Makoto Kanazawa
February 8, 2012

Context-free tree grammars

A *ranked alphabet* is a union $\Delta = \bigcup_{r\in\mathbb{N}} \Delta^{(r)}$ of disjoint sets of symbols. If $f \in \Delta^{(r)}$, r is the *rank* of f.

A tree over a ranked alphabet Δ is a labeled ordered tree where each node with n daughters is labeled by some $f \in \Delta^{(n)}$. It is convenient to use the term representation of trees. The set $\mathbb{T}_\Delta$ of trees over a ranked alphabet Δ is defined inductively as follows:

1. If $f \in \Delta^{(0)}$, then $f \in \mathbb{T}_\Delta$;
2. If $f \in \Delta^{(n)}$ and $t_1, \ldots, t_n \in \mathbb{T}_\Delta$ $(n \geq 1)$, then $(f t_1 \ldots t_n) \in \mathbb{T}_\Delta$.

We usually omit the outermost pair of parentheses.

In what follows, we use a countably infinite supply of variables $x_1, x_2, x_3, \ldots$. The set consisting of the first n variables is denoted X_n (i.e., $X_n = \{x_1, \ldots, x_n\}$). $\mathbb{T}_\Delta(X_n)$ denotes the set of trees over $\Delta \cup X_n$, where members of X_n all have rank 0. A tree in $\mathbb{T}_\Delta(X_n)$ is often written $t[x_1, \ldots, x_n]$, displaying the variables. If $t[x_1, \ldots, x_n] \in \mathbb{T}_\Delta(X_n)$ and $t_1, \ldots, t_n \in \mathbb{T}_\Delta$, then $t[t_1, \ldots, t_n]$ denotes the result of substituting $t_1, \ldots, t_n$ for $x_1, \ldots, x_n$, respectively, in $t[x_1, \ldots, x_n]$. An element $t[x_1, \ldots, x_n]$ of $\mathbb{T}_\Delta(X_n)$ is an *n-context* over Δ if for each $i = 1, \ldots, n$, x_i occurs exactly once in $t[x_1, \ldots, x_n]$. (In the literature, an n-context is sometimes called a *simple tree* in $\mathbb{T}_\Delta(X_n)$.)

A *context-free tree grammar* (Rounds 1970, Engelfriet and Schmidt 1977) is a quadruple $G = (N, \Sigma, P, S)$, where

1. N is a ranked alphabet of nonterminals,
2. Σ is a ranked alphabet of terminals,
3. S is a nonterminal of rank 0, and
4. P is a finite set of productions of the form

$$A x_1 \ldots x_n \to t[x_1, \ldots, x_n],$$

where $A \in N^{(n)}$ and $t[x_1, \ldots, x_n] \in \mathbb{T}_{N\cup\Sigma}(X_n)$.

For every $u, v \in \mathbb{T}_{N\cup\Sigma}$, $u \Rightarrow_G v$ is defined to hold if and only if there is a 1-context $c[x_1]$ over $N \cup \Sigma$, a production $A x_1 \ldots x_n \to t[x_1, \ldots, x_n]$ in P, and trees $u_1, \ldots, u_n \in \mathbb{T}_{N\cup\Sigma}$ such that

$$u = c[A u_1 \ldots u_n]$$

$v = c[t[u_1, \ldots, u_n]]$.

The relation $\Rightarrow_G^*$ on $\mathbb{T}_{N \cup \Sigma}$ is defined as the reflexive transitive closure of $\Rightarrow_G$. The *tree langauge* generated by a context-free tree grammar G, denoted by $L(G)$, is defined as follows:

$$L(G) = \{\, t \in \mathbb{T}_\Sigma \mid S \Rightarrow_G^* t \,\}.$$

The *string language* generated by G is

$$yL(G) = \{\, \mathrm{yield}(t) \mid t \in L(G) \,\}.$$

Consider a context-free tree grammar $G_1 = (N, \Sigma, P, S)$, where

$$\begin{aligned}
N^{(0)} &= \{S\}, \\
N^{(1)} &= \{C\}, \\
N^{(k)} &= \varnothing \quad \text{for all } k \geq 2, \\
\Sigma^{(0)} &= \{a\}, \\
\Sigma^{(2)} &= \{b\}, \\
\Sigma^{(k)} &= \varnothing \quad \text{for all } k \notin \{0, 2\},
\end{aligned}$$

and P consists of the following productions:

$$\begin{aligned}
S &\to Ca, \\
Cx_1 &\to x_1, \\
Cx_1 &\to C(bx_1x_1).
\end{aligned}$$

$L(G_1)$ consists of perfect binary trees where each internal node is labeled by b and each leaf is labeled by a, and $yL(G_1) = \{\, a^{2^n} \mid n \geq 0 \,\}$.

Consider $G_2 = (N, \Sigma, P, S)$, where

$$\begin{aligned}
N^{(0)} &= \{S\}, \\
N^{(2)} &= \{F\}, \\
\Sigma^{(0)} &= \{a\}, \\
\Sigma^{(2)} &= \{b\},
\end{aligned}$$

and P consists of the following productions:

$$\begin{aligned}
S &\to Fa(ba(baa)), \\
Fx_1x_2 &\to x_1,
\end{aligned}$$

$$Fx_1x_2 \to F(bx_1x_2)(bx_2(baa)).$$

$yL(G_2) = \{\, a^{n^2} \mid n \geq 1 \,\}$.

Consider $G_3 = (N, \Sigma, P, S)$, where

$$\begin{aligned} N^{(0)} &= \{S, U\}, \\ N^{(1)} &= \{D\}, \\ \Sigma^{(0)} &= \{0\}, \\ \Sigma^{(1)} &= \{s\}, \\ \Sigma^{(2)} &= \{d\}, \end{aligned}$$

and P consists of the following productions:

$$\begin{aligned} S &\to DU, \\ U &\to 0, \\ U &\to sU, \\ Dx_1 &\to dx_1x_1. \end{aligned}$$

$L(G_3) = \{\, d(s^n0)(s^m0) \mid n, m \geq 0 \,\}$, where s^i0 denotes

$$\underbrace{s(\ldots(s}_{i \text{ times}}0)\ldots).$$

Consider a one-step derivation $u \Rightarrow_G v$, where

$$\begin{aligned} u &= c[Au_1 \ldots u_n], \\ v &= c[t[u_1, \ldots, u_n]], \end{aligned}$$

and $Ax_1 \ldots x_n \to t[x_1, \ldots, x_n]$ is a production. This one-step derivation is *IO* (inside-out) and we write $u \Rightarrow_{G,IO} v$ if $u_1, \ldots, u_n \in \mathbb{T}_\Sigma$. On the other hand, if x_1 does not occur inside an argument of any nonterminal in $c[x_1]$, the one-step derivation is *OI* (outside-in) and we write $u \Rightarrow_{G,OI} v$. The *IO tree language* of G is

$$L_{IO}(G) = \{\, t \in \mathbb{T}_\Sigma \mid S \Rightarrow^*_{G,IO} t \,\}$$

and the *IO string language* of G is

$$yL_{IO}(G) = \{\, \text{yield}(t) \mid t \in L_{IO}(G) \,\}.$$

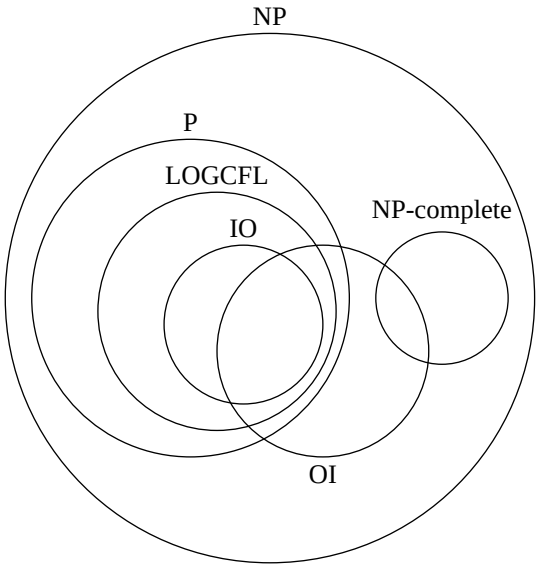
The *OI tree langauge* and *OI string language* of G are defined similarly.

It is known that $L_{OI}(G) = L(G)$ for every context-free tree grammar G. It follows that $L_{IO}(G) \subseteq L_{OI}(G)$ for every G. The inclusion is in general proper (this is so with the above example G_3). The class of OI tree languages and the class of IO tree languages are known to be incomparable.

Define

$$\begin{aligned}
\mathrm{CFT}_{IO} &= \{\, L_{IO}(G) \mid G \text{ is a CFTG} \,\},\\
\mathrm{CFT}_{OI} &= \{\, L_{OI}(G) \mid G \text{ is a CFTG} \,\},\\
\mathrm{yCFT}_{IO} &= \{\, yL_{IO}(G) \mid G \text{ is a CFTG} \,\},\\
\mathrm{yCFT}_{OI} &= \{\, yL_{OI}(G) \mid G \text{ is a CFTG} \,\}.
\end{aligned}$$

yCFT_{IO} (yCFT_{OI}) coincides with the class of languages generated by *macro grammars* (Fisher 1968) in the *inside-out* (*outside-in*) mode. It is known (Fisher 1968) that yCFT_{OI} equals the class of *indexed languages* (Aho 1968), which contains some NP-complete languages (Rounds 1973). yCFT_{IO} is included in LOGCFL (Hunt 1976, Asveld 1981, Engelfriet 1986). The following Venn diagram holds with IO = CFT_{IO}, OI = CFT_{OI} or IO = yCFT_{IO}, OI = yCFT_{OI}:



Exercise 1. Consider the context-free tree grammar $G_4 = (N, \Sigma, P, S)$, where

$$\begin{aligned}
N^{(0)} &= \{S\},\\
N^{(2)} &= \{C, T, F, A\},\\
N^{(3)} &= \{P\},\\
\Sigma^{(0)} &= \{0, \mathbf{true}, \mathbf{false}\},\\
\Sigma^{(1)} &= \{s, \neg\},
\end{aligned}$$

$$\Sigma^{(2)} = \{\wedge, \vee\},$$

and P consists of the following productions:

$$\begin{aligned}
S &\to P\,(s0)\ \textbf{true}\ \textbf{false},\\
Px_1x_2x_3 &\to Cx_2x_3,\\
Px_1x_2x_3 &\to P(sx_1)(Ax_1x_2)x_3,\\
Px_1x_2x_3 &\to P(sx_1)x_2(Ax_1x_3),\\
Cx_1x_2 &\to \wedge(Cx_1x_2)(Cx_1x_2),\\
Cx_1x_2 &\to Tx_1x_2,\\
Tx_1x_2 &\to \vee(Tx_1x_2)(Tx_1x_2),\\
Tx_1x_2 &\to \vee(Tx_1x_2)(Fx_1x_2),\\
Tx_1x_2 &\to \vee(Fx_1x_2)(Tx_1x_2),\\
Fx_1x_2 &\to \vee(Fx_1x_2)(Fx_1x_2),\\
Tx_1x_2 &\to x_1,\\
Tx_1x_2 &\to \neg x_2,\\
Fx_1x_2 &\to \neg x_1,\\
Fx_1x_2 &\to x_2,\\
Ax_1x_2 &\to x_1,\\
Ax_1x_2 &\to x_2.
\end{aligned}$$

We abbreviate $\underbrace{s(\dots(s}_{i \text{ times}}0)\dots)$ by s^i0.

1. Is the following tree in $L_{OI}(G_4)$? Is it in $L_{IO}(G_4)$?

$$\wedge(\vee(s^10)(\vee(\neg(s^20))(s^30)))(\wedge(\vee(\neg(s^10))(\vee(s^20)(s^30)))(\vee(\neg(s^10))(\vee(s^20)(\neg(s^30)))))$$

2. Is the following tree in $L_{OI}(G_4)$?

$$\wedge(\vee(\neg(s^10))(s^20))(\wedge(\vee(s^10)(s^20))(\vee(s^10)(\neg(s^20))))$$

3. Give an example of a tree in $L_{IO}(G_4)$.

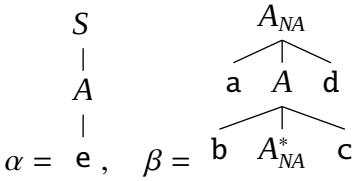
4. Can you describe $L_{OI}(G_4)$?

A production

$$Ax_1 \dots x_n \to t[x_1, \dots, x_n]$$

is *linear* if for each $i = 1, \ldots, n$, x_i occurs in $t[x_1, \ldots, x_n]$ at most once. It is *nondeleting* if for each $i = 1, \ldots, n$, x_i occurs in $t[x_1, \ldots, x_n]$ at least once. A context-free tree grammar is *linear* if all its productions are linear, and it is *nondeleting* if all its productions are nondeleting. A linear and nondeleting CFTG is called *simple*. It is known that if G is linear, $L_{IO}(G) = L_{OI}(G)$ (Kepser and Mönnich 2006). Let $\mathrm{CFT}_{\mathrm{sp}} = \{\, L(G) \mid G \text{ is a simple CFTG} \,\}$. Then, $\mathrm{CFT}_{\mathrm{sp}} \subseteq \mathrm{CFT}_{IO} \cap \mathrm{CFT}_{OI}$.

Consider the tree-adjoining grammar $G = (N, \Sigma, \mathscr{I}, \mathscr{A})$, where $N = \{S, A\}$, $\Sigma = \{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}, \mathsf{e}\}$, $\mathscr{I} = \{\alpha\}$, $\mathscr{A} = \{\beta\}$, and



Define a simple CFTG $G' = (N', \Sigma', P', S')$ as follows:

$$
\begin{aligned}
N'^{(0)} &= \{S'\},\\
N'^{(1)} &= \{A'\},\\
\Sigma'^{(0)} &= \{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}, \mathsf{e}\},\\
\Sigma'^{(1)} &= \{A_1\},\\
\Sigma'^{(3)} &= \{A_3\},\\
P' &= \left\{ \begin{array}{l} S' \to S\,(A'\,(A_1\,\mathsf{e})),\\ A'\,x_1 \to x_1,\\ A'\,x_1 \to A_3\,\mathsf{a}\,(A'\,(A_3\,\mathsf{b}\,(A_1\,x_1)\,\mathsf{c}))\,\mathsf{d} \end{array} \right\}.
\end{aligned}
$$

Then $L(G)$ and $L(G')$ are identical except for the labels A, A_{NA} in the former and A_1, A_3 in the latter.

The class of tree languages of tree-adjoining grammars is included in the class of tree languages generated by *monadic simple context-free tree grammars*. These classes of grammars are equivalent on the level of string languages (Fujiyoshi and Kasai 2000, Mönnich 1997), and almost so on the level of tree languages (Kepser and Rogers 2011).

References

Alfred V. Aho. 1968. Indexed grammars—An extension of context-free grammars. *Journal of the Association for Computing Machinery* **15**, 647–671.

Peter R. Asveld. 1981. Time and space complexity of inside-out macro languages. *International Journal of Computer Mathematics* **10**, 3–14.

Huber Comon, Max Dauchet, Rémi Gilleron, Florent Jacquemard, Denis Lugiez, Sophie Tison, and Marc Tommasi. 2007. *Tree Automata Techniques and Applications*. Available online at `http://tata.gforge.inria.fr/`.

Joost Engelfriet. 1986. The complexity of languages generated by attribute grammars. *SIAM Journal on Computing* **15**, 70–86.

Joost Engelfriet and Erik Meineche Schmidt. 1977. IO and OI. I. *Journal of Computer and System Sciences* **15**, 328–353.

Michael J. Fisher. 1968. *Grammars with Macro-Like Productions*. Ph.D. dissertation. Harvard University.

A. Fujiyoshi and T. Kasai. 2000. Spinal-formed context-free tree grammars. *Theory of Computing Systems* **33**, 59–83.

H. B. Hunt. 1976. On the complexity of finite, pushdown, and stack automata. *Mathematical Systems Theory* **10**, 33–52.

Stephan Kepser and Uwe Mönnich. 2006. Closure properties of linear context-free tree languages with an application to optimality theory. *Theoretical Computer Science* **354**, 82–97.

Uwe Mönnich. 1997. Adjunction as substitution: An algebraic formulation of regular, context-free and tree ajoining languages. In *Proceedings of the Third Conference on Formal Grammar*.

Stephan Kepser and Jim Rogers. 2011. The equivalence of tree adjoining grammars and monadic linear context-free tree grammars. *Journal of Logic, Language and Information* **20**, 361–384.

William C. Rounds. 1970. Mappings and grammars on trees. *Mathematical Systems Theory* **4**, 257–287.

William C. Rounds. 1973. Complexity of recognition in intermediate-level languages. In *14th Annual IEEE Symposium on Switching and Automata Theory*, pages 145–158. IEEE Computer Society.