RAIRO. INFORMATIQUE THÉORIQUE ET APPLICATIONS

A. RUDNICKI

An economical transformation of graph grammars

RAIRO. Informatique théorique et applications, tome 27, n° 4 (1993), p. 311-325

<http://www.numdam.org/item?id=ITA_1993__27_4_311_0>

Informatique théorique et Applications/Theoretical Informatics and Applications
(vol. 27, n° 4, 1993, p. 311 à 325)

AN ECONOMICAL TRANSFORMATION OF GRAPH GRAMMARS (*)

by A. RUDNICKI (1) (2)

Communicated by W. RYTTER

Abstract. – *The main result is a technical contribution in the theory of so called labeled graphs (l-graphs). We present a more economical transformation of a certain type of monotonic graph grammar G to equivalent context-free graph grammar G'. The best result previously known was quadratic size of the part of G' corresponding to a single production of G. Our main result is an improvement which gives linear size reductions.*

Résumé. – *Le résultat principal est une contribution technique à la théorie des graphes étiquetés. Nous présentons une transformation plus économique d'un certain type de grammaires de graphes monotones G dans des grammaires équivalentes G' libres de contexte. Le meilleur résultat jusque là obtenu était quadratique en la taille de la partie de G' correspondant à une production donnée de G. Notre résultat principal est une amélioration qui donne des réductions linéaires.*

1. INTRODUCTION

Graph grammars have been defined in various ways as generalizations of the usual string grammar. Theory of graph grammars can be applied in the following areas: semantics of recursively defined functions, record handling, data base systems, compiler techniques, development and evolution in biology, pattern synthesis and recognition and many others. Efficiency aspects of graph language recognition require that the corresponding graph grammar be in the simplest possible form. Therefore transformations among various classes of graph grammars are of a great concern. Obviously one should expect that such a transformation does not change the generative power of the grammar.

(*) Received December 1991, accepted December 1992.
(1) Supported by grant KBN 2-1190-91-01.
(2) Institute of Informatics, Warsaw University.

Informatique théorique et Applications/Theoretical Informatics and Applications
0988-3754/93/04 311 15/$3.50/

In this paper we follow [3] and consider the problem of effective transformation of monotonic (M) graph grammar G to an equivalent context-free (CF) graph grammar G'. The main idea deals with a direct proof of $M = CF$ of [3] which has been shown in two steps in [3], namely $M = CS$ and $CS = CF$ (CS, context-sensitive grammar). Our main result is an improvement which gives linear size reductions.

Specifically, if t and s denote the size of the left and right side, respectively, of a production of a given monotonic grammar, then the upper bound on the number of productions in the corresponding context-free grammar has been shown in [3] to be $2(2s+t)s$, while our method gives only $2s+t$ productions.

In the next chapter we recall basic and most general definitions concerning graph grammars. Then we formulate and prove the main result. We provide a representative example of the construction used in the proof.

2. DEFINITIONS

Let us consider two finite sets Σ_V and Σ_E, which we call alphabets of labels of nodes and edges, respectively.

DEFINITION 1: $\{l\text{-graph}\}$

A labeled graph (l-graph) over the alphabets Σ_E and Σ_V is a tuple $d = (K, (\rho_a)_{a \in \Sigma_E}, \beta)$, where:

1. K is a finite set of nodes.
2. for each $a \in \Sigma_E$, $\rho_a \in K \times K$ is a binary relation over K.
3. $\beta : K \to \Sigma_V$ is a labeling function of nodes.

Let α and $\alpha' \in K$. Each pair $(\alpha; \alpha') \in \rho_a$, where $a \in \Sigma_E$, can be viewed as a directed edge from α to α', labeled with symbol "a". Each node k is labeled with $\beta(k) \in \Sigma_V$. By $d(\Sigma_V, \Sigma_E)$ we denote the set of all l-graphs over the alphabets Σ_V and Σ_E, d_ε denotes the empty graph and $d^1(\Sigma_V, \Sigma_E)$ is the class of all single-node graphs from $d(\Sigma_V, \Sigma_E)$.

An example of an l-graph is shown in figure 1. Assume that $\Sigma_V = \{v, m, t, f\}$ and $\Sigma_E = \{i, j, k, l\}$.

DEFINITION 2: $\{subgraph\}$

An l-graph $d' = (K', \rho'_a, \beta') \in d(\Sigma_E, \Sigma_V)$ is an induced subgraph of $d = (K, \rho_a, \beta) \in d(\Sigma_E, \Sigma_V)$ (denoted: $d' \subseteqq d$) if $K' \subseteqq K$ and ρ'_a, β' are ρ, β restricted to the set K'.

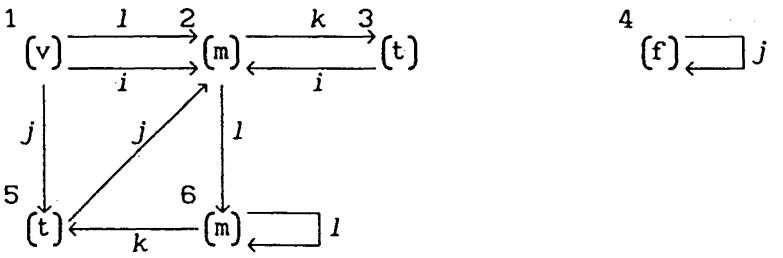


Figure 1

DEFINITION 3: $\{operator\}$

Nonempty strings over an alphabet

$$W=\Sigma_V \cup \{L_a \mid a\in\Sigma_E\} \cup \{R_a \mid a\in\Sigma_E\} \cup \{C, I, \cup, \cap,), (\}$$

are called operators if they are formed according to the following rules:

a) L_a and R_a are operators for each $a\in\Sigma_E$.

b) If A is an operator, then (CA) and $(v\,A)$ are also operators, for each $v\in\Sigma_V^+$.

c) If A and B are operators, then AB, $(A\cup B)$ and $(A\cap B)$ are also operators.

Let $d=(K, \rho_a, \beta)$ and $d'=(K', \rho'_a, \beta')$ be l-graphs over Σ_V, Σ_E and $d'\subseteq d$. For each $x\in K'$ any operator A specifies a subset of node set K, denoted $A(x)$, according to the following recursive definitions:

a) $L_i(x)=\{y\in K-K' : (y; x)\in\rho_i\}$; $R_i(x)=\{y\in K-K' : (x; y)\in\rho_i\}$.

In other words:

$L_i(x)$ specifies the set of source nodes, not contained in K', of i-labeled edges which end at x.

$R_i(x)$ denotes the set of target nodes, not contained in K', of i-labeled edges which start at x.

b) $(CA)(x)=\{y \mid y\in K-K' \text{ and } y\notin A(x)\}$ – the complement of $A(x)$ in $K-K'$.

$(v_1 v_2 \ldots v_m A)(x)=\{y \mid y\in A(x) \text{ and } \beta(y)\in\{v_1, v_2, \ldots, v_m\}\}$, determines the subset of $A(x)$ the nodes of which are labeled with one of the symbol $v_1, v_2, \ldots, v_m$.

c) $I(x)=\{y \mid y\in K-K'\}$ – the set of all nodes outside K'.

d) $AB(x)$ – composition of two operators.

$$(A \cap B)(x) = A(x) \cap B(x)$$
$$(A \cup B)(x) = A(x) \cup B(x)$$

DEFINITION 4: {*graph production*}

Graph production is a triple $p=(d_l, d_r, E)$, where graph d_l is the left side of the production, d_r is the right side, $d_l, d_r \in d(\Sigma_V, \Sigma_E)$ and $E=((l_a, r_a))_{a \in \Sigma_E}$ is an embedding transformation, $l_a = \bigcup_{j=1}^{m} A_j(y_j) \times \{z_j\}$; $r_a = \bigcup_{j=1}^{m} \{z_j\} \times A_j(y_j)$; $m \geqq 1$, $y_j \in K_l$, $z_j \in K_r$, A_j is an operator; K_l, K_r are sets of nodes of d_l and d_r respectively.

Informally, components l_a and r_a assign a-labeled edges which connect d_r and $d-d_l$. $A_j(y_j)$ determines a subset of $K-K_l$ nodes of which are to be adjacent to z_j after the application of the production.

DEFINITION 5: {*direct derivation*}

Graph $d' \in d(\Sigma_V, \Sigma_E)$ is directly derivable from $d \in d(\Sigma_V, \Sigma_E)$ by means of production $p=(d_l, d_r, E)$, which is denoted $d \underset{p}{\rightarrow} d'$, if

a) $d_l \subseteqq d$, $d_r \subseteqq d'$.

b) $d-d_l = d'-d_r$.

c) $\mathrm{In}_a(d_r, d') = l_a$ and $\mathrm{Out}_a(d_r, d') = r_a$, for each $a \in \Sigma_E$, where $\mathrm{In}_a(d_r, d')$ is the set of all a-labeled edges, originating in $d'-d_r$ and terminating in d_r, $\mathrm{Out}_a(d_r, d')$ is the set of all a-labeled edges, originating in d_r and terminating in $d'-d_r$.

Informally the application of production p for graph d modulo renaming of nodes, looks as follows: we find d_l in d, remove it with all its edges and replace it with d_r. Next we link d_r to d with the help of E. Observe that E depends on d_l.

DEFINITION 6: {*graph gramar*}

A labeled graph grammar (*LGG* in short) is a tuple

$$G=(\Sigma_V, \Sigma_E, \Delta_V, \Delta_E, d_0, P, -s\rightarrow),$$

where:

a) Σ_V, Σ_E are nonempty finite alphabets for labeling of nodes and edges.

b) $\Delta_V \subset \Sigma_V$, $\Delta_E \subset \Sigma_E$ are terminal alphabets.

c) $d_0 \in d(\Sigma_V, \Sigma_E) - d(\Delta_V, \Sigma_E) \cup \{d_\varepsilon\}$ is an initial graph.

d) P is a finite set of productions.

d) $-s\rightarrow$ is the transitive closure of the direct derivation described above.

Let D denote the class of all equivalent l-graphs over Σ_V, Σ_E, *i.e.* all structural equivalent graphs d which only have different node labels, then:

DEFINITION 7: {*graph language*}

$L(G) := \{D \mid D \in d(\Delta_V, \Delta_E)$ and $D_0 - s\rightarrow D\}$ is called the language of graph grammar G.

This means that graphs belonging to the language have only terminal nodes and edges, and are derivable from d_0.

DEFINITION 8: {*equivalence*}

Two grammars G_1 and G_2 are called equivalent, denoted by $G_1 \equiv G_2$, if $L(G_1) = L(G_2)$;

DEFINITION 9: {*separable grammar*}

A *LGC* is called separable if for each $p \in P$ either (*a*) or (*b*) take place:

(*a*) $d_l, d_r \in d(\Sigma_V - \Delta_V, \Sigma_E) - \{d_\varepsilon\}$.

(*b*) $d_l \in d^1(\Sigma_V - \Delta_V, \Sigma_E)$, $d_r \in d^1(\Delta_V, \Sigma_E) \cup \{d_\varepsilon\}$.

DEFINITION 10: {*types of productions*}

Graph production $p = (d_l, d_r, E)$ is called:

1. monotonic if $|K_l| \leqq |K_r|$.

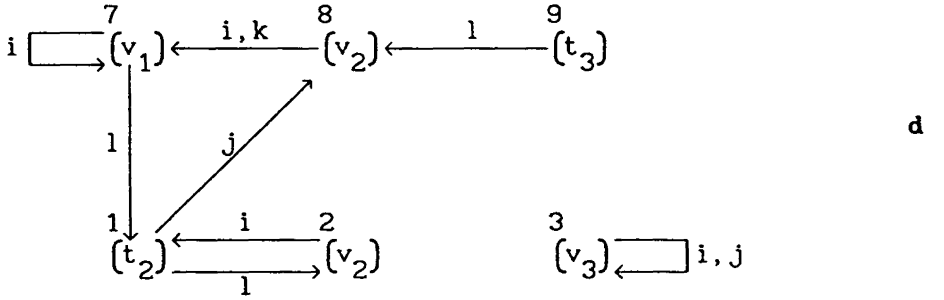
2. context-sensitive if for some $d'_l \subseteq d_l$, $d'_r \subseteq d_r$ where $d'_l \in d^1(\Sigma_V - \Delta_V, \Sigma_E)$ $d'_r \in d(\Sigma_V, \Sigma_E) - \{d_\varepsilon\}$ we have $d_l - d'_l = d_r - d'_r$ and E on the context $d_r - d'_r$ is constant, *i.e.*:

$$\bigcup_{x \in K_l - K'_l} (L_a(x); x) \leqq l_a, \qquad \bigcup_{x \in K_l - K'_l} (x; R_a(x)) \leqq r_a.$$

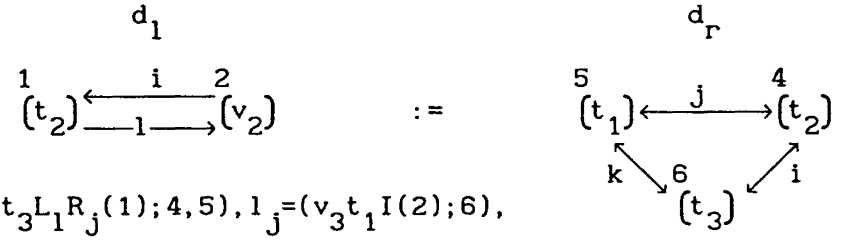
3. Context-free if p is context-sensitive and $|K_l| = 1$.

We shall consider monotonic, context-sensitive and context-free *LGG*, according to the set of productions P.

The following example demonstrates the application of the monotonic production. Consider graph d (taken from [3]):



and monotonic graph production $p=(d_l, d_r, E)$:



E: $l_i=(t_3 L_1 R_j(1); 4,5), l_j=(v_3 t_1 I(2); 6),$

$r_j=(5; R_j \cup L_1 L_1(1,2)), r_k=(4; CR_k R_j(1))$

Figure 2

Replaced d_l by d_r and carefully applied all operators of E we obtain:

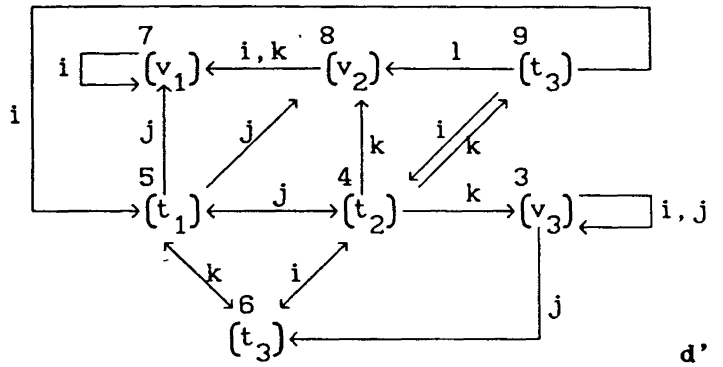


Figure 3

3. TRANSFORMATION OF MONOTONIC *LGG* TO CONTEXT-FREE ONE

Assume that for i-th monotonic production $|K_l| = s_i$ and $|K_r| = t_i$. The structure of the transformation is the following:

Every monotonic production which is not context-free is decomposed into a sequence of context-free productions which, when applied one after the other, give the same result as the monotonic production. Such a sequence consists of three subsequences with the total length of $2s_i + t_i$ context-free productions. First subsequence (part A) identifies a subgraph of d which is isomorphic to d_l (c-edges), assigns new unique labels to its nodes and creates auxiliary $\tilde{a}$-edges. Part A is a preparing part of the transformation. Next part B is the main. With the help of the second subsequence we replace graph isomorphic to d_l by d_r and construct the embedding of d_r into $d' = d - d_l$. Last subsequence of context-free productions (part C) replaces node labels and deletes all unnecessary edges. Derivation steps are controlled by creation blocking f-edges.

THEOREM: *For any monotonic graph grammar G with n productions there exists an equivalent context-free grammar G' with* $\sum_{i=1}^{n} (2s_i + t_i)$ *productions.*

Proof: Let $G = (\Sigma_V, \Sigma_E, \Delta_V, \Delta_E, d_0, P, -s\rightarrow)$ be a monotonic *LGG*. Without loss of generality we can assume that it is separable [3]. Let $P_m \subset P$ be the subset of productions of the grammar G which are monotonic and not context-free. Let:

$$\alpha = \sum_{i=1}^{n} |K_{l_i}|; \qquad \beta = \sum_{i=1}^{n} |K_{r_i}|; \qquad \gamma = \alpha + \beta$$

$$\Pi_{mv} := (N_1, N_2, \ldots, N_\gamma), \quad \Pi'_{mv} = \Pi_{mv} \cup (N'_1, N'_2, \ldots, N'_\gamma), \quad \Sigma_V \cap \Pi'_{mv} = \varnothing.$$

This means that for each monotonic production p_i we introduce $2(|K_{l_i}| + |K_{r_i}|)$ nonterminal labels of nodes.

Denote:

$$\tilde{\Sigma}_E = \{\tilde{a} \mid a \in \Sigma_E\}, \qquad \tau = \max_{i \in P_m} |K_{l_i}|.$$

We put

$$\Sigma'_E = \Sigma_E \cup \tilde{\Sigma}_E \cup c \cup f \cup \{\text{elab}_1, \text{elab}_2, \ldots, \text{elab}_\tau\},$$

thus introducing $|\Sigma_E| + \tau + 2$ new nonterminal edge labels. Similarly, $\Sigma'_V = \Sigma_V \cup \Pi'_{mv}$, *i.e.* 2γ new nonterminal node labels are specified.

I. Let $p=(d_l, d_r, E)\in P_m$ – be a monotonic production with $|K_l|=s$, $|K_r|=t$, clearly $|K_l|\leqq|K_r|$ and, without loss of generality:

$$K_{l_l}=\{1, 2, \ldots, s\} \qquad K_r=\{s+2, s+3, \ldots, s+t+1\}.$$

Next, let $(N_1, N_2, \ldots, N_s)\cup(N'_2, N'_3, \ldots, N'_{s+t+1})$ – be the set of nonterminal labels of nodes, used for the decomposition of this production. Finally, let $K'_l=\{1\}$, $K'_r=\{2s+1, 2s+2, \ldots, s+t+1\}$. The remaining nodes in $d_l-d'_l=\{2, 3, \ldots, s\}$ and $d_r-d'_r=\{s+2, s+3, \ldots, 2s\}$ constitute a fixed context and are paired with the help of the function $h: i\to s+i$, $i=2, \ldots, s$. We will decompose p into a sequence of productions $p^1, p^2, \ldots, p^{2s+t}$.

A: With the help of $p^1, p^2, \ldots, p^s$, which form the first part of the sequence of context-free productions we identify a subgraph of d which is isomorphic to d_l and assign unique labels to its nodes.

*A*1: Production $p^1=(d_l^1, d_r^1, E^1)$ is specified as follows: $d_l^1=\{1\}$, $d_r^1=\{1\}$, labeled N_1, E^1 consists of $E^f_{\mathrm{id}}(1\,|\,1)$ – which means, that all edges are removed except f-edges ("id" – identical). Other E^1 components are the following: $l_{\tilde{a}}=(L_a(1); 1)$ and $r_{\tilde{a}}=(1; R_a(1))$, $a\in\Sigma_E$, which replaces a – by $\tilde{a}$-edge. $r_{\mathrm{elab}_1}=(1; I(1))$, creating elab_1-edges to all nodes of the graph $d-\{1\}$. $l_f=(\Pi'_{mv} I(1); 1)$ which creates f-edges if in the initial graph d there are node labels from Π'_{mv} in d (blocking edges). The existence of such blocking f-edges leads to a blind branch in derivation and, as such edges are preserved, the resulting graph does not belong to the graph language.

*A*2: Let $1<j\leqq s$. Productions $p^j=(d_l^j, d_r^j, E^j)$ are constructed as follows: $d_l^j=\{j\}$, $d_r^j=\{j\}$, labeled N_j E^j consist of $E^{a\in\Sigma_E-(\tilde{\Sigma}_E\cup\Sigma_E)}_{\mathrm{id}}(j\,|\,j)$, complemented by the following components: $l_{\tilde{a}}=(L_a(j); j)$, $r_{\tilde{a}}=(j; R_a(j))$, where $a\in\Sigma_E$. $r_{\mathrm{elab}_j}=(j; I(j))$ which create elab_j-edges incident to all other nodes of the graph d.

$$l_f=(C(L_{\mathrm{elab}_{j-1}}\cup R_{\mathrm{elab}_{j-1}} L_{\mathrm{elab}_{j-1}})(j); j),$$

thus not allowing to omit the application of p^{j-1} before the application of p^j. Furthermore, $l_f=(N_j \ldots N_\gamma I(j); j)$ creates blocking f-edges, in case when labels of nodes $N_j \ldots N_\gamma$ appear in the given graph. This means that a production which comes after the current one or belonging to another sequence has already been applied.

The next component of the embedding transformation of p^j generates c-edges and is defined in general as follows:

Let $1 \leqq i < j$. If $\mu \in \Sigma_E$, $\tilde{\mu} \in \tilde{\Sigma}_E$, $\nu \in \Sigma_E$, $\tilde{\nu} \in \tilde{\Sigma}_E$, $(i; j) \in \rho_{l_\mu}$, $(j; i) \in \rho_{l_\nu}$ then $l_c = (N_i(\bigcap_{\tilde{\mu}} L_{\tilde{\mu}} \cap \bigcap_{\tilde{\nu}} R_{\tilde{\nu}})(j); j)$. In the case $(i; j) \notin \rho_{1_\mu}$, $\bigcap_{\tilde{\mu}} L_{\tilde{\mu}}$ in l_c is replaced by I. Analogously, if $(j; i) \notin \rho_{l_\nu}$, then $\bigcap_{\tilde{\nu}} R_{\tilde{\nu}}$ is replaced by I. Finally, when there are no edges between i and j in d_l, l_c takes the form: $l_c = (N_i I(j); j)$. Hence, an incoming c-edge to node j is created in two cases:

1. If there is an edge connection structure between i and j in d, which corresponds to the analogous one in d_l.

2. If in d_l and d there is no edge between i and j.

In this way we check whether d contains all the edges between j and $1, \ldots, j-1$. The last component checks for superfluous edges in d, in comparison to d_l:

If $\mu, \nu \in \Sigma_E$, $\tilde{\mu}, \tilde{\nu} \in \tilde{\Sigma}_E$, $(i; j) \in \rho_{l_\mu}$, $(j; i) \in \rho_{l_\mu}$, then

$$l_f = (N_i(L_{\bar{\mu}} \cup R_{\bar{\nu}})(j); j), \qquad \text{where} \qquad L_{\bar{\mu}} = \bigcup_{\tilde{a} \neq \tilde{\mu}} L_{\tilde{a}}, \qquad R_{\bar{\nu}} = \bigcup_{\tilde{a} \neq \tilde{\nu}} R_{\tilde{a}}$$

If:

$$(i, j) \notin \rho_{l_\mu}, \quad \forall \mu \in \Sigma_E \qquad \text{then} \quad L_{\bar{\mu}} = \bigcup_{\tilde{a} \in \tilde{\Sigma}_E} L_{\tilde{a}}$$

$$(j; i) \notin \rho_{l_\nu}, \quad \forall \nu \in \Sigma_E \qquad \text{then} \quad R_{\bar{\nu}} = \bigcup_{\tilde{a} \in \tilde{\Sigma}_E} R_{\tilde{a}}$$

This component creates blocking f-edges if there are edges between i and j in d which do not occur in d_l.

B: The subsequence of context-free productions $p^{s+1}, p^{s+2}, \ldots, p^{2s}$ causes that the graph isomorphic to d_l is replaced by d_r, labeled with N'_{j+1}, for $1 \leqq j \leqq t$. Furthermore, we construct the embedding of d_r into $d' = d - d_l$. First we will embed remaining part d'_r:

*B*1: Production $p^{s+1} = (d_l^{s+1}, d_r^{s+1}, E^{s+1})$ is constructed as follows: $d_l^{s+1} = \{1\}$, labeled N_1

$d_r^{s+1} = d_r\{2s+1, \ldots, s+t+1\}$,

labeled $(N'_{s+1}, \ldots, N'_{t+1})$, *i.e.* d'_r. E^{s+1} consists of:

$$l_\mu = (N_j I(1); q) \qquad \text{if} \quad \exists q, j \mid q \in \{2s+1, \ldots, s+t+1\},$$
$$2 \leqq j \leqq s \qquad \text{with} \quad (s+j; q) \in \rho_{r_\mu} \quad \mu \in \Sigma_E$$

$$r_\mu=(q;\ N_j I(1)) \qquad \text{if} \quad \exists\, q, j \,|\, q\in\{2s+1,\ \ldots,\ s+t+1\},$$
$$2\leqq j\leqq s \qquad \text{with} \quad (q;\ s+j)\in \rho_{r_\mu}$$

These components connect d'_r and $d_r-d'_r$. It remains to embed d'_r into d. Consider $(A(j);\ k)=l_a\in E_i$ of p_i labeled with $a\in\Sigma_E$, $1<j\leqq s$, $k\in K'_r$. Observe that:

1. Embedding of d'_r may depend on embedding of nodes $d_r-d'_r$.

2. Operators may assign nodes from d_r which, for a context-free production, are adjacent to the given node.

These problems can be omited if we replace this component by $l_a=(\tilde{A}(N_j R_{\text{elab}_1})(1);\ k)$. Operator $\tilde{A}$ determines nodes from $d-d_l$ using complement of Π_{mv} (*i.e.* Σ_V) and edges $\tilde{a}$. Analogously for $r_a:(k;\ A(j))=r_a$ is replaced by $r_a=(k;\ \tilde{A}(N_j\ R_{\text{elab}_1})(1))$.

All components of the type $(A(1);\ k)=l_a$ with $k\in K'_r$ are replaced by $l_a=(\tilde{A}(1);\ k)$. Similarly, for r_a we obtain $r_a=(k;\ \tilde{A}(1))$.

E^{s+1} also contains $E_{\text{id}}^{a\in\tilde{\Sigma}_E\cup f\cup c}(1|2s+1)$, and is complemented by the following components:

$$l_f=(C(L_{\text{elab}_s}\cup R_{\text{elab}_s}L_{\text{elab}_s})(1);\ 2s+1)$$

– creates an f-edge if production p^{2s} has been applied before.

$$l_f=(N_2\ldots N_s(CR_c)(1);\ 2s+1)$$

– creates blocking f-edges in the case when at least one node of $N_2,\ \ldots,\ N_s$ is not connected with $\{1\}$ with a c-edge.

*B*2: Let $1<j\leqq s$. Productions $p^{s+j}=(d_l^{s+j},\ d_r^{s+j},\ E^{s+j})$ look as follows:

$$d_l^{s+j}=\{j\},\ \text{labeled}\ N_j$$
$$d_r^{s+j}=\{s+j\},\ \text{labeled}\ N'_j$$

E^{s+j} consists of $E_{\text{id}}^{a\in\tilde{\Sigma}_E\cup f\cup c}(j|s+j)$, and the following components: $l_f=(N'_{j+1}\ldots N'_s(CR_c)(j);\ s+j)$ – creates blocking f-edges, if there is no c-edge between i and j $(j<i\leqq s)$, *i.e.* the edge which connects these nodes has no counterpart in d_l.

$l_f=(N_{j-1}I(j);\ s+j)$ – forces the application of production p^{s+j-1} earlier.

Let $1<z<j$

$$l_\mu=(N_z I(j);\ j) \quad \text{if in}\ d_r\ \exists\,(s+j;\ s+z)\in\rho_{r_\mu}, \qquad \text{where} \quad 2\leqq j\leqq s$$
$$r_\mu=(j;\ N_z I(j)) \quad \text{if in}\ d_r\ \exists\,(s+z;\ s+j)\in\rho_{r_\mu}, \qquad \text{where} \quad 2\leqq j\leqq s$$

These components correlate $d_l - d'_l$ with $d_r - d'_r$. Then it is necessary to embed $d_r - d'_r$ into the host graph. For this operation the same remarks as in point $B1$ should be made.

Let $(A(z); s+j) = l_a \in E_i$ of p_i, $1 < j \leqq s$, $z \leqq s (z \neq j)$. This component is replaced by $l_a = (\tilde{A}(\bar{N}_z I(j)); s+j)$, where $a \in \Sigma_E$ and $\bar{N}_z \mp \begin{cases} N'_z, & 1 < z < j \\ N_z, & j < z \leqq s \\ N'_{s+1}, & z = 1 \end{cases}$

Where operator $\tilde{A}$ determines nodes from $d - d_l$ using complement of Π'_{mv} (*i.e.* Σ_V) and edges $\tilde{a}$. Analogously for components r_a we obtain:

$$r_a = (s+j;\ \tilde{A}(\bar{N}_z I(j))).$$

All components of the type $(A(j); s+j) = l_a$ are replaced by $l_a = (\tilde{A}(j); s+j)$ and for r_a we receive: $r_a = (s+j;\ \tilde{A}(j))$.

C: The last subsequence of productions $p^{2s+1}, p^{2s+2}, \ldots, p^{2s+t}$ replaces nodes in d' labeled N'_{j+1} $1 \leqq j \leqq t$, by the same nodes with labels from d_r. Here also all unnecessary edges which were applied in the construction are deleted.

Productions $p^{2s+j} = (d_l^{2s+j}, d_r^{2s+j}, E^{2s+j})$ where $1 \leqq j \leqq t$ are defined as follows:

$$d_l^{2s+j} = \{s+j+1\}, \text{ labeled } N'_{j+1}$$

$$d_r^{2s+j} = \{s+j+1\}, \text{ labeled } \beta\{s+j+1\}$$

E^{2s+j} consists of $E_{\mathrm{id}}^{a \in \Sigma_E \cup f}$.

II. The decomposition into the sequence of productions is performed for all $p \in P_m \subset P$. Let P_1 be the set of all productions received in the above way. Left and right sides (d_l and d_r) or productions from $P - P_m$ are constant and embedding transformations E are modified in such a way that they preserve blocking edges. Let P_2 be the set of all productions obtained from $P - P_m$ by this modification. Then if $P' = P_1 \cup P_2$ we have $G' = (\Sigma'_V, \Sigma'_E, \Delta_V, \Delta_E, d_0, P', -s\rightarrow) \equiv G$. This is because:

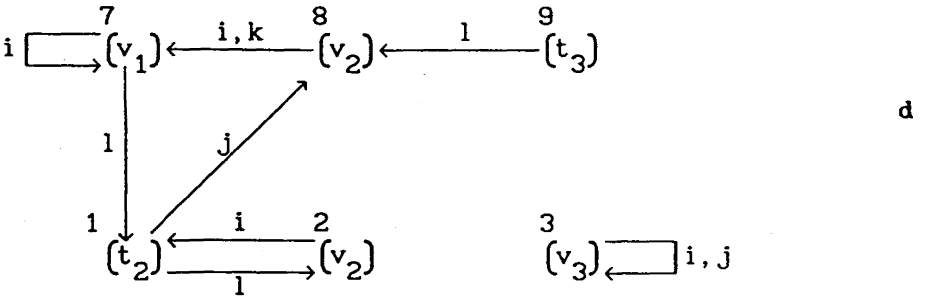
a) For each derivation in G we obtain the corresponding derivation in G', because each step for $p \in P_m$ is replaced by a sequence of steps of the derivation with productions from I, and each step for $p \in P - P_m$ is replaced by a step with a modified production from P_2.

b) For the opposite, assume that the derivation of the graph $D \in L(G')$ is defined. This means that graph D does not contain f-labeled blocking edges. Therefore the sequence of productions which are related to the production in G has been applied according to the order defined in I. It is easy to see

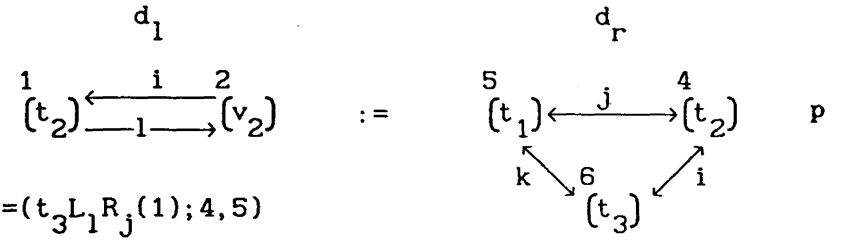
that this sequence does not intersect itself and the others, because embedding transformations E are modified in the sense, that they preserve blocking edges and D does not have such edges. We obtain the same derivation in G by replacing the part of derivation sequence by corresponding derivation in G, *i.e.* $G' \equiv G$. ■

4. EXAMPLE

The following example demonstrates the technique of decomposing a monotonic production into a sequence of $2s+t$ context-free productions. Consider graph d (Taken from page 316):



and monotonic graph production $p=(d_l, d_r, E) \in P_m$:



E: $l_i=(t_3 L_1 R_j(1); 4, 5)$
$l_j=(v_3 t_1 I(2); 6)$
$r_j=(5; R_j \cup L_1 L_1(1, 2))$
$r_k=(4; CR_k R_j(1))$

Figure 4

According to the algorithm of the proof denote: $(s=2,\ t=3)$

$$K_l=\{1, 2\}, \qquad K_r=\{4, 5, 6\}, \qquad K_l'=\{1\}, \qquad K_r'=\{5, 6\}$$

$$h: 2 \to 4, \qquad d_l - d'_l = \{2\}, \qquad d_r - d'_r = \{4\}.$$

We are decomposing p into a sequence $p^1, p^2, \ldots, p^7$ of context-free productions.

A1. $p^1 = (d_l^1 . d_r^1, E^1)$, $d_l^1 = \{1\}$, labeled t_2; $d_r^1 = \{1\}$, labeled N_1, E^1 consist of: $E_{\mathrm{id}}^f(1 \mid 1)$ complemented by:

$$l_{\tilde{i}} = (L_i(1);\ 1), \qquad r_{\tilde{l}} = (1;\ R_l(1)), \qquad l_{\tilde{l}} = (L_l(1);\ 1),$$
$$r_{\tilde{j}} = (1;\ R_j(1)), \qquad r_{\mathrm{elab}_1} = (1;\ I(1)), \qquad l_f = (\Pi'_{mv} I(1);\ 1)$$

A2. $p^2 = (d_l^2, d_r^2, E^2)$, $d_l^2 = \{2\}$, labeled v_2; $d_r^2 = \{2\}$, labeled N_2, $E^2 : E_{\mathrm{id}}^{a \in \Sigma'_E - (\tilde{\Sigma}_E \cup \Sigma_E)}(2 \mid 2)$, thus preserving c-, f-, elab_1- and elab_2-edges, complemented by:

$$l_{\tilde{l}} = (L_l(2);\ 2), \qquad r_{\tilde{i}}(2;\ R_i(2)), \qquad r_{\mathrm{elab}_2} = (2;\ I(2)),$$
$$l_f = (C(L_{\mathrm{elab}_1} \cup R_{\mathrm{elab}_1} L_{\mathrm{elab}_1})(2);\ 2), \qquad l_f = (N_2 \ldots N_\gamma I(2);\ 2),$$
$$l_c = (N_1(L_{\tilde{l}} \cap R_{\tilde{i}})(2);\ 2), \qquad l_f = (N_1(L_{\tilde{l}} \cup R_{\tilde{i}})(2);\ 2).$$

Now we have:

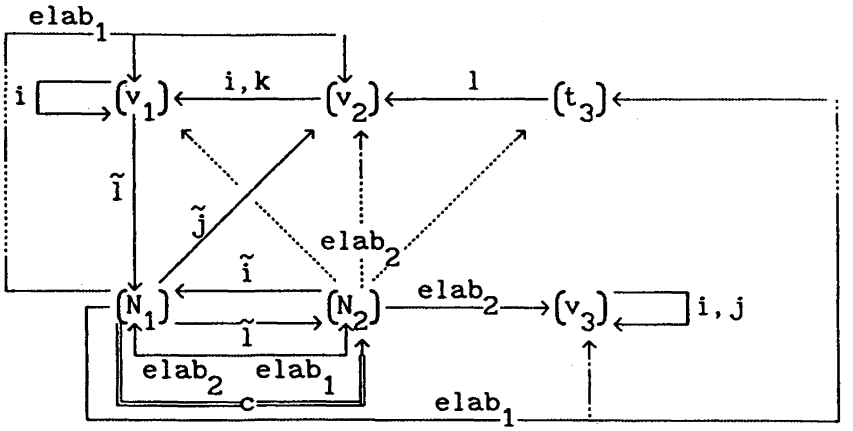


Figure 5

B1. $p^3 = (d_l^3, d_r^3, E^3)$, $d_l^3 = \{1\}$, labeled N_1; $d_r^3 = \{5, 6\}$, labeled N'_3, N'_4 $E^3 : E_{\mathrm{id}}^{a \in \tilde{\Sigma}_E \cup f \cup c}(1 \mid 5)$, and also:

$$l_j = (N_2 I(1);\ 5), \qquad l_i = (N_2 I(1);\ 6), \qquad r_j = (5;\ N_2 I(1)), \qquad r_i = (6;\ N_2 I(1)),$$
$$l_j = (v_3 t_1 I(N_2 R_{\mathrm{elab}_1})(1);\ 6), \qquad r_j = (5;\ R_{\tilde{j}} \cup L_{\tilde{l}} L_{\tilde{l}} (N_2 R_{\mathrm{elab}_1}(1))),$$

$$l_i = (t_3 L_l R_{\tilde{j}}(1);\ 5), \qquad r_j = (5;\ R_{\tilde{j}} \cup L_l L_{\tilde{l}}(1)),$$

$$l_f = (C(L_{\text{elab}_2} \cup R_{\text{elab}_2} L_{\text{elab}_2})(1);\ 5), \qquad l_f = (N_2 (CR_c)(1);\ 5).$$

B2. $p^4 = (d_l^4,\ d_r^4,\ E^4)$, $d_l^4 = \{2\}$, labeled N_2; $d_r^4 = \{4\}$, labeled N_2'.
$E^4 : E_{\text{id}}^{a \in \tilde{\Sigma}_E \cup f \cup c}(2 \mid 4)$, and also: $l_f = (N_1 I(2);\ 4)$,

$$l_i = (t_3 L_l R_{\tilde{j}}(N_3' I(2));\ 4), \qquad r_k = (4;\ v_1 v_2 v_3 t_1 t_2 t_3 CR_k R_{\tilde{j}}(N_3' I(2))).$$

Now looks at the intermediate graph:

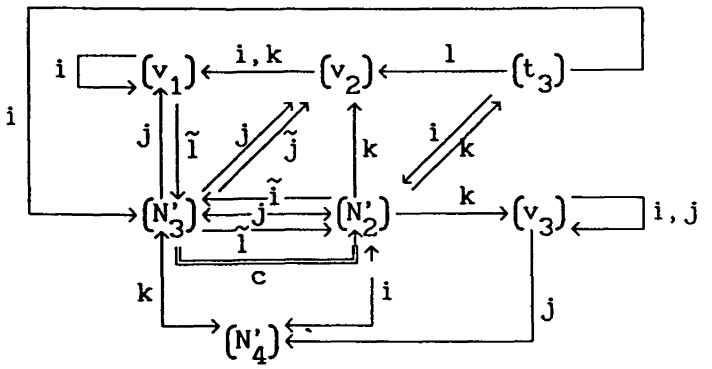


Figure 6

C. $p^5 = (d_l^5,\ d_r^5,\ E^5)$, $d_l^5 = \{4\}$, labeled N_2'; $d_r^5 = \{4\}$, labeled t_2

$$E^5 : E_{\text{id}}^{a \in \Sigma_E \cup f}(4 \mid 4).$$

$p^6 = (d_l^6,\ d_r^6,\ E^6)$, $d_l^6 = \{5\}$, labeled N_3'; $d_r^6 = \{5\}$, labeled t_1

$$E^6 : E_{\text{id}}^{a \in \Sigma_E \cup f}(5 \mid 5).$$

$p^7 = (d_l^7,\ d_r^7,\ E^7)$, $d_l^7 = \{6\}$, labeled N_4'; $d_r^7 = \{6\}$, labeled t_3

$$E^7 : E_{\text{id}}^{a \in \Sigma_E \cup f}(6 \mid 6).$$

It is clear that if we take nodes distinct from $\{1\}$ and $\{2\}$ as d_l in d then in the process of decomposition we obtain blocking f-edges. Branches of derivation tree which correspond to these cases are blind. Continuing derivation along such a branch we shall never obtain a graph labeled only by terminal labels. Only the branch on which d_l consists of $\{1\}$ and $\{2\}$ does not lead to the creation of f-edges.

Now, using all 7 context-free productions one after the other, we get the following result (graph d'):

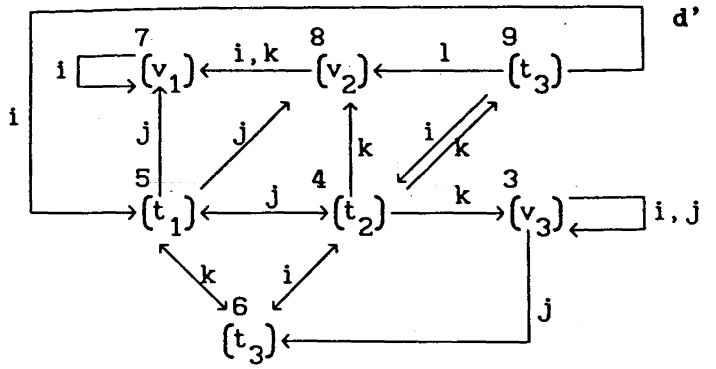


Figure 7

Resulting graph d' is identical to the graph received as the result of applying initial monotonic production (see *fig.* 3 on page 316).

REFERENCES

1. E. Grotsch and M. Nagl, *Explicit Versus Implicit Parallel Rewriting on Graphs*, LNCS Springer-Verlag, *N73*, 1979, pp. 237-254.
2. L. Levy and Yueh Kang, On Labeled Graph Grammars, *Computing*, *20*, 1978, pp. 109-125.
3. M. Nagl, *Graph-Grammatiken Theorie, Anwendungen, Implementierung*, Wiesbaden, 1979.
4. M. Nagl, A tutorial and bibliographical survey on graph grammars, *LCNS*, Springer-Verlag, *N73*, 1979, pp. 70-126.
5. M. Nagl, Set Theoretic Approaches to Graph Grammars. *NLCS*, Springer-Verlag, *N291*, 1987, pp. 41-54.