominates y or is left-of y, and y is the 0^{th} successor of x iff it is a child of x and no child of x is left-of y. The $i+1^{\text{st}}$ successor function is defined in terms of the i^{th} : y is the $i+1^{\text{st}}$ successor of x iff it is a child of x and every point right-of the i^{th} successor of x is either dominated by y or to its right. This, of course, introduces r_i, which then must be expanded in the same way. Since $i \in \mathbb{N}$ the translation of any SωS formula will terminate after finitely many such steps. □

Note that this theorem is not true for any $n < \omega$, even if we restrict ourselves to finite trees. For any finite n it is simple to construct an $L^2_{K,P}$ formula that requires, for instance, every point with any children at all to have at least $n+1$ children. While there are infinitely many trees satisfying this formula, there can only be one satisfying assignment in n_n—the trivial tree (containing just the root)—for any translation of it into the language of SnS.

Lemma 2. *Every set of finite trees definable in* SnS *for some* $n < \omega$ *is recognizable.*

[4] For full details see [Rog96a]

This is a consequence of the fact that, if we restrict ourselves to sets of finite trees, Rabin's tree automata reduce to ordinary tree automata. Note that this lemma holds only for finite n, which insures that the branching factor of the models is finitely bounded. It is easy to define sets of trees, even sets of finite trees, in SωS in which there is no finite bound on the branching factor; in fact, unless a formula explicitly includes such a bound, if it has models with n-ary branching it will have models with branching factors greater than n as well. Such sets are not recognizable.

Theorem 5. *A set of finite trees with bounded branching is definable in $L^2_{K,P}$ iff it is recognizable.*

Proof. Clearly, if the set is definable in $L^2_{K,P}$ and has bounded branching it is definable in SnS for some finite n and is, by the lemma, recognizable. For the other direction, we must show that every recognizable set is definable in $L^2_{K,P}$. This is straightforward. Given some automaton $\mathcal{A} = \langle \delta_1, \ldots, \delta_l \rangle \subseteq \boldsymbol{P} \times Q \times Q^*$ over the alphabet $\boldsymbol{P}$ and the set of states Q, the idea is to interpret the states in Q as set variables (which we will denote as the sequence $\vec{Q}$). Each of the triples defining the automaton can then be directly encoded into an $L^2_{K,P}$ formula that holds at a sequence of points iff those points form a local tree at which the triple is satisfied. E.g., for the triple $\delta_i = \langle P, Q_0, \langle Q_1, Q_2, \ldots, Q_m \rangle \rangle$:

$$\begin{aligned}\varphi_i(x_0, \ldots, x_m, \vec{Q}) \quad = \quad & (\forall y)[x_0 \triangleleft y \leftrightarrow (y \approx x_1 \vee y \approx x_2 \vee \cdots \vee y \approx x_m)] \wedge \\ & x_1 \prec x_2 \wedge x_2 \prec x_3 \wedge \cdots \wedge x_{m-1} \prec x_m \wedge \\ & P(x_0) \wedge Q_0(x_0) \wedge Q_1(x_1) \wedge \cdots \wedge Q_n(x_m).\end{aligned}$$

We then collect these into a universally quantified formula that requires some triple to be satisfied by every local tree:

$$\varphi_{\mathcal{A}}(\vec{Q}) \quad = \quad (\forall x_0, \ldots, x_n)[\textstyle\bigvee_i \varphi_i(\vec{x}, \vec{Q})].$$

(This formula is finite only because there is a finite bound on the branching of the local trees that satisfy the triples; otherwise we would have to quantify over all sequences of nodes, which would require unboundedly many individual variables.) This gives us a formula that is satisfied by a tree relative to an assignment of the variables in $\vec{Q}$ to subsets of the tree. Such an assignment is, in essence, a run of the automaton on the tree. All that is then required is to add a conjunct requiring the root to be in the set assigned to one of the initial states in Q_S and to existentially quantify the $\vec{Q}$ and we have a sentence that is satisfied by a tree iff it is in $\mathcal{A}(Q_S)$. $\square$

Theorem 6 (Büchi [Büc60])**.** *Every language definable in* S1S *is regular.*

Here we must modify our notion of the string language defined by a set of models. Clearly, S1S models—unary branching trees—have yield languages that are limited to words of length one (and are consequently finite). Instead, we will take the string language defined by an S1S formula to be the *path language* of its set of models, that is, the strings labeling the path from the root to the leaf of these models. Since tree automata on unary-branching trees reduce to ordinary finite-state automata on strings these languages are regular. The other direction is a special case of the previous theorem.

Theorem 7 (Doner [Don70])**.** *A language is context-free iff it is the yield of a set of finite trees definable in wSnS for some $n < \omega$.*

Since we are restricting ourselves to sets of finite trees and finiteness is definable in SnS, we need not restrict Doner's result to weak SnS. Similarly, when we are working with sets of trees with bounded branching we can assume that the bound is given explicitly and work in SωS, equivalently $L^2_{K,P}$.

Corollary 2. *A language is context-free iff it is the yield of a set of finite trees definable in SnS for some $n < \omega$; equivalently, if it is the yield of a set of finite trees with bounded branching that is definable in $L^2_{K,P}$.*

5. Unbounded Branching

As should be clear from the discussion of the previous section, there is something of a misfit between $L^2_{K,P}$ and the recognizable and local sets. While the requirement that grammars and automata be finite restricts the latter to defining sets of trees with finitely bounded branching $L^2_{K,P}$ suffers from no such restriction. The distinction would be, perhaps, a minor point if it were not for the fact that sets of trees in which there is no finite bound on the branching are characteristic of some linguistic analyses, in particular *flat* analyses of coordination. Perhaps the best known presentation of this is in GPSG [GKPS85]. The relevant component consists of three (finite) rule schemas and a linear precedence rule:

$$\begin{array}{l} \mathrm{X} \longrightarrow \mathrm{H}[\mathrm{CONJ}\ a_0], \mathrm{H}[\mathrm{CONJ}\ a_i]^+ \\ \quad \langle a_0, a_1\rangle \in \{\langle \mathit{and}, \mathrm{NIL}\rangle, \langle \mathrm{NIL}, \mathit{and}\rangle, \langle \mathit{neither}, \mathit{nor}\rangle, \langle \mathit{or}, \mathrm{NIL}\rangle, \langle \mathrm{NIL}, \mathit{or}\rangle\} \end{array}$$

$$\mathrm{X[CONJ\ NIL]} \longrightarrow \mathrm{H}$$

$$\mathrm{X[CONJ}\ a] \longrightarrow \{[\mathrm{SUBCAT}\ a]\}, \mathrm{H}$$
$$a \in \{\textit{and, both, but, neither, nor, or}\}$$

$$[\mathrm{CONJ}\ a_0] \prec [\mathrm{CONJ}\ a_1]$$
$$a_0 \in \{\textit{both, either, neither}, \mathrm{NIL}\}$$
$$a_1 \in \{\textit{and, but, nor, or}\}$$

Here X is not a variable but, rather, is a radically underspecified category (no features are specified at all). The effect of the first schema (the *iterating coordination schema*) is to allow any category to be expanded to a sequence of two or more categories, all of which are heads (and therefore are of the same basic grammatical type although they can differ considerably in details), in which any one is marked as a CONJ of type a_0 while all the rest are marked as CONJ of the corresponding type a_1. The second and third schemas expand categories marked as [CONJ NIL] simply as the same category unmarked, and those marked [CONJ a] as *and* X, *both* X, etc. for a in *and*, *both*, etc. Finally the linear precedence rule requires categories marked with CONJ of type *both*, *either*, etc. to precede in the sequence all categories marked with CONJ of type *and*, *but*, etc. The effect is to license expansions of any category X into one of

X *and* X
X *or* X
neither X *nor* X
X ... X *and* X
X *and* X ... *and* X
X ... X *or* X
neither X *nor* X ... *nor* X

Such an account is easy to capture in $L^2_{K,P}$. We will assume that $\boldsymbol{P}$ is the (finite) set of feature matrices employed in the GPSG account.[5] One can assert, for example, that the $\langle \textit{and}, \mathrm{NIL}\rangle$ instance of the iterating coordination schema is satisfied in the local tree rooted at the node assigned to x with:

$$\begin{aligned}(\exists y_0, y_1)[\quad & x \triangleleft y_0 \wedge x \triangleleft y_1 \wedge \\ & \mathrm{H}(y_0) \wedge [\mathrm{CONJ}\ \textit{and}](y_0) \wedge \\ & (\forall z)[(x \triangleleft z \wedge z \not\approx y_0) \rightarrow (\mathrm{H}(z) \wedge [\mathrm{CONJ\ NIL}](z))]]\end{aligned}$$

This just says that x has at least two children, that one of them is a head and is marked [CONJ *and*], and that all the rest are heads and are marked [CONJ NIL]. The other rule schemas can be treated similarly.

[5]For details of this and other aspects of the treatment see [Rog97]

The linear precedence schema is just as easy to capture:

$(\forall x, y)[((\exists z)[z \triangleleft x \wedge z \triangleleft y] \wedge [\text{CONJ NIL}](x) \wedge [\text{CONJ } \mathit{and}](y)) \rightarrow x \prec y]$.

Which just says that whenever x and y are siblings, x is marked [CONJ NIL] and y is marked [CONJ *and*], then x will precede y.

6. Generalized Local and Recognizable Sets

The iterating coordination schema is stateable in GPSG because GPSG specifically allows the Kleene star to be applied to categories on the right-hand side of its rewrite rules. Thus these grammars are no longer finite, but are still *finitely presentable*. That is to say, the grammar consists of an infinite set of rewrite rules, but that set itself is generated by a finite grammar, in this case a regular grammar. Such a notion of grammar is not new. Langendoen refers to them as *hypergrammars* in [Lan76], where he motivates them with a flat account of coordination quite similar to the GPSG account. A class of hypergrammars is determined by both the class of the generated grammar and the class of the generating grammar, with the weak generative capacity being the larger of the capacities of the two. Here we will focus on the generated grammars, referring to infinite, but finitely presentable, sets of context-free rewrite rules as *generalized CFGs*, and to the sets of trees generated by them as *generalized local sets*. We will refer to the equivalent notions in the realm of tree automata as *generalized tree automata* and *generalized recognizable sets*. We are interested, in particular, in the variants in which the generated grammar (or automaton) is a regular set. This gives a slight generalization of the GPSG style grammars, in that, rather than restricting application of the Kleene star to individual non-terminals, we, in essence, admit arbitrary regular expressions on the rhs of rewrite rules.

Definition 10 (Regular Generalized Context-free Grammars)**.** *A* regular generalized context-free grammar *is a regular set of local trees over some alphabet* Σ*:*

$$G \subseteq \Sigma \times \Sigma^*, \ \textit{regular}.$$

Definition 11 (Regular Generalized Local Sets)**.** *A set of trees is a* regular generalized local set *iff it is admitted by a regular generalized CFG in the sense of Definition 3.*

Definition 12 (Regular Generalized Tree Automata)**.** *A* regular generalized tree automaton *over an alphabet* Σ *and a finite set of states* Q *is a* *regular set of triples:*

$$\mathcal{A} \subseteq \Sigma \times Q \times Q^*, \text{ regular.}$$

Definition 13 (Regular Generalized Recognizable Sets)**.** *A set of trees is a* regular generalized recognizable set *iff it is $\mathcal{A}(Q_S)$ for some regular generalized tree automaton $\mathcal{A}$ and Q_S, where $\mathcal{A}(Q_S)$ is defined as in Definition 5.*

As the fact that the iterating coordination schema of GPSG can be captured in $L^2_{K,P}$ suggests, $L^2_{K,P}$ suffices to define any regular generalized local set, and in fact any regular generalized recognizable set.

Lemma 3. *Every regular generalized recognizable set is definable in $L^2_{K,P}$.*

Proof. The construction of the proof of Theorem 5 is limited to bounded branching, first of all, because it employs individual variables to pick out the children of a node and only a fixed number of these can occur in a formula. We can circumvent this obstacle by using a single set variable in their stead, but this is still not quite enough. Since the set of triples is, in general, infinite, the disjunction over the formulae encoding the triples which is used to require one of these to hold at every local tree will not necessarily be finite. The way we overcome this is to follow the approach used in capturing the iterating coordination scheme. Rather than having a distinct formula for each triple, we use a distinct formula for each label/state pair $\langle P, Q\rangle \in \boldsymbol{P} \times \boldsymbol{Q}$, which will require the string of states assigned to the children of a point with label P and state Q to be in the set of sequences of states associated with $\langle P, Q\rangle$ in $\mathcal{A}$. (If we regard the triples of the automaton as a set of $(\boldsymbol{P} \times \boldsymbol{Q}) \cup \boldsymbol{Q}$-labeled local trees, this set is just the child language $\mathrm{Ch}_{\langle P,Q\rangle}(\mathcal{A})$.) Since $\mathcal{A}$ is a regular generalized tree-automaton, this set of sequences is a regular language and thus, by Theorem 6, definable in S1S. Now, the set of children of a node, when ordered by linear precedence, is isomorphic to a unary branching tree. It follows that the set of sequences is definable in terms of $\prec$ (and some auxiliary labels, $\vec{S}$, say) on the set of children of a node. Let $\psi_{\langle P,Q\rangle}(X, \vec{Q}, \vec{S})$ define $\mathrm{Ch}_{\langle P,Q\rangle}(\mathcal{A})$ on X. Using these we can replace the formulae $\varphi_i(x_0, \ldots, x_m, \vec{Q})$ with:

$$\begin{aligned}\varphi_{\langle Q,P\rangle}(x_0, X, \vec{Q}) \quad = \quad & (P(x_0) \wedge Q(x_0) \wedge (\forall y)[x_0 \triangleleft y \leftrightarrow X(y)]) \rightarrow \\ & (\exists \vec{S})[\psi_{\langle P,Q\rangle}(X, \vec{Q}, \vec{S})],\end{aligned}$$

Since $\boldsymbol{P}$ and $\boldsymbol{Q}$ are both finite, there are but finitely many of such formulae. With this modification, the proof of Theorem 5 goes through. □

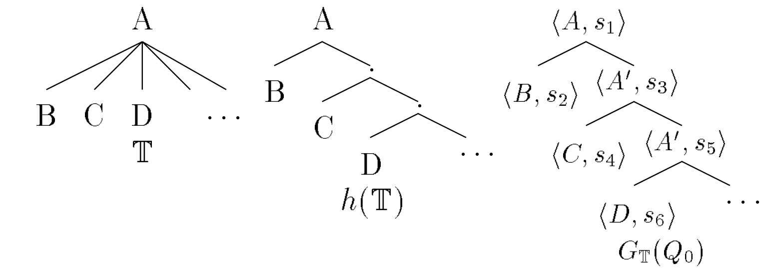


FIGURE 5. Proof of Lemma 4

Corollary 3. *Every regular generalized local set is definable in* $L^2_{K,P}$.

This follows from the fact that the regular generalized local sets are a subclass of the regular generalized recognizable sets.

The question now arises whether we can do more than this. While $L^2_{K,P}$ clearly cannot capture any class stronger than the context-free generalized recognizable sets (since such sets can have non-context free yield languages) is it possible to capture the context-free generalized recognizable sets in $L^2_{K,P}$? The answer is no.

Lemma 4. *Every set of finite trees definable in* $L^2_{K,P}$ *is a regular generalized recognizable set.*

Proof. Suppose $\mathbb{T}$ is a set of finite $\boldsymbol{P}$-labeled trees definable in $L^2_{K,P}$. The fact that $\mathbb{T}$ is definable in $L^2_{K,P}$ implies that it is definable in SωS, which in turn implies that $h(\mathbb{T})$ ($\stackrel{\text{def}}{=} \{h(\mathcal{T}) \mid \mathcal{T} \in \mathbb{T}\}$) is definable in S2S (where h is the embedding of SωS into S2S). From the proof of Lemma 1 we have, then, that there is CFG $G_{\mathbb{T}} \subseteq (\boldsymbol{P'} \times \boldsymbol{Q}) \times (\boldsymbol{P'} \times \boldsymbol{Q})^*$ and some $Q_S \subseteq \boldsymbol{P'} \times \boldsymbol{Q}$ such that $h(\mathbb{T}) = \pi_1(G_{\mathbb{T}}(Q_S))$. (Here $\boldsymbol{P'}$ extends $\boldsymbol{P}$ with some arbitrary labels for the nodes falling between the nodes in the range of h.) Now, $\mathbb{T}$ is certainly recognizable in the sense that the states (and hence labels) of any set of children depend only on the state and label of their parent (this follows from the fact that $G_{\mathbb{T}}(Q_S)$ is local), so it is a generalized recognizable set of some sort. It remains only to show that the child languages of $\mathbb{T}$ are regular. Note that, for any $\langle P, Q\rangle \in \boldsymbol{P} \times \boldsymbol{Q}$ we can pick out the subset of $G_{\mathbb{T}}$ that licenses the images in $G_{\mathbb{T}}(Q_S)$ of local trees in $\mathbb{T}$ rooted at nodes labeled P and assigned state Q; and that these form a CFG $G_{\langle P,Q\rangle}$ generating, in essence, $\mathrm{Ch}_{\langle P,Q\rangle}(\mathbb{T})$ (modulo taking the first projection of the leaves). In the example of Figure 5, for the pair

$\langle A, s_1\rangle$ this would include

$$\begin{aligned}\langle A, s_1\rangle &\longrightarrow B\ \langle A', s_3\rangle\\ \langle A', s_3\rangle &\longrightarrow C\ \langle A', s_5\rangle\\ \langle A', s_5\rangle &\longrightarrow D\ \langle A', s_7\rangle\\ &\vdots\end{aligned}$$

Because h maps local trees in $\mathbb{T}$ to right branching trees these CFGs are all right linear. Consequently, $\mathrm{Ch}_{\langle P,Q\rangle}(\mathbb{T})$ is regular for all P and Q and $\mathbb{T}$ is a regular generalized recognizable set. □

This, then, establishes the strong generative power of $L^2_{K,P}$ with unrestricted branching.

Corollary 4. *A set of finite trees is definable in* $L^2_{K,P}$ *iff it is a regular generalized recognizable set, equivalently, iff it is a projection of a regular generalized local set.*

7. Conclusions

The characterization of the recognizable sets in terms of definability in $L^2_{K,P}$ was originally developed as a means of establishing language-theoretic complexity results for theories of syntax that are presented as systems of constraints on the structure of trees. Here we have shown that if we don't require $L^2_{K,P}$ theories to explicitly establish the finite bound on branching that is a characteristic of recognizable sets then the class of sets they can define is a mild generalization of the class of sets of trees definable by GPSG-style grammars. Thus the natural strong generative capacity of $L^2_{K,P}$ coincides with a linguistically natural class of sets of structures.

When coupled with the flexibility of $L^2_{K,P}$ in expressing principles and constraints occurring in a variety of theories of syntax, this offers a new perspective on relationships between these theories. We have, for instance, shown that $L^2_{K,P}$ suffices to capture both a substantially complete GB account of English D- and S-structure [Rog96a] and the syntactic component of the GPSG account of English of [GKPS85] (see [Rog97]). Because $L^2_{K,P}$ gives a common formalization that is independent of the grammars and mechanisms employed to present these theories, it provides a framework for direct comparison of the properties of the sets of structures they license. In the case of GB and GPSG, issues that have been the focus of many contrasts of the two—like the distinction between transformational and monostratal theories—turn out to have little in the way of actual formal consequences. Distinctions that have been previously overlooked, on the other hand, in particular differences in the nature of linguistic universals, appear to be formally more significant [Rog96b]. As this model-theoretic approach is extended to more complex classes of sets of structures and is applied to a broader range of theories it promises to offer considerable insight into the regularities of natural language syntax that transcend specific theories and to the issues that actually distinguish them.

References

[Büc60] J. R. Büchi. Weak second-order arithmetic and finite automata. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 6:66–92, 1960.

[Don70] John Doner. Tree acceptors and some of their applications. *Journal of Computer and System Sciences*, 4:406–451, 1970.

[GKPS85] Gerald Gazdar, Ewan Klein, Geoffrey Pullum, and Ivan Sag. *Generalized Phrase Structure Grammar.* Harvard University Press, 1985.

[Gor67] Saul Gorn. Explicit definitions and linguistic dominoes. In John F. Hart and Satoru Takasu, editors, *Systems and Computer Science, Proceedings of the Conference held at Univ. of Western Ontario, 1965.* Univ. of Toronto Press, 1967.

[GS84] Ferenc Gécseg and Magnus Steinby. *Tree Automata.* Akadémiai Kiadó, Budapest, 1984.

[Kay84] Richard S. Kayne. *Connectedness and Binary Branching.* Foris, Dordrecht, 1984.

[Lan76] D. Terrence Langendoen. On the weak generative capacity of infinite grammars. *CUNYForum*, 1:13–24, 1976.

[Rab69] Michael O. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the American Mathematical Society*, 141:1–35, July 1969.

[Rog96a] James Rogers. *A Descriptive Approach to Language-Theoretic Complexity.* Studies in Logic, Language, and Information. CSLI Publications, 1996. To appear.

[Rog96b] James Rogers. A model-theoretic framework for theories of syntax. In *34th Annual Meeting of the Association for Computational Linguistics*, UC Santa Cruz, 1996.

[Rog96c] James Rogers. Strict LT2 : Regular :: Local : Recognizable. Paper presented to LACL'96—Logical Aspects of Computational Linguistics, Villers-lès Nancy, FR, 1996.

[Rog97] James Rogers. "Grammarless" phrase structure grammar. *Linguistics and Philosophy*, 1997. To Appear.

[Tha67] J. W. Thatcher. Characterizing derivation trees of context-free grammars through a generalization of finite automata theory. *Journal of Computer and System Sciences*, 1:317–322, 1967.

University of Central Florida
E-mail address: jrogers@cs.ucf.edu