ON THE MEMBERSHIP PROBLEM FOR REGULAR
DNLC GRAMMARS

by

IJ.J. Aalbersberg*, A. Ehrenfeucht** and G. Rozenberg***

CU-CS-297-85 April, 1985

*Institute of Applied Mathematics and Computer Science,
University of Leiden, Leiden, The Netherlands.

**University of Colorado, Department of Computer Science,
Boulder, Colorado

***Institute of Applied Mathematics and Computer Science,
University of Leiden, Leiden, The Netherlands and University
of Colorado, Department of Computer Science, Boulder,
Colorado
ANY OPINIONS, FINDINGS, AND CONCLUSIONS OR RECOMMENDATIONS EXPRESSED IN THIS PUBLICATION ARE THOSE OF THE AUTHOR AND DO NOT NECESSARILY REFLECT THE VIEWS OF THE NATIONAL SCIENCE FOUNDATION.
ON THE MEMBERSHIP PROBLEM FOR REGULAR DNLC GRAMMARS

by

IJ.J. Aalbersberg*
A. Ehrenfeucht**
G. Rozenberg*

* Institute of Applied Mathematics and Computer Science
  University of Leiden
  Wassenaarseweg 80, 2333 AL LEIDEN
  The Netherlands

** Department of Computer Science
University of Colorado at Boulder
Boulder, Colorado 80309
U.S.A.

All correspondence to the third author
ABSTRACT

There are (at least) three motivations to study the class of regular directed node-label controlled graph grammars (regular DNLC grammars for short): (1) it fits very well into the hierarchy of subclasses of DNLC grammars, (2) it generalizes naturally right-linear string grammars and (3) it provides a useful framework for the theory of concurrent systems based on the theory of traces.

The complexity of (the membership problem for) the class of regular DNLC grammars is investigated.

INTRODUCTION

The theory of graph grammars is a natural extension of the formal string language theory and it has become a well-established topic of research (see, e.g., Ehrig et al., 1983). The potential applicability of graph grammars in many fields of computer science provides substantial motivation for theoretical studies.

One of such fields is the theory of traces, initiated by Mazurkiewicz (1977), which has become quite popular as an approach to the theory of concurrent systems (see, e.g., Mazurkiewicz, 1984a, Mazurkiewicz 1984b, Bertoni et al., 1981, and Aalbersberg and Rozenberg, 1984b). In Aalbersberg and Rozenberg (1984a) the theory of traces was related to the theory of graph grammars through regular directed node-label controlled graph grammars (abbreviated regular DNLC grammars) - a subclass of the directed node-label controlled graph grammars (see, e.g., Janssens and Rozenberg, 1981). It also turns out that the class of regular DNLC grammars fits very well into the hierarchy of various subclasses of DNLC grammars - e.g., it is a very natural subclass of the (directed version of) boundary NLC grammars (see, Rozenberg and Welzl, 1984). Moreover the notion of a regular DNLC grammar transfers very nicely the notion of a right-linear grammar into the framework of graph grammars.

In this note we investigate the complexity of regular DNLC grammars and in particular we prove that the membership problem for regular DNLC grammars is NP-complete.

0. PRELIMINARIES

We assume the reader to be familiar with both, basic formal string language theory (see, e.g., Salomaa, 1973) and the theory of NP-Completeness (see, e.g., Garey and Johnson, 1979).

For sets $A$ and $B$, $A - B$ denotes their difference; $\emptyset$ denotes the empty set. For a set $A$, $\#A$ denotes the cardinality of $A$.

A directed node labeled graph, in the sequel simply called a graph, will be
specified in the form \( \gamma = (V, E, \Sigma, \ell) \) where \( V \) is its set of nodes, 
\( E \subseteq (V \times V) - \{(v, v) \mid v \in V\} \) is its set of edges, \( \Sigma \) is its label alphabet and 
\( \ell : V \rightarrow \Sigma \) is its (node) labeling function. If graphs \( \gamma \) and \( \gamma' \) are isomorphic (and we consider only the node-label preserving isomorphisms), then we write 
\( \gamma \cong \gamma' \). We will sometimes identify isomorphic graphs as equal (then we really consider the so-called abstract graphs). For an alphabet \( \Sigma \), \( G_{\Sigma} \) denotes the set of all graphs with the label alphabet \( \Sigma \).

For a graph \( \gamma \), \( |\gamma| \) denotes the number of nodes of \( \gamma \). For a symbol \( c \), a graph 
\( \gamma = (V, E, \Sigma, \ell) \) such that \( c \in \Sigma \), \( V = \{v_1, \ldots, v_n\} \) for some \( n \geq 1 \), \( \ell(v_i) = c \) for 
every \( 1 \leq i \leq n \), and \( E = \{(v_i, v_{i+1}) \mid 1 \leq i \leq n-1\} \) is called a \( c \)-path. A graph 
\( \gamma \) is a \( k \)-element \( c \)-path, where \( k \geq 1 \), if \( \gamma \) is a \( c \)-path and \( |\gamma| = k \). For graphs 
\( \gamma = (V, E, \Sigma, \ell) \) and \( \gamma' = (V', E', \Sigma', \ell') \) where \( V \cap V' = \emptyset \), their union is the graph 
\( (V \cup V', E \cup E', \Sigma \cup \Sigma', \ell \cup \ell') \). (Hence we consider unions of disjoint graphs only, or, if we consider abstract graphs, then disjoint representatives are chosen).

We recall now basic notions concerning regular DNLC grammars.

Definition 1. (1) A directed node-label controlled graph grammar, abbreviated 
DNLC grammar, is a system \( G = (\Gamma, \Delta, P, C_{\text{in}}, C_{\text{out}}, Z) \), where: (i) \( \Gamma \) is an alphabet, 
called the total alphabet of \( G \), (ii) \( \Delta \subseteq \Gamma \) is called the terminal alphabet of \( G \),
(iii) \( P \subseteq (\Gamma - \Delta) \times G \) is called the set of productions of \( G \), (iv) \( C_{\text{in}} \subseteq \Gamma \times \Gamma \) is called the \( \text{in-connection relation of} \ G \) and \( C_{\text{out}} \subseteq \Gamma \times \Gamma \) is called the \( \text{out-connection relation of} \ G \), and (v) \( Z \), called the axiom of \( G \), is a graph over \( \Gamma \) consisting of one node labeled by an element of \( \Gamma - \Delta \).
(2) A DNLC grammar \( G = (\Gamma, \Delta, P, C_{\text{in}}, C_{\text{out}}, Z) \) is called regular, if every production
\( a \gamma \) of \( G \) is either of the form \( (X, \cdots \cdots) \) or of the form \( (X, \mathbf{a}) \), with \( a \in \Delta \) and 
\( \gamma \in \Gamma - \Delta \). □

Informally speaking, a DNLC grammar \( G = (\Gamma, \Delta, P, C_{\text{in}}, C_{\text{out}}, Z) \) generates a set 
of graphs as follows. Given a graph \( \gamma \) to be rewritten and a production of the 
form \( (X, \beta) \), where \( X \in \Gamma - \Delta \) and \( \beta \in G_{\Gamma} \), one chooses a node \( v \) of \( \gamma \) labeled by \( X \) 
and replaces it by (a graph isomorphic to) \( \beta \). Then, in order to embed \( \beta \) in "the 
remainder of \( \gamma \)" (i.e., the graph resulting from \( \gamma \) by removing \( v \)), one uses rela-
tions \( C_{\text{in}} \) and \( C_{\text{out}} \) as follows. For every pair \( (b, c) \in C_{\text{in}} \), one establishes an 
(incoming) edge from each direct neighbour node of \( v \) labeled \( c \) to each node of \( \beta \)
labeled \( b \). Analogously, for every pair \( (b, c) \in C_{\text{out}} \), one establishes an (outgoing) 
edge from each node labeled \( b \) in \( \beta \) to each direct neighbour node of \( v \) labeled \( c \).
Every graph \( \gamma' \) isomorphic to the resulting graph is said to be directly derived 
from \( \gamma \) in \( G \). Iterating the direct derivation step (starting with the axiom graph
Z of G) and choosing only those derived graphs that are labeled by labels from the terminal alphabet Δ one gets the (graph) language $L(G)$ of G.

These notions are defined formally in Janssens and Rozenberg (1981).

1. THE COMPLEXITY OF REGULAR DNLC GRAMMARS

We consider now the complexity of the membership problem for regular DNLC grammars.

**Theorem 2.1.** There exists a regular DNLC grammar G (with the empty out-connection relation) such that the membership problem for $L(G)$ is NP-complete.

**Proof.** Let $G = (Γ, Δ, P, C_{in}, C_{out}, Z)$ be the regular DNLC grammar, such that:

(i) $Γ = \{A_0, A_1, B_0, B_1, a, b\}$,

(ii) $Δ = \{a, b\}$,

(iii) $P = \{(B_0, \overset{a}{\bullet\rightarrow A_0}), (B_1, \overset{b}{\bullet\rightarrow A_0}), (B_1, \overset{b}{\bullet\rightarrow A_1}), (A_0, \overset{a}{\bullet\rightarrow B_0}), (A_0, \overset{a}{\bullet\rightarrow B_1}), (A_0, \overset{a}{\bullet\rightarrow A_0}), (A_1, \overset{a}{\bullet\rightarrow B_1}), (A_1, \overset{a}{\bullet\rightarrow A_1})\}$,

(iv) $C_{in} = \{(a, a), (b, b), (B_1, b), (A_1, a)\}$,

(v) $C_{out} = \emptyset$, and

(vi) Z is a graph consisting of one node labeled by $B_0$.

Informally speaking G works as follows. G generates sets of b-paths and a-paths by generating alternatively a b-labeled node and an a-labeled node ("b-generating" nonterminals $B_0$ and $B_1$ always introduce one of the "a-generating" nonterminals $A_0$ and $A_1$). In this way in each graph of $L(G)$ the number of b-labeled nodes equals the number of a-labeled nodes. Moreover, at any moment G can "decide" to break the paths it is generating, by introducing one of the "disconnecting" nonterminals $A_0$ and $B_0$. However, after breaking a b-path one has to break the "associated" a-path: $(B_0, \overset{b}{\bullet\rightarrow A_0})$ is the only production for the "breaking" nonterminal $B_0$.

A "typical" graph in $L(G)$ looks as follows:

![fig. 1](image-url)

The following result follows directly from the construction of G.

**Lemma 1.** Let $Γ$ be a graph which is the non-empty union of a-paths and b-paths. Let, for $m ≥ 0$, $S_b = \{β_1, \ldots, β_m\}$ be the set of disjoint b-paths of $Γ$ and let $S_a$ be the collection of a-paths of $Γ$. Then $Γ \subset L(G)$ if and only if there exists a partition $\{S_{a_1}, \ldots, S_{a_n}\}$ of $S_a$, such that, for every $1 ≤ i ≤ m$, $\#β_i = \sum_δ \delta S_{a_i}$. ■
Lemma 2. The membership problem for \( L(G) \) is NP-complete.

Proof. Obviously the membership problem for \( L(G) \) is in NP.

In order to show that the membership problem for \( L(G) \) is NP-hard, consider the following NP-complete problem (see Garey & Johnson, 1979, Problem SP15, page 224).

**3 - PARTITION**

**Instance:** A finite set \( S \) of \( 3n \) elements, where \( n \geq 1 \), a positive integer \( k \) and, for every \( s \in S \), a positive integer \( v(s) \), such that, for every \( s \in S \),

\[
\frac{k}{4} < v(s) < \frac{k}{2} \text{ and } \sum_{s \in S} v(s) = kn.
\]

**Question:** Can \( S \) be partitioned into \( n \) sets \( S_1, \ldots, S_n \) in such a way that, for every \( 1 \leq i \leq n \),

\[
\sum_{s \in S_i} v(s) = k.
\]

Let \( f \) be the function which maps every instance \( I = (S,n,k,v) \) of 3-PARTITION into a graph \( f(I) \) in \( G_\Delta \) as follows. \( f(I) \) is the union of (i) \( n \) \( k \)-element b-paths, and (ii) for every \( s \in S \), a \( v(s) \)-element a-path (hence, every \( s \in S \) corresponds uniquely to an a-path in \( f(I) \)). (Thus, for every instance \( I \) of 3-PARTITION, \( f(I) \) is the union of at least one a-path and at least one b-path.)

It is easily seen that \( f \) has a polynomial-bounded time-complexity. Furthermore, for every instance \( I \) of 3-PARTITION, \( I \) is a "yes"-instance of 3-PARTITION if and only if \( f(I) \) is an element of \( L(G) \); this can be seen as follows.

Assume that \( I = (S,n,k,v) \) is a "yes" - instance of 3-PARTITION. Hence, there exists a partition \( (S_1, \ldots, S_n) \) of \( S \), such that, for every \( 1 \leq i \leq n \),

\[
\sum_{s \in S_i} v(s) = k.
\]

Furthermore, since \( f(I) \) is the union of at least one a-path and at least one b-path, we can partition \( f(I) \) into \( S_b = (\beta_1, \ldots, \beta_m) \) and \( S_a \), in the way described in the first part of the statement of Lemma 1 (note that now \( m \geq 1 \)).

Let, for every \( 1 \leq i \leq n \), \( S_a^i \) be the set of a-paths in \( S_a \) corresponding to the elements of \( S_i \). It is easily seen that: (1) \( m = n \), and (2) for every \( 1 \leq i \leq n \),

\[
\#\beta_i = k = \sum_{s \in S_i} v(s) = \sum_{s \in S_a^i} \#s.
\]

Consequently, by Lemma 1, \( f(I) \in L(G) \) which proves the "only if" - part of the claim.

Assume now that \( I = (S,n,k,v) \) is an instance of 3-PARTITION, such that \( f(I) \in L(G) \). Hence, because \( f(I) \) is the union of at least one a-path and at least one b-path, it follows from Lemma 1 that \( f(I) \) can be partitioned into

\[ S_b = \{\beta_1, \ldots, \beta_m\} \text{ consisting of b-paths, where } m \geq 1, \text{ and } S_a \text{ consisting of a-paths in such a way that there exists a partition } (S_a^1, \ldots, S_a^n) \text{ of } S_a, \text{ such that for} \]
every $1 \leq i \leq m$, $\#\beta_i = \sum_{\delta \in S_i} \#\delta$. Let, for every $1 \leq i \leq m$, $S_i$ be the set of elements in $S$ corresponding to the a-paths of $S_i$. It is easily seen that: (1) $m = n$, and (2) for every $1 \leq i \leq n$, $\sum_{s \in S_i} v(s) = \sum_{\delta \in S_i} \#\delta = \#\beta_i = k$.

Consequently, $I$ is a "yes"-instance of 3-PARTITION, which proves the "if"-part of the claim.

Since 3-PARTITION is NP-hard, the membership problem for $L(G)$ is NP-hard, and consequently, the membership problem for $L(G)$ is NP-complete. $\blacksquare$

The theorem follows directly from Lemma 2. $\blacksquare$

This is a rather surprising result. A regular DNLC grammar (with the empty out-connection relation) can be considered as a right-linear string grammar in which during the (left-to-right) generation of (the graph representation of) a string (with all edges resulted by "transitivity" also added) some edges are removed (cutted) according to the in-connection relation. It is well-known that the membership problem for right-linear grammars is linear while the membership problem for regular DNLC grammars turns out to be so difficult!!!

ACKNOWLEDGEMENTS

The second and the third author gratefully acknowledge the support by NSF Grant MCS 83-05245. The authors are indebted to H.J. Hoogeboom, H.C.M. Kleijn and E. Welzl for useful comments concerning the first version of this note.

REFERENCES


Figure 1. A "typical" graph in L(G).