COLING 82, J. Horecký (ed.)
North-Holland Publishing Company

TREE DIRECTED GRAMMARS

Werner Dilger

Universität Kaiserslautern
Fachbereich Informatik
D-6750 Kaiserslautern
FR Germany

Tree directed grammars as a special kind of translation grammars are defined. It is shown that a loop-free tree directed grammar can be transformed into an equivalent top-down tree transducer, and from this fact it follows that given an arbitrary context-free language as input, a tree directed grammar produces an output language which is at most context-sensitive.

INTRODUCTION

Within the natural language information system PLIDIS [6] a semantic processor was implemented for the translation of syntactically analyzed sentences into expressions of a predicate calculus-oriented internal representation language. This semantic processor was designed according to a translation grammar defined by Wulz [8], which is similar to the transformation grammar introduced by Chomsky [3]. The operations on trees which are defined in the transformation grammar, i.e. deletion, insertion, and transposition of subtrees, are also available in the Wulz grammar. Therefore it can be assumed that it is equivalent to the transformation grammar with regard to the input/output-relation.

But when the Wulz grammar was realized within PLIDIS for a section of German, only of a few of its possibilities was made use. No real transformation was prescribed by the PLIDIS translation rules, they only checked the parse tree and produced an output separated from this tree. Thus, what was realized in the PLIDIS translation rules can be better described by another kind of translation grammar, namely the tree directed grammar (TDG). When we investigate the TDGs and their relation to tree transducers it turns out that they are less powerful than transformation grammars.

TREE DIRECTED GRAMMARS

We define trees in the manner of [2] and [7] as mappings from tree domains (special subsets of N^*, where N is the set of natural numbers) into an alphabet Σ and call them therefore trees "over" Σ. We assume for the rest of the paper that Σ is ranked. Because trees are finite mappings it is convenient to identify a tree with its graph. So e.g. the set

$$\{<(),a>,<(0),b>,<(1),d>,<(2),a>,<(0,0),e>,<(0,1),c>, <(2,0),d>,<(2,1),b>,<(2,2),e>,<(0,1,0),e>,<(2,1,0),c>, <(2,1,1),d>,<(2,1,0,0),d>\}$$

represents the tree of fig. 1.

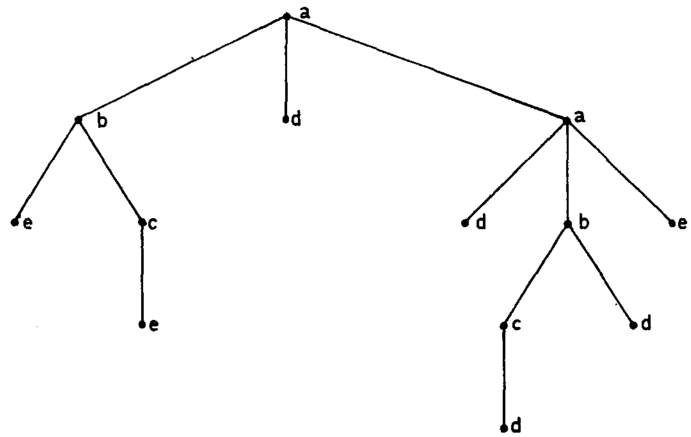

figure 1

If u is an element of a tree domain, $\alpha \in \Sigma$, and $t(u) = \alpha$, then the pair $\langle u,\alpha\rangle$ is called a *node* of t.

Let T be any set of trees over Σ. A *TDG* G_T for T is a quadruple

$$G_T = (\Sigma,\Delta,\Pi,\sigma)$$

where Δ is the alphabet of terminals of G_T, Π is the set of productions of G_T, and $\sigma \in \Sigma \cup \Delta$. It follows from this definition that the elements of Σ play the role of nonterminals in G_T. When they are used for this purpose in the productions, they are enclosed in brackets, so we get from Σ the set

$$[\Sigma] = \{[\alpha] \mid \alpha \in \Sigma\}$$

The elements of Σ are further used in the structural condition parts of the productions. There we should be able to distinguish between different occurrences of the same symbol in a tree. In order to represent such distinctions, the symbols are provided with indices, so we get from Σ the set

$$\Sigma_{IND} = \bigcup_{i \in IND} \{\alpha_i \mid \alpha \in \Sigma\}$$

for some index set IND (in general a subset of N).

Now a *production* $p \in \Pi$ is a triple

$$([\alpha_1],sc,\omega)$$

with $\alpha \in \Sigma$, $\omega \in (\Delta \cup [\Sigma_{IND}])^*$, and sc is a structural condition which contains the symbol α_1.

In order to explain the application of a production we have to define the structural conditions. Assume, $x \notin \Sigma$ and $X = \{x_1,x_2,\ldots\}$. Then the set of *structural individuals* is

$$SI = \Sigma_{IND} \cup X$$

There are four two-place predicates defined on SI, namely DOM ("dominates immediately"), DOM* ("dominates"), LFT ("is immediately left from"), and LFT* ("is left from"). Atomic structural conditions are

$$\text{TRUE, FALSE, } P(\phi,\psi)$$

where P is one of the four predicates above and $\phi,\psi \in SI$. A *structural condition* is then an atomic structural condition or a Boolean expression over the set of atomic structural conditions. For example, if $\Sigma = \{a,b,c,d,e\}$, IND = {1,2}, then the following expressions are structural conditions:

1. $DOM(a_1,b_1)$
2. $DOM(b_1,x_1) \wedge LFT(e_1,x_1)$
3. $DOM(x_1,c_1) \wedge LFT^*(x_1,x_2) \wedge DOM^*(x_2,e_1)$
4. $DOM(a_1,b_1) \wedge (\sim DOM^*(b_1,e_1) \vee LFT(b_1,d_1))$

The semantics of a structural condition is defined in the usual way by an interpreting function from the condition into a semantic domain. Here, the trees of T are semantic domains. The four predicates DOM, ...,LFT* are always interpreted in the same way, and this interpretation should be obvious. The main part of the interpretation is the assignment of the structural individuals to the nodes of a tree, which is called the *node assignment*. A mapping of the individuals of a structural condition into the set of nodes of a tree is a node assignment, if it obeys the following restrictions: If $\alpha \in \Sigma$, then an individual α_i ($i \in$ IND) should be assigned to a node with label α, whereas the individuals α_i and α_j ($i \neq j$) should be assigned to different nodes with the same label α. An individual $x_i \in X$ can be assigned to an arbitrary node. A tree t *satisfies* a structural condition sc if there exists a node assignment such that sc holds for the assigned nodes of t under the assumed interpretation of the four predicates and the usual interpretation of the Boolean operators. The reader is invited to check, how the tree of the example above satisfies the structural conditions 1. - 4.

The structural conditions are similar to the local constraints of Joshi and Levy [5], and it can be shown that both are equivalent with regard to their ability to describe relations on the set of nodes of a tree.

Assume, $p = ([\alpha_1],sc,\omega)$ is a production of G_T. Then the structural individual α_1 must occur in sc. Assume further that

$$y = y_1[\alpha_i]y_2$$

where $y_1,y_2 \in (\Delta \cup [\Sigma_{IND}])^*$, $i \in$ IND, and there is a node assignment which maps α_i on a node $\langle u,\alpha \rangle$ in tree t and t satisfies sc in such a way that α_1 is mapped on $\langle u,\alpha \rangle$ as well, then p can be applied to y:

$$y_1[\alpha_i]y_2 \underset{G_T,t}{\longmapsto} y_1\omega y_2$$

Some of the individuals of X occurring in ω may be replaced by the node assignment for sc by individuals of $[\Sigma_{IND}]$. In this way derivations in G_T with regard to a tree t are defined. If a derivation stops with a word $y \in \Delta^*$, y can be regarded as a translation of t.

Assume e.g. we are given the following four productions:

$$([a_1], DOM(a_1, b_1), [b_1][b_1])$$
$$([b_1], DOM(b_1, x_1) \wedge LFT(e_1, x_1), H[x_1])$$
$$([c_1], DOM(x_1, c_1) \wedge LFT(x_1, x_2) \wedge DOM^*(x_2, e_1), [e_1]E)$$
$$([e_1], TRUE, AR)$$

By means of these productions we can perform the derivation

$$[a_1] \vdash [b_1][b_1] \vdash H[c_1][b_1] \vdash H[e_1]E[b_1] \vdash HARE[b_1] \overset{*}{\vdash} HAREHARE$$

with regard to the tree of the example above.

TOP-DOWN TREE TRANSDUCERS

A *top-down tree transducer* (TDTT) (cf. [4]) is a transducing automaton which proceeds top-down from the root to the leaves in a tree and in each step yields an output. It is defined as a quintuple

$$M = (Q, \Sigma, \Delta, q_o, R)$$

where Σ and Δ are defined as before, Q is a finite set of states, $q_o \in Q$ is the initial state and R is a finite set of rules of the form

$$q(\alpha(\tau_1 \ldots \tau_k)) \longrightarrow y_1 q_1(\tau_{i_1}) y_2 q_2(\tau_{i_2}) \ldots y_n q_n(\tau_{i_n}) y_{n+1}$$

with $n, k \geq 0$; $1 \leq i_j \leq k$ for $1 \leq j \leq n$; $q, q_1, \ldots, q_n \in Q$, $\alpha \in \Sigma$, $y_1, \ldots, y_{n+1} \in \Delta^*$. k is the rank of α and the τ_i are variables over T. When such a rule is applied to a tree t at a node with label α, the variables τ_i are replaced by those subtrees of t whose roots are immediately dominated by the node with label α.

Assume e.g. we are given the TDTT

$$M = (\{q_o, q_1\}, \{a,b,c,d,e\}, \{A,E,H,R\}, q_o, R)$$

with

$$R = \{\ q_o(a(\tau_1\tau_2\tau_3)) \longrightarrow Hq_1(\tau_3)q_o(\tau_1)Hq_1(\tau_3)q_o(\tau_1),$$
$$q_o(b(\tau_1\tau_2)) \longrightarrow q_o(\tau_2),$$
$$q_o(c(\tau_1)) \longrightarrow E,$$
$$q_1(a(\tau_1\tau_2\tau_3)) \longrightarrow q_1(\tau_3),$$
$$q_1(e) \longrightarrow AR\ \}$$

M performs on the tree of the example above the derivation

$$q_o(a(b(ec(e))da(db(c(d)d)e)))$$
$$\vdash Hq_1(a(db(c(d)d)e))q_o(b(ec(e)))Hq_1(a(db(c(d)d)e))q_o(b(ec(e)))$$
$$\vdash Hq_1(e)q_o(b(ec(e)))Hq_1(a(db(c(d)d)e))q_o(b(ec(e)))$$
$$\vdash HARq_o(b(ec(e)))Hq_1(a(db(c(d)d)e))q_o(b(ec(e)))$$
$$\vdash HARq_o(c(e))Hq_1(a(db(c(d)d)e))q_o(b(ec(e)))$$
$$\vdash HAREHq_1(a(db(c(d)d)e))q_o(b(ec(e))) \overset{*}{\vdash} HAREHARE$$

TDGs AND TDTTs

There are some obvious similarities between TDGs and TDTTs. It is easy to see that not every TDTT can be transformed into an equivalent TDG, because the TDTTs have the states as an additional means to direct derivations. In some cases the derivation can be directed by appropriate structural conditions in the same way as it is done by states, but it is easy to construct examples where this is impossible. On the other hand, each TDG can be transformed into an equivalent TDTT. The main step of this transformation is to put together some of the productions so that the resulting productions satisfy the condition that all symbols of the structural condition part except α_1 are situated below the symbol α_1 in each tree, where α_1 corresponds to the first component of the production.

Take e.g. the productions

$$([a_1],DOM(a_1,b_1),[b_1][b_1])$$
$$([b_1],DOM(b_1,x_1) \wedge LFT(e_1,x_1),H[x_1])$$
$$([c_1],DOM(x_1,c_1) \wedge LFT(x_1,x_2) \wedge DOM^*(x_2,e_1),[e_1]E)$$

The first and the second production satisfy the condition, the third one does not, because the nodes assigned to x_1 and x_2 are above that one assigned to c_1 in each tree which satisfies the structural condition. But we can put together the second and the third production and get a new one:

$$([b_1],DOM(b_1,c_1) \wedge LFT(e_1,c_1) \wedge LFT(b_1,x_2) \wedge DOM^*(x_2,e_1), H[e_1]E)$$

Now this production is "better" than the third above, but it does not yet satisfy our condition. Therefore we put it together with the first one and get

$$([a_1],DOM(a_1,b_1) \wedge DOM(b_1,c_1) \wedge LFT(e_1,c_1) \wedge LFT(b_1,x_2) \wedge DOM^*(x_2,e_1),H[e_1]EH[e_1]E)$$

This production is acceptable and together with the production

$$([e_1],TRUE,AR)$$

it performs the same derivation as the four productions above. The productions resulting from this transformation process are all proceeding downward in a tree. Each of them can be transformed into a TDTT of its own and finally these single TDTTs are composed to one TDTT which is equivalent to the TDG.

The transformation process sketched above can be made only if the TDG is loop-free. That means that each node of a tree is passed during a derivation in TDG at most once.

Now we can adopt the result of Baker [1] about top-down tree transductions. It states that the family of the images of recognizable sets of trees (e.g. the set of derivation trees of a context-free grammar) under a top-down transduction is properly contained in the family of deterministic context-sensitive languages. In other words, the result of the translation of the set of derivation trees of a context-free grammar by a TDG is at most a deterministic context-sensitive language.

REFERENCES

[1] Baker, B.S., Generalized Syntax Directed Translation, Tree Transducers, and Linear Space, SIAM J. Comput. 7 (1978) 376 - 391

[2] Brainerd, W.S., Tree Generating Regular Systems, Inf. and Control 14 (1969) 217 - 231

[3] Chomsky, N., Aspects of the Theory of Syntax, MIT Cambridge, Mass., 1965

[4] Engelfriet, J., Rozenberg, G., and Slutzki, G., Tree Transducers, L Systems, and Two-Way-Machines, JCSS 20 (1980) 150 - 202

[5] Joshi, A.K., and Levy, L.S., Constraints on structural descriptions: local transformations, SIAM J. Comput. 6 (1977) 272 - 284

[6] Kolvenbach, M., Lötscher, A., and Lutz, H.-D., (eds.), Künstliche Intelligenz und natürliche Sprache, Forschungsberichte des Instituts für deutsche Sprache 42, Narr-Verlag, Tübingen, 1979

[7] Rosen, B.K., Tree-Manipulating Systems and Church-Rosser Theorems, JACM 20 (1973) 160 - 187

[8] Wulz, H., Formalismen einer Übersetzungsgrammatik, Forschungsberichte des Instituts für deutsche Sprache 46, Narr-Verlag, Tübingen, 1979