

THE BOUNDED DEGREE PROBLEM FOR
NLC GRAMMARS IS DECIDABLE

by

Dirk Janssens ² , Grzegorz Rozenberg ¹ , and Emo Welz ^{2,3}

CU-CS-267-84

April, 1984

(1) University of Colorado at Boulder, Department of Computer Science, Campus Box 430, Boulder, Colorado, USA.

(2) Institute of Applied Mathematics and Computer Science, University of Leiden, 2300 RA Leiden, THE NETHERLANDS.

(3) On absence from: Institutes for Information Processing, IIG, Technical University of Graz and Austrian Computer Society, A-8010 Graz, AUSTRIA.

ANY OPINIONS, FINDINGS, AND CONCLUSIONS
OR RECOMMENDATIONS EXPRESSED IN THIS PUB-
LICATION ARE THOSE OF THE AUTHOR AND DO
NOT NECESSARILY REFLECT THE VIEWS OF THE
NATIONAL SCIENCE FOUNDATION.

The degree of a graph H is the maximum among the degrees of its nodes. A set of graphs L is of bounded degree if there exists a positive integer n such that the degree of each graph in L does not exceed n .

In this paper we demonstrate that it is decidable whether or not the (graph) language of an arbitrary node label controlled (NLC) grammar is of bounded degree. Moreover, it is shown that, given an arbitrary NLC grammar G generating the language $L(G)$ of bounded degree, one can effectively compute the maximum integer which appears as the degree of a graph in $L(G)$.

INTRODUCTION

The notion of a node label controlled (NLC) graph grammar was introduced by Janssens & Rozenberg (1980a) as an underlying model for a systematic build-up of a mathematical theory of graph grammars. Since then the theory of NLC grammars was quite extensively investigated (see, e.g., Janssens & Rozenberg, 1980b, 1981a, 1981b; Janssens, 1983; or Turán, 1983).

One of the important research areas within the theory of NLC grammars is that of the decision problems. Most of the decision problems for NLC grammars considered until now are of graph-theoretic nature. The typical questions are: (i) Does the generated language contain a discrete (planar, hamiltonian etc.) graph? (ii) Are all generated graphs connected? It turns out that most of the decision problems of this nature considered so far are undecidable.

In this paper we consider a decision problem that is quite fundamental from the graph theoretical point of view and which is also of conceivable interest in practical considerations. The problem can be stated as follows: "Given an NLC grammar G , is it decidable whether or not there is an integer n such that the degree of any node in any graph in the language, $L(G)$, generated by G does not exceed n , i.e., whether or not $L(G)$ is of bounded degree?" It turns out that this problem is decidable (-this settles the open problem from Janssens & Rozenberg, 1981a). Thus not all fundamental decision problems for NLC grammars are undecidable!

The decidability of this problem is rather surprising, since there are already two results on related questions: "It is undecidable whether or not the set of connected graphs in an NLC language is of bounded degree," (see Janssens & Rozenberg, 1981a), and "It is undecidable whether or not the axiom of an NLC grammar generates a graph (not necessarily terminal labeled) which, on its part, generates a set of graphs of bounded degree", (see Janssens, 1983,

Theorem 4.4). Hence our result points somehow to a borderline between decidable and undecidable for NLC grammars.

Moreover we would like to point out the following. Until now one has established a rather strong connection between the theory of NLC graph grammars and the theory of string grammars, in particular grammars from the classical Chomsky hierarchy (see, e.g., Janssens & Rozenberg, 1980b,1981b; and Janssens, 1983). A characteristic feature of these connections is that a number of techniques were established where derivations in a string grammar were simulated by derivations in NLC grammars. In the present paper we also establish a connection between string grammars and NLC grammars. However, two points appear to be novel. First of all, we establish a connection with parallel type grammars, namely with ETOL systems. Second, the technique used here allows one to "simulate" (for our special purposes!) a derivation in an NLC grammar by a derivation in a string grammar (that is in an ETOL system). As a matter of fact our key idea points out even a closer relationship - we discuss this briefly in the last section of the paper.

It is instructive to consider the present paper as a "companion paper" of Janssens & Rozenberg (1981a)- the two papers together shed more light on the nature of decision problems for NLC grammars. Although the basic notions concerning NLC grammars are recalled in this paper - this is done briefly and somewhat informally - the reader is referred to Janssens & Rozenberg (1981a) for complete and formal definitions. For basic notions concerning the theory of ETOL systems, the reader is referred to Rozenberg & Samlmaa (1980).

PRELIMINARIES

We recall here a number of definitions and notions from graph-, graph grammar-, and string grammar theory as far as they are needed in this paper.

Graphs.

We consider finite undirected node labeled graphs without loops and without multiple edges. Each such graph X is specified as a four-tuple $X=(V_X, E_X, \Sigma_X, \varphi_X)$, where V_X is a finite nonempty set of nodes, E_X is a set of two element subsets of V_X , (the set of edges), Σ_X is a set of labels, and φ_X is a function from V_X into Σ_X , (the labelling function). By $lab(X)$ we denote the set of labels which actually occur in X , that is $lab(X)=\{\varphi_X(x) | x \in V_X\}$. If, for some set of labels Σ , $lab(X) \subseteq \Sigma$, then X is called a graph over Σ . The set of all graphs over Σ is denoted by G_Σ .

Notions like connected graph, induced subgraph, discrete graph, graph isomorphism, etc. are defined in the usual way (see, e.g., Harary, 1969). Since the degree of a graph (language) is the central notion of this paper, we recall its definition now.

For a node x in a graph X , its degree, $deg_X(x)$, in X is defined as $deg_X(x) = \#\{y \in V_X | \{x, y\} \in E_X\}$. (For a finite set V , its cardinality is denoted by $\#V$.) The degree, $deg(X)$, of a graph X is the maximum integer which occurs as a degree of a node in X . A nonempty graph language L is of bounded degree, if $\{deg(X) | X \in L\}$, is finite. We define $deg(L) = \max\{deg(X) | X \in L\}$, if L is of bounded degree and $deg(L) = \infty$, otherwise.

For a graph X , $ind(X)$ is the set of (node) induced subgraphs of X . For a graph language L , $ind(L) = \bigcup_{X \in L} ind(X)$ and $conn(L)$ is the set of all connected graphs in L .

Graph grammars (NLC).

A node label controlled (NLC) grammar is a system $G=(\Sigma, \Delta, P, C, Z)$, where Σ is a finite nonempty set of labels, Δ is a nonempty subset of Σ (set of terminal labels), P is a finite set of pairs (d, Y) , where $d \in \Sigma$ and $Y \in G_\Sigma$, (set of productions), C is a subset of $\Sigma \times \Sigma$, (connection relation), and Z is a graph in G_Σ , (start graph).

Let X, \bar{Y}, \bar{X} be graphs in G_Σ with $V_X \cap V_{\bar{Y}} = \emptyset$ and let $x \in V_X$. Then X concretely derives \bar{X} (in G , replacing x by \bar{Y}), denoted $X \Rightarrow_{(x, \bar{Y})} \bar{X}$, if there is a production $(d, Y) \in P$, such that Y is isomorphic to \bar{Y} , $d = \varphi_X(x)$ and $V_{\bar{X}} = (V_X - \{x\}) \cup V_{\bar{Y}}$,

$$E_{\bar{X}} = (E_X - \{\{x, y\} \mid y \in V_X - \{x\}\}) \cup E_{\bar{Y}} \cup \\ \cup \{\{y, z\} \mid \{x, z\} \in E_X, y \in V_{\bar{Y}}, (\varphi_{\bar{Y}}(y), \varphi_X(z)) \in C\},$$

and, for $y \in V_{\bar{X}}$, $\varphi_{\bar{X}}(y) = \varphi_X(y)$, if $y \in V_X$ and $\varphi_{\bar{X}}(y) = \varphi_{\bar{Y}}(y)$, if $y \in V_{\bar{Y}}$.

Intuitively, we replace x in X by the graph \bar{Y} and connect a node y in \bar{Y} to a (former) neighbor z of x if and only if $(\varphi_{\bar{Y}}(y), \varphi_X(z)) \in C$.

A graph X directly derives a graph X' (in G), in symbols $X = X'$, if there is a graph \bar{X} isomorphic to X' , such that X concretely derives \bar{X} . $\overset{*}{=}$ is the transitive and reflexive closure of $=$. If $X \overset{*}{=} X'$, then we say that X derives X' (in G). A sequence of successive derivation steps

$$X_0 = X_1 = X_2 \dots = X_n, n \geq 0$$

is called a derivation (of X_n from X_0). Finally, the language, $L(G)$, generated by G is $L(G) = \{X \in G_\Delta \mid Z \overset{*}{=} X\}$.

String grammars (ETOL).

An ETOL system G is specified in the form $G=(V, \Delta, P, w)$ where V is its total alphabet, Δ is its terminal alphabet, P is its set of tables, and w is its axiom.

THE RESULT

The following three notions will be crucial in our considerations. In what follows, let Σ be a finite alphabet and let $\Sigma^\$ = \{a^\$ \mid a \in \Sigma\}$; we assume that $\Sigma^\$ \cap \Sigma = \emptyset$.

For a graph $X \in G_\Sigma$, $\text{mark}(X)$ denotes the set of graphs defined by

$$\text{mark}(X) = \{Y \in G_{\Sigma \cup \Sigma^\$} \mid V_Y = V_X, E_Y = E_X, \text{ and there is exactly one node } y \in V_Y, \text{ such that } \varphi_Y(y) = a^\$, \text{ where } a = \varphi_X(y), \text{ and for all } x \in V_Y, x \neq y, \varphi_Y(x) = \varphi_X(x)\}.$$

Hence a graph in $\text{mark}(X)$ is obtained from X by replacing the label, say a , of exactly one node in X by $a^\$$. For a language $L \subseteq G_\Sigma$, we define $\text{mark}(L) = \bigcup_{X \in L} \text{mark}(X)$. For $X \in \text{mark}(G_\Sigma)$, the (unique) node x of X that is labeled by an element in $\Sigma^\$$ is called the marked node of X .

Let $X \in G_\Sigma \cup \text{mark}(G_\Sigma)$. By $\text{red}(X)$ we denote the graph Y , such that $V_Y = V_X, \varphi_Y = \varphi_X$, and $E_Y = \{\{x, y\} \in E_X \mid \text{either } \varphi_X(x) \in \Sigma^\$ \text{ or } \varphi_X(y) \in \Sigma^\$\}$. Hence, if $X \in \text{mark}(G_\Sigma)$, then $\text{red}(X)$ is obtained from X by omitting all edges in X which are not incident with the marked node of X , and if $X \in G_\Sigma$, then $\text{red}(X)$ is the discrete graph on V_X . For a language $L \subseteq G_\Sigma \cup \text{mark}(G_\Sigma)$, we define $\text{red}(L) = \{\text{red}(X) \mid X \in L\}$.

For a language $L \subseteq G_\Sigma$, we denote by $\text{star}(L)$ the set

$$\text{star}(L) = \text{conn}(\text{ind}(\text{red}(\text{mark}(L)))) \cap \text{mark}(G_\Sigma).$$

Note that each graph in $\text{star}(L)$ is obtained from a graph in L by (i) replacing the label $a \in \Sigma$ of one node by $a^\$$, (ii) removing all nodes except for the $a^\$$ -labeled node itself and some of its neighbors, and (iii) removing all edges apart from those between the $a^\$$ -labeled node and its not removed neighbors.

It is easily seen that $\text{deg}(L) = \text{deg}(\text{star}(L))$.

THEOREM 1. It is decidable whether or not the language of an arbitrary NLC grammar is of bounded degree.

Proof. Let $G=(\Sigma,\Delta,P,C,Z)$ be an NLC grammar. Clearly, we can assume that $\#V_Z=1$. Let $L=L(G)$. The proof is presented in two steps.

In the first one an NLC grammar G''' is constructed, such that $\text{conn}(L(G'''))=\text{star}(L)$. It easily follows that $\text{deg}(L)=\text{deg}(\text{conn}(L(G''')))$ and that $\text{conn}(L(G'''))$ is of bounded degree if and only if it is finite.

In the second step we introduce an ETOL system G_0 which generates the language $L(G_0)$ in which each word of length n corresponds to a graph on n nodes from $\text{conn}(L(G'''))$. In this way the "bounded degree problem" for NLC grammars is reduced to the "finiteness problem" for ETOL systems which is known to be decidable (see, e.g., Rozenberg & Salomaa, 1980).

STEP I.

Let $L'=\text{mark}(L)$. Consider the NLC grammar $G'=(\Sigma\cup\Sigma^\$, \Delta\cup\Delta^\$, P', C', Z')$, where $\Delta^\$=\{a^\$ \in \Sigma^\$ \mid a \in \Delta\}$,

$$P' = \text{PU}\{(d^\$, Y) \mid Y \in \text{mark}(Y'), \text{ for some production } (d, Y') = P\},$$

$$C' = \text{CU}\{a^\$, b) \mid (a, b) \in C\} \cup \{(a, b^\$) \mid (a, b) \in C\},$$

and if a is the label of the node in Z , then let Z' be a node graph with its node labeled by $a^\$$. It is easily seen that $L(G')=L'$.

Let $L''=\text{red}(L')$. Consider the NLC grammar $G''=(\Sigma\cup\Sigma^\$, \Delta\cup\Delta^\$, P'', C'', Z'')$, where

$$P'' = \{(d, \text{red}(Y)) \mid (d, Y) \in P'\},$$

$$C'' = C' \cap (\Sigma \times \Sigma^\$ \cup \Sigma^\$ \times \Sigma),$$

and $Z''=Z'$. It is easily seen that $L(G'')=L''$. Observe that each graph in $L(G'')$

consists of an element of $\text{star}(L)$ together with zero or more isolated nodes.

Let $L''' = \text{ind}(L'') \cap \text{mark}(G_\Sigma)$. Consider the NLC grammar $G''' = (\Sigma \cup \Sigma^\$, \Delta \cup \Delta^\$, P''', C''', Z''')$, where

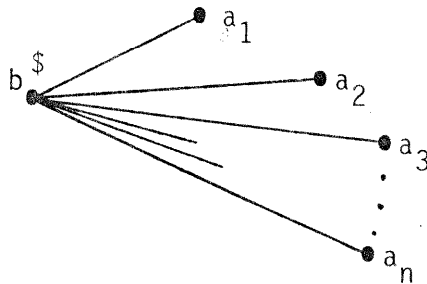
$$P''' = \{(d, Y) \mid d \in \Sigma, Y \in \text{ind}(Y'), \text{ for a production } (d, Y') \in P''\} \cup \{(d^\$, Y) \mid d^\$ \in \Sigma^\$, Y \in \text{ind}(Y') \cap \text{mark}(G_\Sigma), \text{ for a production } (d^\$, Y') \in P''\},$$

$C''' = C''$, and $Z''' = Z''$. It is easily seen that $L(G''') = L'''$ and it follows from the construction that $\text{conn}(L(G''')) = \text{star}(L)$.

STEP II

Consider a derivation in G''' of a graph in $\text{conn}(L(G'''))$. Since in an NLC grammar a graph derived from a disconnected graph is also disconnected, each graph occurring in such a derivation is an element of $\text{star}(G_\Sigma)$. We distinguish two kinds of derivation steps in such a derivation.

(i) Derivation steps $X = \bar{X}$ in which the marked node m of X is rewritten, that is, $\varphi_X(m) \in \Sigma^\$$. Then m is replaced by a graph of the form



$n \geq 0$, where $b^\$ \in \Sigma^\$$, $a_1, a_2, a_3, \dots, a_n \in \Sigma$, and for each neighbor x of m in X , $(b^\$, \varphi_X(x)) \in C'''$. That is, $\text{lab}(X) \cap \{a \mid (b^\$, a) \in C'''\} = \emptyset$. Hence whether or not the application of a production $(a^\$, Y)$, $a^\$ \in \Sigma^\$, Y \in \text{star}(G_\Sigma)$, to a graph X in $\text{star}(G_\Sigma)$ results in a connected graph depends only on $\text{lab}(X)$. Accordingly, a production $p = (a^\$, Y)$, with $a^\$ \in \Sigma^\$, Y \in \text{star}(G_\Sigma)$, is called good production on $\Sigma^\$$ with forbidden set $F_p = \{a \mid (b^\$, a) \notin C'''\}$, where $b^\$$ is the label of the marked node of Y .

(ii) Derivation steps $X \rightarrow \bar{X}$ in which a neighbor x of the marked node m in X is rewritten, that is, $\varphi_X(x) \in \Sigma$, $\varphi_X(m) \in \Sigma^\$$. Then x is replaced by a discrete graph Y such that for each node $y \in V_Y$, $(\varphi_Y(y), \varphi_X(m)) \in C'''$. That is, $\text{lab}(Y) \subseteq \{b \mid (b, \varphi_X(m)) \in C'''\}$, or, equivalently, formulated as a condition on X , $\text{lab}(X) \cap \{a^\$ \in \Sigma^\$ \mid \text{lab}(Y) \not\subseteq \{b \mid (b, a^\$) \in C'''\}\} = \emptyset$. Hence, again, whether or not the graph \bar{X} is connected depends only on $\text{lab}(X)$. Accordingly, a production $p = (a, Y)$ with $a \in \Sigma$, Y a discrete graph, is called good production on Σ with forbidden set $F_p = \{a^\$ \in \Sigma^\$ \mid \text{lab}(Y) \not\subseteq \{b \mid (b, a^\$) \in C'''\}\}$.

Note that the above observations show that our problem became rather independent of the graph structures involved. This fact will be exploited as follows.

We will define an ETOL system $G_0 = (V_0, \Delta U \Delta^\$, P_0, w_0)$ such that:

(*) A graph X is in $\text{conn}(L(G'''))$ if and only if (X is in $\text{star}(G_\Delta)$ and) there is a word w in $L(G_0)$ with $\#_a(w) = \{x \in V_X \mid \varphi_X(x) = a\}$, for all $a \in \Delta U \Delta^\$$.

($\#_a(w)$ denotes the number of occurrences of a in w .) Consequently, $L(G)$ is of bounded degree if and only if $L(G_0)$ is finite. Since the finiteness problem for ETOL systems is known to be decidable, the theorem follows.

Thus to complete the proof we provide now the construction of an ETOL system $G_0 = (V_0, \Delta U \Delta^\$, P_0, w_0)$ satisfying (*) above. Let $V_0 = \Sigma U \Sigma^\$ U \{N\}$, where N is a garbage letter with $N \in \Sigma U \Sigma^\$$. $w_0 \in \Sigma^\$$ is the label of the axiom Z''' in G''' . Let $p = (d, Y) \in P'''$ be a good production on Σ or $\Sigma^\$$ with forbidden set F_p and let u be a word in $(\Sigma U \Sigma^\$)^*$ such that $\#_a(u) = \#\{y \in V_Y \mid \varphi_Y(y) = a\}$ for all $a \in (\Sigma U \Sigma^\$)^*$. Then there is a table P_p in P_0 with the following productions:

$d \rightarrow u$ (called the essential production of P_p),
 $a \rightarrow a$ for all $a \in V_0 - F_p$, and
 $a \rightarrow N$ for all $a \in F_p$.

Note that $d \rightarrow d$ is always in the table, since $d \notin F_p$. No other tables but those (of the form P_p) defined above are in P_0 .

We will not explicitly prove that G_0 fulfills assumption (*) above, rather we point out two crucial observations.

(i) Whenever there is a derivation of a word w over

$\Sigma U \Sigma^\$$ in G_0 , then there is always a derivation of w in G_0 such that in every step the essential production of the used table P_p (from P_0) is applied exactly once. This stems essentially from the facts that (a) there is always at most one letter from $\Sigma^\$$ in a word derived from the axiom w_0 , and (b) whenever there is a table P_p corresponding to a good production $p \in P'''$ on Σ , and $d \rightarrow u$ is the essential production of P_p , no letter of u appears in F_p , since in this case $u \in \Sigma^*$, while $F_p \subset \Sigma^\$$. (Hence parallel applications of an essential production can be "sequentialized").

(ii) Deriving in G_0 a word containing the garbage letter N corresponds to deriving a disconnected graph in G''' .

Observations (i) and (ii) indicate clearly how derivations of connected graphs in $L(G''')$ correspond to derivations of words in $L(G_0)$ and the other way round. As discussed above, this completes the proof by reduction to a standard result in L-theory. \square

Actually, one can prove a stronger result.

THEOREM 2. For an arbitrary NLC grammar G , $\text{deg}(L(G))$ can be effectively computed.

Proof. Let G be an NLC grammar. By Theorem 1 we can decide whether or not $L(G)$ is of bounded degree.

If G is not of bounded degree, then $\text{deg}(L(G)) = \infty$.

If G is of bounded degree, then, for $k=1,2,3,\dots$, (in this order) we decide whether or not $L(G''')$ contains a graph $X \in \text{star}(G_{\Delta})$ on k nodes (where G''' is defined as in the proof of Theorem 1). If k_0 is the smallest number for which such a graph does not exist, then $\text{deg}(L(G)) = k_0 - 2$. (Note that if $L(G''')$ contains a graph of degree j , $j \geq 1$, then it contains a graph of degree $j-1$.) \square

DISCUSSION

The proof we presented here essentially relied on the reduction to the finiteness problem for ETOL systems. It turns out that a crucial idea of our proof is similar to that of a proof in Penttonen (1975), where it is shown that every language generated by an N-grammar without recurrent productions can be generated by an ETOL-system. To be more precise, an N-grammar is a "context-free" grammar, where every production has an additional "context condition" in the form of a set of symbols. A production is applicable to a word w only if no symbol of the corresponding set (representing the context condition) occurs in w . The direct correspondence between these "context condition sets" and the forbidden sets F_p from the proof of Theorem 1 should be obvious.

The problem whether or not an N-grammar generates a finite set seems to be open. (At least, there is no solution known to the authors and, moreover, this problem has been explicitly stated in Stotskii (1971) for ordered context-free grammars which can be easily shown to be constructively equivalent to N-grammars.)

Obviously, the reduction to ETOL systems in Theorem 1 gives a solution to the finiteness problem for N-grammars in a very special case. The more general problem stated above has also a direct correspondence in the framework of NLC grammars. It is not too difficult to prove that the finiteness problem for N-grammars is decidable if and only if it is decidable whether or not the graph language generated by an NLC grammar contains a finite number of complete graphs.

Hence this paper demonstrates, once again, various ways in which string and NLC grammars interact, extending thus the picture given in Janssens & Rozenberg (1980b,1981b) and Janssens (1983).

Acknowledgement: We thank Jetty Kleijn for discussion and for pointing out some references. The third author acknowledges support of NSF grant No. MCS 83-05245.

REFERENCES

- Harary, F. (1969), Graph Theory, Addison Wesley, Reading Mass.
- Janssens, D. (1983), Node Label Controlled Graph Grammars, Ph.D.Thesis, University of Antwerp.
- Janssens, D. & Rozenberg, G. (1980a), On the structure of node label controlled graph languages, Inform. Sci. 20, 191-216.
- Janssens, D. & Rozenberg, G. (1980b), Restrictions, extensions, and variations of NLC grammars, Inform. Sci. 20, 217-244.
- Janssens, D. & Rozenberg, G. (1981a), Decision problems for node label controlled graph grammars, J.Comput.System Sci. 22, 144-174.
- Janssens, D. & Rozenberg, G. (1981b), A characterization of context-free string languages by directed node label controlled graph grammars, Acta Inform. 16, 63-85.
- Penttonen, M. (1975), ETOL grammars and N-grammars, Inform. Process. Lett. 4, 11-13.
- Rozenberg, G. & Salomaa, A. (1980), The Mathematical Theory of L-Systems, Academic Press, New York.
- Stotskii, E., D. (1971), Control of the conclusion in formal grammars, Problems Inform.Transmission 7, 257-270.
- Turán, Gy. (1983), On the complexity of graph grammars, Acta Cybernet. 6, 271-280.