TREE-ORIENTED PROOFS OF SOME THEOREMS ON CONTEXT-FREE AND INDEXED LANGUAGES

William C. Rounds
Case Western Reserve University
Cleveland, Ohio 44106

Summary

In this paper we study some applications and generalizations of the yield theorem: the yield of a recognizable set of trees (dendrolanguage) is an indexed language [1]. Standard results on context-free languages can be obtained quickly using this theorem. We consider here the Peters-Ritchie theorem [4]: the language analyzable by a finite set of CS rules is CF.

An extension of the yield theorem reads: the yield of a CF set of trees is an indexed language. We prove some closure properties of CF sets of trees. Applying the yield theorem, we obtain properties of indexed languages. As a special result, we can solve the infiniteness problem for such languages.

Introduction

Recent work on generalized finite automata has shown that many standard facts in the theory of CF languages can be obtained by considering, instead of grammars, finite automata which check derivation or parsing trees. As examples of this technique (for motivational purposes) we will sketch tree automaton arguments which prove: (1) CF languages are closed under intersection with regular sets; and (2) the result of Peters and Ritchie [4], presented last year at this conference, that the language analyzable by a finite set of CS rules is a CF language. These techniques are not original; Thatcher [8], Rabin [5], and others have made use of the same constructions, but we would like to use them here as background for the similar applications to the theory of indexed languages.

In the second section of the paper, our concern will be context-free sets of trees. Here we are using still another definition of CF grammar on trees, but by a theorem in [6], it is equivalent to the one we presented here last year. This definition is a natural extension to trees of the ordinary definition of CF grammar for strings. By the yield theorems, CF sets of trees bear the same relation to indexed languages as recognizable sets of trees do to CF languages. Some indexed-language theorems are obtained using this fact; hopefully, more such theorems could be proved similarly.

We assume a familiarity with basic tree automata theory. We adopt the notation of Thatcher [9] with minor changes. Although we have written out detailed constructions and inductive assertions, we have omitted detailed inductive proofs.

Notational Preliminaries

Let (Σ,r) be a ranked alphabet, where $r \subseteq \Sigma \times \mathbb{N}$. Define $\Sigma_n = r^{-1}\{n\}$. Let I be a set disjoint from Σ. The set $\mathcal{T}_\Sigma(I)$ of terms (trees) indexed by I is the smallest set such that

(i) $\Sigma_o \cup I \subseteq \mathcal{T}_\Sigma(I)$

(ii) $A \in \Sigma_n$, and $t_o, \ldots, t_{n-1} \in \mathcal{T}_\Sigma(I)$ imply $A(t_o, \ldots, t_{n-1}) \in \mathcal{T}_\Sigma(I)$.

In particular, I is often a countable set $X = \{x_o, x_1, \ldots\}$ of variables. $\mathcal{T}_\Sigma(X)$ is written $\mathcal{T}_\Sigma$, and $\mathcal{T}_\Sigma(\emptyset)$ is written $\mathcal{T}^o_\Sigma$.

A Σ-automaton $\mathcal{A}$ consists of a finite set Q of states, together with a specification for each $A \in \Sigma_n$ a partial function

$$f_A: Q^n \to Q.$$

If $\lambda \in \Sigma_o$, f_λ is either a constant or undefined. Each automaton also has a set $F \subseteq Q$ of designated final states. Every automaton has an associated response function $\| \ \|_{\mathcal{A}}$:

(i) $\|\lambda\|_{\mathcal{A}} = f_\lambda$ for $\lambda \in \Sigma_o$ such that f_λ is defined;

(ii) $\|A(t_o,\ldots,t_{n-1})\|_{\mathcal{A}} = f_A(\|t_o\|_{\mathcal{A}},\ldots,\|t_{n-1}\|_{\mathcal{A}})$

iff all $\|t_i\|_{\mathcal{A}}$ are defined, and f_A is defined for these values.

$t \in \mathcal{T}^o_\Sigma$ is accepted by $\mathcal{A}$ iff $\|t\|_{\mathcal{A}}$ is defined and in F.

This is the so-called "frontier-to-root" version of a tree automaton. No loss of general theory results from allowing partial transition functions.

A dendrolanguage over Σ is a subset of $\mathcal{T}^o_\Sigma$. A dendrolanguage $\mathcal{L}$ is recognizable iff for some automaton $\mathcal{A}$, we have

$$\mathcal{L} = \{t \mid t \text{ is accepted by } \mathcal{A}\}.$$

Define the function yield: $\mathcal{T}^o_\Sigma \to \Sigma_o^+$ by induction:

$$\text{yield}(\lambda) = \lambda \quad \text{for } \lambda \in \Sigma_o;$$

$$\text{yield}(A(t_o,\ldots,t_{n-1})) = \text{yield}(t_o)\cdot\ldots\cdot\text{yield}(t_{n-1}).$$

For later purposes, we want another yield function for indexed terms in $\mathcal{T}_\Sigma(I)$. This is the "indexed yield" $y_I: \mathcal{T}_\Sigma(I) \to I^*$

$$y_I(i) = i \text{ for } i \in I;$$

$$y_I(\lambda) = \Lambda \text{ (empty string) for } \lambda \in \Sigma_o;$$

$$y_I(A(t_o,\ldots,t_{n-1}) = y_I(t_o) \cdot \ldots \cdot y_i(t_{n-1}).$$

If $u \in \mathcal{T}_\Sigma(I)$ and $y_I(u) = i_o \ldots i_{p-1}$, we will write $u = u[i_o,\ldots,i_{p-1}]$.

Context-free Language Applications

The yield theorem (Mezei-Wright [3], Thatcher [9]), states that a language $L \subseteq \Sigma_o^+$ is CF if and only if $L = \text{yield}[\mathcal{L}]$ for $\mathcal{L}$ a recognizable dendrolanguage. We can show without any further preliminaries how this theorem gives us results about CF languages.

Lemma. Let $R \subseteq \Sigma_o^+$ be a regular set. Then $\{t \in \mathcal{T}_\Sigma^o \mid \text{yield}(t) \in R\}$ is a recognizable dendrolanguage.

Proof. (Rabin, personal comm.) Recall that R is regular iff there is a finite semigroup S, epimorphism $\varphi: \Sigma_o^* \to S$, and subset T of S such that $R = \varphi^{-1}[T]$. Define a finite tree automaton $\mathcal{A}$ with state set S, $f_\lambda = \varphi(\lambda)$ for $\lambda \in \Sigma_o$, and

$$f_A(s_o,\ldots,s_{n-1}) = s_o \cdot \ldots \cdot s_{n-1}$$

(product in S.) Let F = T. Then

$$\|t\|_{\mathcal{A}} = \varphi(\text{yield}(t))$$

so that

$$\|t\|_{\mathcal{A}} \in F \iff \varphi(\text{yield}(t)) \in T$$
$$\iff \text{yield}(t) \in R.$$

Corollary. The class of CF subsets of Σ_o^+ is closed under intersection with regular sets.

Proof. Let $L = \text{yield}(\mathcal{L})$ for $\mathcal{L}$ recognizable. Let $\mathcal{R} = \{t \mid \text{yield}(t) \in R\}$. Then

$$L \cap R = \text{yield}(\mathcal{R} \cap \mathcal{L}).$$

But $\mathcal{R} \cap \mathcal{L}$ is recognizable, because $\mathcal{R}$ and $\mathcal{L}$ are. Hence, the result follows by the yield theorem.

We note that since the semigroup S can be effectively obtained, both the lemma and its corollary are effective.

Now let us turn to an application in linguistic theory discovered by Peters and Ritchie [4]. Suppose Γ is a finite set of rules over $\Sigma = N \cup T$, of the form

$$\alpha A \beta \to \alpha \omega \beta$$

where $A \in N$, α and $\beta \in \Sigma^*$, and $\omega \in \Sigma^+$. Σ becomes a ranked alphabet if we define $r(a,o)$ for $a \in T$, and $r(A,n)$ if there is a production $\alpha A \beta \to \alpha \omega \beta$ in Γ with length $(\omega) = n$. For each tree $t \in \mathcal{T}_\Sigma^o$ we define the set of proper analyses $P(t) \subseteq \Sigma^+$ inductively:

(i) $P(a) = \{a\}$ for $a \in T$;

(ii) $P(A(t_o,\ldots,t_{n-1}))$
$$= \{A\} \cup P(t_o) \cdot P(t_1) \cdot \ldots \cdot P(t_{n-1}).$$

Thus, $P[A(B(a,b), C(D(a)))]$
$$= \{A, BC, BD, Ba, abC, abD, aba\}.$$

For the given set Γ of rules, define

$$\Pi = \{A \to \omega \mid \text{for some } (\alpha,\beta),\ \alpha A \beta \to \alpha \omega \beta \in \Gamma\}.$$

If $t' = B(s_o,\ldots,s_{m-1})$ define $\text{top}(t') = B$.

Let $u \in \mathcal{T}_\Sigma^o$ and let $A(t_o,\ldots,t_{n-1})$ be a subtree of u. This subtree is an occurrence of a rule π in Π if π is

$$A \to \text{top}(t_o) \cdot \ldots \cdot \text{top}(t_{n-1}).$$

Given an occurrence of π in u, temporarily replace $A(t_o,\ldots,t_{n-1})$ by $\bar{A}(t_o,\ldots,t_{n-1})$ where $\bar{A}$ is a new symbol. Let $\bar{u}$ be the tree so obtained. Consider the set of proper analyses of $\bar{u}$ of the form $x\bar{A}y$. Then, the set of analyses of u which include the occurrence of π is the set

$$\{x A y \mid x \bar{A} y \text{ is a p.a. of } \bar{u}\}.$$

We say that $t \in \mathcal{T}_\Sigma^o$ is analyzable by the set Γ if every subtree of t is an occurrence of some rule in Π (except the terminal subtrees) and for every occurrence of a rule $A \to \omega$ in t, there is a proper analysis $\rho \alpha A \beta \rho'$ of t which includes the occurrence of $A \to \omega$ and such that $\alpha A \beta \to \alpha \omega \beta \in \Gamma$. Note: if $\rho x A y \rho'$ is a proper analysis including the occurrence of $A \to \omega$, we call (x,y) a context of $A \to \omega$.

Theorem. (Peters and Ritchie.) The set of all trees analyzable by Γ is a recognizable dendrolanguage.

(If we define the language analyzable by Γ to be the yield of the dendrolanguage analyzable by Γ, the result as stated by Peters and Ritchie follows immediately by the yield theorem.)

Proof. We have only to construct a tree automaton to check a tree for compatibility with the rules Γ. If we think in terms of a bottom-up sweep, we immediately see that this construc-

tion can be done. We try to determine all possible contexts of every occurrence of rules $A \to \omega$ in a given tree t, and to check that at least one context is a pair (α,β) such that $\alpha A\beta \to \alpha\omega\beta$ is in Γ. Let m be the maximum length of α, or β, with (α,β) such a pair. We have only to remember contexts (x,y) such that $\ell(x)$ and $\ell(y) \leq m$. We first introduce some notation. For $L \subseteq \Sigma^*$, let

$$In_m(L) = \{x \in \Sigma^* \mid \ell(x) \leq m \text{ and } (\exists y)\ xy \in L\}$$

$$Fin_m(L) = \{x \in \Sigma^* \mid \ell(x) \leq m \text{ and } (\exists y)\ yx \in L\}$$

$$(L)_m = \{x \in L \mid \ell(x) \leq m\}.$$

For $t \in \mathcal{T}^o_\Sigma$, define

$$S(t) = In_m(P(t))$$

$$S'(t) = Fin_m(P(t))$$

$$Y(t) = (P(t))_m.$$

The state set of the automaton $\mathcal{A}$ will now be constructed. The states will be tuples of the form

$$(A,\varphi,S,S',Y)$$

where $A \in \Sigma$; S,S', and Y are subsets of $(\Sigma^*)_m$, and φ is a function with domain a subset of Π such that for each $\pi \in \text{dom}\ \varphi, \varphi(\pi)$ is a pair (C,g) where $C \subseteq (\Sigma^*)_m \times (\Sigma^*)_m$ is the set of contexts of π and g is a function from C to $\{0,1\} \times \{0,1\}$ giving information about completed contexts. We want

$$\|t\| = (A,\varphi,S,S',Y)$$

if and only if $A = \text{top}(t)$, $S = S(t)$, $S' = S'(t)$, $Y = Y(t)$, and if $\varphi(\pi) = (C,g)$ then C will be the set of contexts of π taken over arbitrary occurrences of π in t, and g tells for each $(\alpha,\beta) \in C$ whether or not there is a proper analysis (p.a.) of t, say $\rho\alpha A\beta\rho'$, including π with ρ empty, ρ' empty, both, or neither. Specifically, $g(\alpha,\beta) = (0,0)$ if every p.a. $\rho\alpha A\beta\rho'$ is such that $\rho \neq \Lambda$ and $\rho' \neq \Lambda$; $g(\alpha,\beta) = (0,1)$ if there is a p.a. $\rho\alpha A\beta\rho'$ with $\rho' = \Lambda$ but none with $\rho = \Lambda$; $g(\alpha,\beta) = (1,0)$ if there is a p.a. with $\rho = \Lambda$ but none with $\rho' = \Lambda$, and is $(1,1)$ if there is a p.a. $\alpha A\beta$. Note that this last condition is equivalent to the denial of the other three, because if $\alpha A\beta\rho'$ and $\rho\alpha A\beta$ are two p.a.'s with $\rho' \neq \Lambda$, $\rho \neq \Lambda$, then $\alpha A\beta$ is also a p.a.

The foregoing description is actually the inductive assertion to be verified for the response function of our automaton. If we let $(A,\varphi,S,S',Y) \in F$ if and only if for each $\pi \in \text{dom}\ \varphi$, there is an (α,β) in C where $\varphi(\pi) = (C,g)$, such that $\alpha A\beta \to \alpha\omega\beta \in \Gamma$ (where π is $A \to \omega$), we will have the desired result.

Now let us construct the transition functions. First, for $a \in \Sigma_o$, define

$$f_a = (a,\emptyset,\{a\},\{a\},\{a\}).$$

Here, the function φ is totally undefined. Now suppose that for $i \leq n-1$, we are given the states $(A_i,\varphi_i,S_i,S_i',Y_i)$, and $A \in \Sigma_n$. We show how to calculate a new state

$$(A,\varphi,S,S',Y).$$

First of all, $A \to A_o \ldots A_{n-1}$ must be a rule $\pi_o \in \Pi$. Otherwise the next state will be undefined. Assuming that $A \to A_o \ldots A_{n-1}$ is legitimate, let us construct S, S' and Y. To construct S, form the set

$$\bar{S} = \{A\} \cup S_o \cup Y_o S_1 \cup Y_o Y_1 S_2 \cup \ldots$$

$$\cup\ Y_o \cdot \ldots \cdot Y_{n-2} S_{n-1} \cup Y_o \cdot \ldots \cdot Y_{n-1}.$$

Let $S = In_m(\bar{S})$. Similarly, let $S' = Fin_m(\bar{S}')$, where

$$\bar{S}' = \{A\} \cup S'_{n-1} \cup S'_{n-2}Y_{n-1} \cup S'_{n-3}Y_{n-2}Y_{n-1} \cup \ldots$$

$$\cup\ S'_o Y_1 \cdot \ldots \cdot Y_{n-1} \cup Y_o \cdot \ldots \cdot Y_{n-1}.$$

Define $Y = (Y_o \cdot \ldots \cdot Y_{n-1})_m$.

Now let us construct φ. For each $i < n$ define

$$\bar{S}_i = \{\Lambda\} \cup S_{i+1} \cup Y_{i+1}S_{i+2} \cup Y_{i+1}Y_{i+2}S_{i+3} \cdots$$

$$\cup\ Y_{i+1} \cdot \ldots \cdot Y_{n-1}$$

and

$$\bar{S}'_i = \{\Lambda\} \cup S'_{i-1} \cup S'_{i-2}Y_{i-1} \cup \ldots$$

$$\cup\ S'_o Y_1 \cdot \ldots \cdot Y_{i-1} \cup Y_o \cdot \ldots \cdot Y_{i-1}.$$

For $\pi \in \text{dom}\ \varphi_i$, and $(\alpha,\beta) \in C_i$, where $\varphi_i(\pi) = (C_i,g_i)$, define

$$H_i^\pi(\alpha,\beta) = \begin{cases} \{(\alpha,\beta)\} & \text{if } g_i(\alpha,\beta) = (0,0) \\ \{\alpha\} \times In_m(\beta\bar{S}_i) & \text{if } g_i(\alpha,\beta) = (0,1) \\ Fin_m(\bar{S}'_i\alpha) \times \{\beta\} & \text{if } g_i(\alpha,\beta) = (1,0) \\ Fin_m(\bar{S}'_i\alpha) \times In_m(\beta\bar{S}_i) & \text{otherwise.} \end{cases}$$

Set $H_i(\pi) = \bigcup_{(\alpha,\beta)\in C_i} H_i^\pi(\alpha,\beta)$.

Define $\varphi(\pi) = (C,g)$, where

$$C = \begin{cases} \bigcup_i H_i(\pi) & \text{if } \pi \neq \pi_o \\ \bigcup_i H_i(\pi) \cup \{(\Lambda,\Lambda)\} & \text{if } \pi = \pi_o. \end{cases}$$

Define $g(\alpha,\beta)$ as follows: Let $T_i = In_m(Y_o \cdot \ldots \cdot Y_{i-1} S_i)$ and also $T_i' = Fin_m(S_i' Y_{i+1} \cdot \ldots \cdot Y_{n-1})$. If for each i, $\alpha \notin T_i$ and $\beta \notin T_i'$, put $g(\alpha,\beta) = (0,0)$. If for some i, $\alpha \in T_i$, but for each i, $\beta \notin T_i'$, put $g(\alpha,\beta) = (1,0)$. Conversely, if for each i, $\alpha \notin T_i$ but for some i, $\beta \in T_i'$, put $g(\alpha,\beta) = (0,1)$. Otherwise, put $g(\alpha,\beta) = (1,1)$. Note that $g(\Lambda,\Lambda) = (1,1)$.

This completes the construction. It may be verified that the inductive assertion holds for these transition functions.

Remarks. We were not able to understand the proof of this theorem as presented in [4]. In particular, we were not able to verify that without loss of generality, the maximum-length context has length $m = 1$.

The theorem seems to generalize in various ways. For example, we can allow Γ to be a set of type 0 rules $\alpha A \beta \to \alpha \omega \beta$ with ω possibly empty. Trees to be checked in this case will have a special leaf e which counts as a member of Σ_o but which is interpreted as the empty string by the checking automaton. The yield theorem tells us that the analyzable language is CF over the set $\Sigma_o \cup \{e\}$, but we can obtain the desired language simply by erasing e.

Another generalization which seems to be valid comes when we allow scattered contexts in the definition of analyzability. Instead of requiring that for each occurrence of $A \to \omega$, there be a p.a. $\rho \alpha A \beta \rho'$ with $\alpha A \beta \to \alpha \omega \beta \in \Gamma$, we require only a p.a.

$$x_1 a_1 x_2 \cdots a_{k-1} x_k A y_1 \beta_1 y_2 \cdots \beta_{j-1} y_j$$

for some j and k, x_i, y_i, such that $a_1 a_2 \cdots a_{k-1} = \alpha$, $\beta_1 \cdots \beta_{j-1} = \beta$, and $\alpha A \beta \to \alpha \omega \beta \in \Gamma$.

The theorem stating that the language accepted by a pda is context-free follows from a Peters-Ritchie argument about trees which represent computations by the pda. Constructing the standard Turing machine rewriting system for a pda, one is forced to put in context-sensitive rules for states in which the stack symbol is erased. By using essentially these rules as analyzing rules, however, one can show that the analyzed language is the language accepted by the pda; hence it is CF.

Context-free Dendrolanguages

In this section we discuss a definition of grammar on trees which is a natural generalization of grammars on strings. We shall not prove it here, but this type of grammar is equivalent to the grammars which were presented at this conference last year [7]. (A proof of the equivalence can be found in [6].)

Trees will have two types of nodes: finished and operative. Finished nodes are analogous to terminal symbols in the string case, but may appear anywhere in a tree. Productions apply only to operative nodes, in the same way that CF productions apply to nonterminal symbols. The effect will be to replace the operative node by the right hand side of the production which has the subtrees of the original node grafted (substituted) into it.

Let (Σ,r) be a ranked alphabet, and assume $\Sigma = F \cup N$, where $F \cap N = \emptyset$. N is the set of operative symbols, and F the set of finished symbols. Productions are pairs $(A(x_o,\ldots,x_{n-1}),u)$, also written $A(x_o,\ldots,x_{n-1}) \to u$, such that $u \in \mathcal{T}_\Sigma(\{x_o,\ldots,x_{n-1}\})$, and $r(A,n)$. They apply as in this example: Let $t = a(b(x_o,x_1),B(C(a),D(x_o)))$. Suppose $F = \{a,b,c\}$, $N = \{B,C,D\}$. Let $B(x_o,x_1) \to B(b(x_1,x_o),C(x_o))$ be a production. Then

$$B(C(a),D(x_o)) \Rightarrow B(b(D(x_o),C(a)),C(C(a))) = \beta.$$

Also

$$t \Rightarrow a(b(x_o,x_1),\beta).$$

Formally, we give an inductive definition of direct generation via a production π.

(i) If $t = \lambda \in \Sigma_o$, then $t \Rightarrow_\pi t'$ iff π is $\lambda \to t'$, where $t' \in \mathcal{T}_\Sigma^o$.

(ii) If $t = x \in X$, then there is no such tree t';

(iii) If $t = A(t_o,\ldots,t_{n-1})$, then either for some i, $t_i \Rightarrow_\pi t_i'$ and t' is $A(t_o,\ldots,t_i',\ldots,t_{n-1})$, or else π is $A(x_o,\ldots,x_{n-1}) \to u$, and t' is obtained from u by substituting t_i for each occurrence of x_i in u.

Definition. A context-free dendrogrammar is a 3-tuple (Σ,S_o,Π) where Σ is a ranked alphabet $(\Sigma = F \cup N)$, $S_o \subseteq \mathcal{T}_\Sigma^o$ is a finite set of starting trees, and Π is a finite set of productions.

Write $t \Rightarrow t'$ if for some $\pi \in \Pi$, $t \Rightarrow_\pi t'$. Let $\Rightarrow^*$ be the reflexive, transitive closure of $\Rightarrow$. Define

$$\mathcal{L}(G) = \{u \in \mathcal{T}_F^o \mid (\exists s \in S_o)(s \Rightarrow^* u)\}.$$

Example. $F = \{a,b,c,k,h\}$; $N = \{H,K\}$. Let $S_o = \{H(a,b,c)\}$. Take 3 productions:

$$H(x_o,x_1,x_2) \to H(K(a,x_o),K(b,x_2),K(c,x_2))$$

$$H(x_o,x_1,x_2) \to k(h(x_o,x_1,x_2),h(x_o,x_1,x_2))$$

$$K(x_o,x_1) \to k(x_o,x_1).$$

For a derivation in G, consider

$$H(a,b,c) \Rightarrow H(K(a,a),K(b,b),K(c,c))$$

$$\Rightarrow k(h(K(a,a),K(b,b),K(c,c)),h(K(a,a),K(b,b),K(c,c)))$$

$$\Rightarrow^* k(h(k(a,a),k(b,b),k(c,c)),h(k(a,a),k(b,b),k(c,c))).$$

Apparently, $\text{yield}[\mathcal{L}(G)] = \{a^n b^n c^n a^n b^n c^n \mid n \geq 1\}$.

This definition of dendrogrammar is plainly related to the definition of unrestricted macro grammar by M. Fischer [2]. Given a CF dendrogrammar G, one can immediately define a macro grammar G' such that $L(G') = \text{yield}[\mathcal{L}(G)]$. Conversely given a macro grammar with no rules of the form $S \to \Lambda$ and no productions containing Λ as an argument, one can define a dendrogrammar with the same property. Fischer's theorem [2, theorem 5.3] thus implies the extended yield theorem: the yield of a CF dendrolanguage is an indexed language; and conversely, except for the empty string. By considering $\mathcal{L}(G)$ instead of its yield, we may be able to say something about indexed languages in the same way we did about context-free languages.

We want to state some easy consequences of the definitions. These results are all generalizations of corresponding results for CF grammars. First, we have the tree analogue of left-to-right derivations.

Definition. A derivation in a CF dendrogrammar is top-down (Fischer: outside-in) if whenever a production $A(x_o,\ldots,x_{n-1}) \to u$ is applied, the node A has no operative nodes above it. More formally, $t \Rightarrow_\pi t'(TD)$ iff t' is either $A(t_o,\ldots,t_{n-1})$ and π applies to A, or t is $a(t_o,\ldots,t_{n-1})$ with $a \in F$, and for some i, $t_i \Rightarrow_\pi t_i'(TD)$.

Lemma (Fischer). Given a CF dendrogrammar G, every tree in $\mathcal{L}(G)$ can be obtained via a top-down derivation (with possibly many more steps).

Trying to generalize the standard proof for strings, one realizes why many more steps are needed in a top-down derivation. Also, it is clear why an analogous lemma for "bottom-up" derivations is false: trees are not reversible.

Definition. A CF dendrogrammar is in normal form if each production is in one of the two forms

(1) $A(x_o,\ldots,x_{n-1}) \to a(x_o,\ldots,x_{n-1})$,

where $A \in N$, $a \in F$, or

(2) $A(x_o,\ldots,x_{n-1}) \to u$

where $u \in \mathcal{T}_N(\{x_o,\ldots,x_{n-1}\})$.

Lemma. For every CF dendrogrammar G, we can effectively find a grammar G' in normal form with $\mathcal{L}(G') = \mathcal{L}(G)$.

This lemma is again a special case of a result of Fischer which gives a strict normal form for OI grammars. Our result is, however, trivial. The form we have here is the analogue of the normal form $A \to a$; $A \to A_1A_2\ldots A_p$; for CF grammars.

Define a production $A(x_o,\ldots,x_{n-1}) \to u$ to be useless if there is no tree $t \in \mathcal{T}_F(X)$ such that $u \Rightarrow_G^* t$.

Lemma. If G' is obtained from G by discarding all useless productions, then $\mathcal{L}(G') = \mathcal{L}(G)$.

Clearly $\mathcal{L}(G') \subseteq \mathcal{L}(G)$. If $t \in \mathcal{L}(G)$, consider the sequence of productions applied to derive t. Let $t_i \Rightarrow_\pi t_{i+1}$ be the last application of a useless production. t_i has a subtree $A(u_o,\ldots,u_{k-1})$, and t_{i+1} has a subtree $\bar{u}$ such that A is transformed via π, and there is no tree t' in $\mathcal{T}_F^o$ with $\bar{u} \Rightarrow_G^* t'$. We can therefore assume that no further productions apply in $\bar{u}$; thus that if $t_{i+1} \Rightarrow_{\pi'} t_{i+2}$, then $t_i \Rightarrow_{\pi'} t_{i+2}$. This eliminates the use of π.

The emptiness problem is solvable for indexed languages; hence for CF dendrolanguages. Thus we can decide if a given production is useless.

Finally, we notice that CFD's are closed under union, and that we can take the starting set to consist of a single symbol in Σ_o.

II Closure Properties

One of the first theorems one would like to prove is that the class of CF dendrolanguages is closed under intersection with recognizable sets. This is the case, and we shall prove it as a consequence of closure under certain finite-state mappings. (A general treatment of finite-state mappings can be found in [6].)

Definition. A (nondeterministic) linear finite-state transformation is a 4-tuple

$$T = (\Sigma, Q, Q_o, \Pi)$$

where Σ is a ranked alphabet, Q is a finite set of states, $Q_o \subseteq Q$ are the initial states, and Π is a finite set of productions

$$a(x_o,\ldots,x_{n-1}) \to \alpha$$

where $\alpha \in \mathcal{T}_\Sigma(Q \times \{x_o,\ldots,x_{n-1}\})$. Further if the indexed yield of α is

$$(q_o,x_{i_o})(q_1x_{i_1})\ldots(q_{p-1}x_{i_{p-1}})$$

we require that $i:\{0,\ldots,p-1\}\to\{0,\ldots,m-1\}$ be injective. (This means that every variable x_j occurs at most once in α.)

Linear productions apply to trees in $\mathcal{T}_\Sigma(Q\times\mathcal{T}_\Sigma)$ as follows: Let $(q,a(t_o,\ldots,t_{n-1}))$ be an index of a tree $u\in\mathcal{T}_\Sigma(Q\times\mathcal{T}_\Sigma)$. Let $a(x_o,\ldots,x_{n-1})\to\alpha[(q_o,x_{i_o}),\ldots,(q_{p-1},x_{i_{p-j}})]$ be a production. Form the tree

$$\alpha'=\alpha[(q_o,t_{i_o}),\ldots,(q_{p-1},t_{i_{p-1}})].$$

Replace the given index $(q,a(t_o,\ldots,t_{n-1}))$ of u by α'. The result is the tree u' obtained by applying the given production.

Definition. For $t\in\mathcal{T}_\Sigma$, $q\in Q$, define

$$\overline{T}(q,t)=\{t'\in\mathcal{T}_\Sigma(Q\times\mathcal{T}_\Sigma)\mid(q,t)\Rightarrow_L^* t'\}$$

$$T(q,t)=\{t'\in\mathcal{T}_\Sigma^o\mid(q,t)\Rightarrow_L^* t'\}.$$

For $R\subseteq\mathcal{T}_\Sigma^o$, define

$$T[R]=\bigcup_{\substack{q\in Q_o\\ s\in R}}T(q,s).$$

Theorem. The class of CF dendrolanguages is closed under linear finite-state transformations (effectively.)

Proof. Let G be a normal form dendrogrammar from which all useless productions have been eliminated. (This can be done effectively.) Let T be a linear transformation. Our technique is a standard one -- run T and G simultaneously. G will carry out top-down derivations; as soon as G produces finished symbols, T will transform them. Since T is not allowed to make copies of its input, the simultaneous grammar will not ever produce more trees than T could produce, acting on $\mathcal{L}(G)$.

Productions of G' will now be given. Whenever $A(x_o,\ldots,x_{n-1})\to a(x_o,\ldots,x_{n-1})$ is in $\Pi(G)$, put

$$(q,A(x_o,\ldots,x_{n-1}))\to\alpha[(q_o,x_{i_o}),\ldots,(q_{p-1},x_{i_{p-1}})]$$

into $\Pi(G')$, for each production

$$(q,a(x_o,\ldots,x_{n-1}))\to\alpha[(q_o,x_{i_o}),\ldots,(q_{p-1},x_{i_{p-1}})]$$

in $\Pi(T)$. If $A(x_o,\ldots,x_{n-1})\to u\in\Pi(G)$, then in each $q\in Q$, put the production

$$(q,A(x_o,\ldots,x_{n-1}))\to(q,u)$$

into $\Pi(G')$. (Recall $u\in\mathcal{T}_N(X)$).

The productions of G' have the form of creative productions as defined in [6]. They apply to trees in $\mathcal{T}_\Sigma(Q\times\mathcal{T}_\Sigma)$ in the same manner as productions of T do. [The starting trees of G' will be of the form (q,t) where $q\in Q_o, t\in S_o$.] G' is a creative dendrogrammar, exactly like the grammars defined at this conference last year. G' is thus equivalent to a CF dendrogrammar (effectively). The result follows when we show that $\mathcal{L}(G')=T[\mathcal{L}(G)]$. To verify this, we need some notation. Since derivations in G are top-down, we may view intermediate stages as trees of the form $v[v_o,\ldots,v_{m-1}]$ where $v\in\mathcal{T}_F(X)$ and $v_i\in\mathcal{T}_N^o$. We claim that for each $k\geq 0$, that if for some α and p,

$$(q,A(y_o,\ldots,y_{n-1}))\Rightarrow_{G'}^{(k)}\alpha[(q_o,w_o),\ldots,(q_{p-1},w_{p-1})],$$

then there is a $v[v_o,\ldots,v_{m-1}]\in\mathcal{T}_\Sigma^o$ and a 1-1 function $f:\{0,\ldots,m-1\}\to\{0,\ldots,m-1\}$ such that $w_{f(i)}=v_i$, and such that $\alpha[(q_o,x_{f^{-1}(0)}),\ldots,(q_{p-1},x_{f^{-1}(p-1)})]$ is a member of $\overline{T}(q,v[x_o,\ldots,x_{m-1}])$, and $A(y_o,\ldots,y_{n-1})\Rightarrow_G^{(k)}v[v_o,\ldots,v_{m-1}]$. Conversely, if there is such a v and f, then an appropriate α exists.

The proof of this statement and its converse, is by induction on k and is omitted. Linearity of T is essential to the proof.

Having these results, it is easy to see that $T[\mathcal{L}(G)]\subseteq\mathcal{L}(G')$. If $\alpha\in\mathcal{L}(G')$, let $v[v_o,\ldots,v_{m-1}]$ be given by the inductive statement. All productions of G are useful, so each $v_i\Rightarrow_G^*\bar{v}_i$, where $\bar{v}_i\in\mathcal{T}_F^o$. Since $p=o$, and $\alpha\in T(q,v_o,\ldots,v_{m-1}])$ we have $\alpha\in T(q,v[\bar{v}_o,\ldots,\bar{v}_{m-1}])$, so $\alpha\in T[\mathcal{L}(G)]$.

Corollary. The class of CF dendrolanguages is closed under intersection with recognizable sets.

Proof. Every recognizable set is the domain of a linear FST which acts as the identity when defined.

Corollary. Indexed languages are closed under intersection with regular sets.

Proof. This follows by the yield theorem exactly as for CF languages.

An interesting linear transformation is the fan transformation which eliminates unary branches from trees. Specifically,

$$\text{fan}(\lambda)=\lambda;$$

$$\text{fan}(a(t_o))=t_o;$$

$$\text{fan}(a(t_o,\ldots,t_{p+2}))=a(t_o,\ldots,t_{p+2}).$$

If $\mathcal{L}$ is a dendrolanguage, then, by the theorem, so is fan[$\mathcal{L}$]. Moreover, yield[fan[$\mathcal{L}$]] = yield[$\mathcal{L}$], and fan[$\mathcal{L}$] is infinite if and only if yield[$\mathcal{L}$] is infinite. (Recall that the empty symbol does not occur in dendrolanguages.)

Theorem. The infiniteness problem is solvable for indexed languages.

Sketch of proof. Consider an indexed grammar G without Λ-rules. Such a grammar can be converted immediately into a CF dendrogrammar G' with yield[$\mathcal{L}$(G')] = L(G). As we have seen above, by taking fan[$\mathcal{L}$(G')] we reduce the problem to the infiniteness problem for dendrolanguages. The solvability of this problem is a consequence of the fact that if $\mathcal{L}$ is a CF dendrolanguage, then the set of paths through trees of $\mathcal{L}$ is (effectively) a CF language. To decide infiniteness of $\mathcal{L}$, simply decide infiniteness for the set of paths of $\mathcal{L}$. This trick, incidentally, avoids the problem of developing an intercalation theorem [10] for indexed languages. We outline some of the details needed to prove the path result.

For each letter $A \in \Sigma$ let A_n be a new symbol, where $n \geq 0$. For each $\lambda \in \Sigma_o$ define the set of λ-paths through $t \in \mathcal{T}_\Sigma(X)$ inductively:

(i) $P_\lambda(x) = \emptyset, \quad x \in X;$

(ii) $P_\lambda(\mu) = \begin{cases} \emptyset, & \mu \in \Sigma_o,\ \mu \neq \lambda; \\ \{\lambda\}, & \lambda \in \Sigma_o,\ \mu = \lambda. \end{cases}$

(iii) $P_\lambda(A(t_o,\ldots,t_{n-1}) = \bigcup_{i=o}^{n-1} \{A_i(w) | w \in P_\lambda(t_i)\}.$

Example. $P_\lambda(A(B(\lambda),C(\lambda,\mu))$
$= \{A_o(B_o(\lambda)),A_1(C_o(\lambda))\}.$

Define the symbols A_i to have rank 1. Terms in such symbols can be regarded as strings; thus P_λ above can be written

$$\{A_oB_o\lambda,A_1C_o\lambda\}.$$

For $\mathcal{S} \subseteq \mathcal{T}^o_\Sigma$, define

$$P[\mathcal{S}] = \bigcup_{\lambda\in\Sigma_o} \bigcup_{t\in\mathcal{S}} P_\lambda(t).$$

Proposition. If $\mathcal{S}$ is a CF dendrolanguage, then $P[\mathcal{S}]$, regarded as a set of strings, is a CF language.

To prove this, let G be a normal-form, useful CF dendrogrammar. The idea is to take paths of trees occurring in all productions. We must be careful which paths we choose, however. If u is the right-hand side of a production, and x is a variable occurring in u, we want to distinguish the paths of u which end in x. Thus,

(i) $P_x(\lambda) = \emptyset, \quad \lambda \in \Sigma_o$

(ii) $P_x(y) = \begin{cases} \emptyset, & y \neq x; \\ \{x^*\}, & y = x. \end{cases}$

Here, $x^* \notin X$ is a new variable

(iii) $P_x(A(t_o,\ldots,t_{n-1})) = \bigcup_{i=o}^{n-1} \{A_i(w) | w \in P_x(t_i)\}.$

Now if $B(x_o,\ldots,x_{n-1}) \to u$ is a production of G, put the productions $B_i(x^*) \to w$ into G' for each $w \in P_{x_i}(t)$, and each $i < n$. Also, put the productions $B_i(x^*) \to w$ (for each $w \in P_\lambda(t)$, each i, and each λ) into G'.

Let the initial strings of G' be $P[S_o]$ where S_o is the set of initial trees of G. It follows via an easy inductive proof that $\mathcal{L}(G') = P[\mathcal{L}(G)]$. But G' is in essence a CF grammar. If $A_i(x^*) \to w(x^*)$ then let $A_i \to w$ be a CF production. (If $w(x^*) = x^*$ let $A_i \to \Lambda$.) If $A_i(x^*) \to w(\lambda)$, let $A_i \to w\lambda$. Thus, $P[\mathcal{L}(G)]$ is indeed a CF language.

References

[1] Aho, A.V. "Indexed grammars -- an extension of the context-free grammars," JACM 15 (1968), 647-671.

[2] Fischer, M.J. "Grammars with macro-like productions," Proc. 9th IEEE Symp. on Switching and Automata Theory, October, 1968.

[3] Mezei, J. and J.B. Wright, "Algebraic automata and context-free sets," Inf. Control 11 (1967), 3-29.

[4] Peters, P.S. and R.W. Ritchie, "Context-sensitive immediate constituent analysis--context-free languages revisited," Proc. ACM Symp. on Theory of Computing, May, 1969.

[5] Rabin, M.O. "Mathematical Theory of Automata," Mathematical Aspects of Computer Science (Proc. Symposia Appl. Math., XIX, 173-175) American Mathematical Society, Providence, R.I., 1967.

[6] Rounds, W.C. "Mappings and grammars on trees," submitted to Math. Systems Theory.

[7] Rounds, W.C. "Context-free grammars on trees," Proc. ACM Symp. on Theory of Computing, May, 1969.

[8] Thatcher, J.W., personal communication.

[9] Thatcher, J.W. "Transformations and translations from the point of view of generalized finite automata theory, "Proc. ACM Symp. on Theory of Computing, May, 1969.

[10] Ogden, W.F. "Intercalation theorems for stack automata," Proc. ACM Symp. on Theory of Computing, May, 1969.