GRAPH THEORETIC CLOSURE PROPERTIES OF THE FAMILY OF BOUNDARY NLC GRAPH LANGUAGES

by

G. Rozenberg and E. Welzl

CU-CS-294-85 April, 1985

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

GRAPH THEORETIC CLOSURE PROPERTIES OF THE FAMILY OF BOUNDARY NLC GRAPH LANGUAGES

by

Grzegorz Rozenberg and Emo Welzl*)

November 1984

Institute of Applied Mathematics and Computer Science

University of Leiden

Wassenaarseweg 80, Leiden

The Netherlands

*) On leave from: Institutes for Information Processing IIG, Technical University of Graz and Austrian Computer Society, Schiesstattgasse 4a, A-8010 Graz, AUSTRIA.

Node label controlled (NLC) *grammars* are graph grammars (operating on node labeled undirected graphs) which rewrite single nodes only and establish connections between the embedded graph and the neighbors of the rewritten node on the basis of the labels of the involved nodes only. They define (possibly infinite) languages of undirected node labeled graphs (or, if we just omit the labels, languages of unlabeled graphs). *Boundary* NLC (BNLC) *grammars* are NLC grammars with the property that whenever - in a graph already generated - two nodes may be rewritten, then these nodes are not adjacent. The graph languages generated by this type of grammars are called BNLC *languages*.

The present paper continues the investigations of basic properties of BNLC grammars and languages where the central question is the following: "If L is a BNLC language and $\mathcal{P}$ is a graph theoretic property, is the set of all graphs from L satisfying $\mathcal{P}$ again a BNLC language?" We demonstrate that the class of BNLC languages is very "stable" in the sense that for almost all properties we consider the resulting languages are BNLC. In particular, the above question gets an affirmative answer, if the property $\mathcal{P}$ is: being k-colorable, being connected, having a subgraph homeomorphic to a given graph, and being nonplanar.

INTRODUCTION

Node label controlled (NLC) grammars are graph grammars operating on node labeled undirected graphs. A production in an NLC grammar is a pair (d,Y), where d is a label and Y is a graph. Such a production is applicable to a node x in a graph X if and only if x is labeled by d. The rewriting process consists of (i) deleting x in X (together with incident edges), (ii) adding Y disjointly to the remainder of X and (iii) establishing connections between nodes in Y and ("former") neighbors of x in the remainder of X. This embedding is controlled by a so-called connection function conn which maps labels to sets of labels. More specifically, a neighbor z (of x) labeled by a is connected to a node y (of Y) labeled by b if and only if $a \in$ conn(b). The graph language generated by an NLC grammar consists of the set of all graphs such that (i) they can be obtained from the axiom (graph) Z_{ax} of the grammar by a sequence of rewritings, and (ii) they have labels only from the set Δ of terminal labels of the grammar.

NLC grammars have been introduced by Janssens & Rozenberg (1980a,b) as a basic framework for the mathematical investigation of graph grammars (the more general work on the theory of graph grammars is well presented in Nagl, 1979, and Ehrig, 1979). Since then this model has been intensively investigated, see e.g. Janssens & Rozenberg (1981), Brandenburg (1983), Turán (1983), Ehrenfeucht et al. (1984) and Janssens et al. (1984). In particular, it has turned out that most basic problems of graph theoretic nature concerning NLC grammars (languages) are undecidable. Although the membership problem for NLC grammars is decidable, NLC grammars can generate PSPACE-complete graph languages. Results like this have inspired a search for feasible but "nontrivial" subclasses of the class of NLC grammars (see e.g. Janssens, 1983).

The class of boundary NLC grammars, BNLC grammars for short, has been defined as follows (Rozenberg & Welzl, 1984). An NLC grammar is a BNLC grammar if (i) the left-hand side of each production is a nonterminal label, and (ii) all the graphs involved (i.e., the axiom and the right-hand sides of productions) are such that two nonterminally labeled nodes are never adjacent. It turns out that the class of BNLC languages (i.e., the graph languages generated by BNLC grammars) can be defined by using the subclass of NLC grammars in which (i) the left-hand side of each production is a nonterminal label and (ii) the range of the connection function consists of terminal labels only. Hence, on the one hand one can view BNLC grammars as an analogue (in the framework of NLC grammars) of fundamental subfamilies of context-free string grammars (such as linear grammars or context-free grammars in operator normal form), while, on the other hand, one gets a characterization of BNLC languages by considering a restriction on NLC grammars that is certainly a very natural one from the mathematical point of view.

In Rozenberg & Welzl (1984) a systematic investigation of BNLC grammars has been initiated. Among others, it has been demonstrated that quite a number of interesting families of graphs can be generated by BNLC grammars (e.g. maximal outerplanar graphs, 2-trees, graphs of cyclic bandwidth ≤ 2) and that (as opposed to the general NLC case) BNLC languages can be attractive from the "complexity" point of view (the membership problem in BNLC languages can be solved in polynomial time for connected graphs of fixed bounded degree).

In the present paper we continue the investigation of BNLC grammars and languages. In particular, this paper focusses on the behaviour of BNLC languages under various "squeezing mechanisms". The typical question considered is of the following type. "Let P be a graph theoretic property. Is, for an arbitrary BNLC language L, the set of all graphs from L satisfying P again a BNLC language?" Typical properties considered are connectedness, k-colorability, having a subgraph homeomorphic to a fixed graph, planarity, having clique number not exceeding k, where k is a fixed positive integer.

It turns out that the family of BNLC languages is "very stable" in the sense that for almost all properties we consider (and we consider many of them) the resulting languages are again BNLC languages. This certainly sheds light on the mathematical nature of the family of BNLC languages, and, moreover, such a stability turns out to be technically very useful in proving various properties (e.g. decidability, complexity and combinatorial properties).

This paper is organized as follows. After recalling in Section 1 some preliminaries from graph theory and the theory of graph grammars, basic definitions, examples and basic properties concerning BNLC grammars are given in Section 2. In Section 3 we investigate "subgraph-taking operations" when applied to BNLC languages. It appears that the set of all induced subgraphs from a BNLC language is again a BNLC language while, in general, this is not the case when one considers arbitrary subgraphs. In Section 4 we show that, for a natural number k, the set of all graphs in a BNLC language which are k-colorable is again a BNLC language. Moreover, "relabelings" of BNLC grammars are considered which are technically important in various proofs. Section 5 deals with connectedness and we prove there that both the set of all disconnected graphs and the set of all connected graphs from a BNLC language are BNLC languages. In Section 6 we demonstrate that, for an arbitrary given graph Z, the set of all graphs from a BNLC language which have a subgraph homeomorphic to Z is again a BNLC language. From this result it easily follows that the nonplanar graphs from a BNLC language form again a BNLC language. Finally, in Section 7, it is shown that, for an arbitrary given natural number k, the set of all graphs from a BNLC language which have no complete subgraph on k nodes is again a BNLC language. A short discussion in Section 8 concludes the paper.

1. PRELIMINARIES

We start with basic notations concerning graphs and graph grammars which we need for this paper. We assume familiarity with rudimentary graph theory. In particular, we use the following notions as defined in Harary (1969): adjacent, neighbor, degree of a node in a graph, subgraph, induced subgraph, path in a graph, complete graph, cycle, totally disconnected graph. Note, however, that (i) a k-coloring of a graph in "Harary's sense", corresponds to a proper k-coloring as we will define it (following Bondy & Murty, 1976), and (ii) two graphs are homeomorphic in "Harary's sense", if they are homeomorphic to a common graph as we will define it (following Garey & Johnson, 1979).

For a finite set V, we denote its cardinality by #V.

Graphs

We consider finite undirected node labeled graphs without loops and without multiple edges. For a set of labels Σ, a graph X (over Σ) is specified by a finite set V_X of nodes, a set E_X of two element subsets of V_X (the set of edges), and a function φ_X from V_X into Σ (the labeling function). The set of all graphs over Σ is denoted by G_Σ. The unique graph on the empty set of nodes is called the empty graph and it is denoted by λ.

Let X be a graph and let $x \in V_X$. The label set of X, lab(X), is the set $\{\varphi_X(y) \mid y \in V_X\}$. The neighborhood of x in X, $neigh_X(x)$, is the set $\{y \in V_X \mid \{x,y\} \in E_X\}$. The context of x in X, $cont_X(x)$, is the set $\{\varphi_X(y) \mid y \in neigh_X(x)\}$. The graph X-x is the subgraph of X induced by $V_X - \{x\}$. A graph X' is isomorphic to X, if there is a bijection from $V_{X'}$ to V_X which preserves labels and adjacencies. The set of all graphs isomorphic to X is denoted by [X]. The size of X, #X, is the number of nodes in X, i.e., $\#X = \#V_X$.

Let Σ and Σ' be sets of labels. A relabeling ρ from Σ' (to Σ) is a function from Σ' into Σ. If X is a graph over Σ, then the ρ-image of X, $\rho(X)$, is defined by $V_{\rho(X)} = V_X$, $E_{\rho(X)} = E_X$ and for all $x \in V_{\rho(X)}$, $\varphi_{\rho(X)}(x) = \rho(\varphi_X(x))$.

Disregarding the labeling function of X, one gets the underlying unlabeled graph of X, denoted by und(X). For a set L of graphs we denote by und(L) the set $\{\underline{\text{und}}(X) \mid X \in L\}$.

Graph Grammars

A node label controlled (NLC) grammar is a system $G = (\Sigma, \Delta, P, \underline{\text{conn}}, Z_{ax})$, where Σ is a finite nonempty set of labels, Δ is a nonempty subset of Σ (the set of terminals), P is a finite set of pairs (d,Y), where $d \in \Sigma$ and $Y \in G_\Sigma$ (the set of productions), conn is a function from Σ into 2^Σ (the connection function), and $Z_{ax} \in G_\Sigma$ (the axiom).

By [P] we denote the abstract production set $\{(d,Y') \mid Y' \in [Y] \text{ for some } (d,Y) \in P\}$. By maxr(G) we denote $\underline{\max}(\{\#Z_{ax}\} \cup \{\#Y \mid (d,Y) \in P \text{ for some } d \in \Sigma\})$.

The set $\Sigma - \Delta$ is referred to as the set of nonterminals and we will reserve the symbol Γ (possibly with an appropriate inscription) to denote $\Sigma - \Delta$. In the context of G, given a graph $X \in G_\Sigma$ we refer to nodes labeled by elements of Γ (Δ, respectively) as nonterminal nodes (terminal nodes, respectively).

Let X,Y,Z be graphs over Σ with $V_X \cap V_Y = \emptyset$ and let $x \in V_X$. Then X concretely derives Z (in G, replacing x by Y), denoted by $X \underset{G}{\Rightarrow}_{(x,Y)} Z$, if

$(\varphi_X(x),Y) \in [P]$, $V_Z = V_{X-x} \cup V_Y$,

$$E_Z = E_{X-x} \cup E_Y \cup \{\{x',y\} \mid x' \in \underline{\text{neigh}}_X(x),\ y \in V_Y,\ \varphi_X(x') \in \underline{\text{conn}}(\varphi_Y(y))\},$$

φ_Z equals φ_{X-x} on V_{X-x}, and φ_Z equals φ_Y on V_Y. (Intuitively speaking, we replace x in X by the graph Y and connect a node y of Y to a neighbor x' of x if and only if $\varphi_X(x') \in \underline{\text{conn}}(\varphi_Y(y))$.)

A graph X <u>directly derives</u> a graph Z (<u>in</u> G), in symbols $X \underset{G}{\Longrightarrow} Z$, if there is a graph $Z' \in [Z]$, such that X concretely derives Z' in G. $\overset{*}{\underset{G}{\Longrightarrow}}$ is the transitive and reflexive closure of $\underset{G}{\Longrightarrow}$. If $X \overset{*}{\underset{G}{\Longrightarrow}} Z$, then we say that X <u>derives</u> Z (<u>in</u> G). If G is understood, then we often omit the inscription G in $\underset{G}{\Rrightarrow}$, $\underset{G}{\Longrightarrow}$, and $\overset{*}{\underset{G}{\Longrightarrow}}$.

The <u>exhaustive language of</u> G, S(G), is the set $\{X \in G_\Sigma \mid Z_{ax} \overset{*}{\underset{G}{\Longrightarrow}} X\}$ and the <u>language of</u> G, L(G), is the set $\{X \in G_\Delta \mid Z_{ax} \overset{*}{\underset{G}{\Longrightarrow}} X\}$.

A graph language L is an NLC <u>language</u> if there is an NLC grammar G such that L = L(G).

2. DEFINITIONS

Let Φ be a set of labels. A graph X is a Φ-*boundary* graph, if no two adjacent nodes of X that are labeled by elements of Φ are adjacent.

A *boundary* NLC (BNLC) *grammar* is an NLC grammar $G = (\Sigma, \Delta, P, \underline{conn}, Z_{ax})$, where Z_{ax} is a Γ-boundary graph and, for all $(d,Y) \in P$, $d \in \Gamma$ and Y is a Γ-boundary graph. A graph language L is a BNLC *language*, if there is a BNLC grammar G such that $L = L(G)$. A language L of unlabeled graphs is a u-BNLC *language*, if there is a BNLC language L' such that $L = \underline{und}(L')$. (Recall, that we set implicitely $\Sigma - \Delta = \Gamma$!)

We give now examples of BNLC grammars and BNLC languages, which we need also for forthcoming proofs (Theorems 3.2 and 4.1).

Example 1 (*complete graphs*). Consider the BNLC grammar $G_1 = (\{A,b\},\{b\}, P_1, \underline{conn}_1, Z_1)$, where $\underline{conn}_1(b) = \underline{conn}_1(A) = \{b\}$, Z_1 is a graph with a single node labeled by A, and P_1 consists of the productions $(A, {}^{b}\bullet\!\!-\!\!\bullet^{A})$ and $(A, \bullet^{b})$. Then it is easily seen that $L(G_1)$ consits of all complete graphs with all nodes labeled by b. □

Example 2 (*cycles*). Consider the BNLC grammar $G_2 = (\{A,b,c\},\{b,c\},P_2,\underline{conn}_2,Z_2)$, where $\underline{conn}_2(A) = \{c\}$, $\underline{conn}_2(b) = \{b\}$, $\underline{conn}_2(c) = \{b,c\}$, Z_2 is a triangle with its nodes labeled by A, b and c, and P_2 consists of the productions $(A, {}^{b}\bullet\!\!-\!\!\bullet^{A})$ and $(A, \bullet^{c})$. Then $L(G_2)$ consists of all cycles where all nodes are labeled by b except for two adjacent nodes labeled by c. Hence, $\underline{und}(L(G_2))$ is the set of all unlabeled cycles. □

The *bandwidth* of a graph X is the minimum integer k for which there exists a bijection f from V_X to $\{1,2,\ldots,\#X\}$ such that for all $\{x,y\} \in E_X$, $|f(x) - f(y)| \le k$.

Example 3 (graphs of bandwidth $\leq k$). Let k be a natural number and let $K = \{1,2,\ldots,k\}$. Consider the BNLC grammar $G_3 = (\Sigma_3,\Delta_3,P_3,\underline{conn}_3,Z_3)$ where $\Delta_3 = K \times 2^K$, $\Sigma_3 = \Delta_3 \cup K$, $\underline{conn}_3((i,r)) = \{(j,s) \in \Delta_3 | j \in r\}$ for all $(i,r) \in \Delta_3$ and $\underline{conn}_3(i) = \{(j,s) \in \Delta_3 | j \neq i\}$, for all $i \in K$, and Z_3 is a graph consisting of one node labeled by $1 \in K$. P_3 consists of all productions of the form $(i, {}^{(j,s)}\bullet\!\!-\!\!-\!\!\bullet^{j})$ and $(i, \bullet^{(j,s)})$ where $s \subseteq K$, $j = 1$ for $i = k$, while $j = i+1$, otherwise.

Fig. 2.1. depicts a derivation in G_3. It is not difficult to see that $\underline{und}(L(G_3))$ consists of all graphs of bandwidth $\leq k$. □

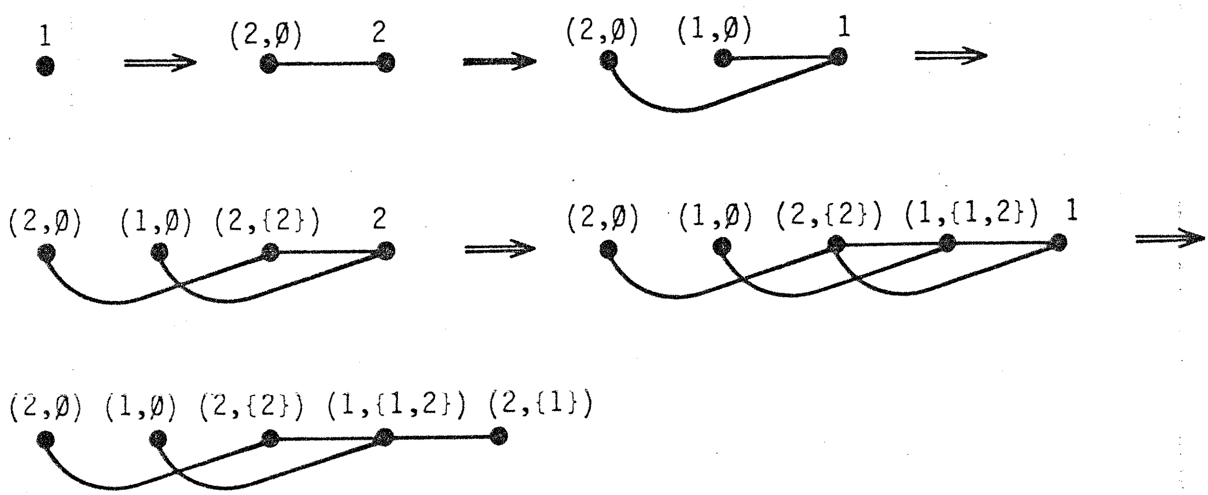

Fig. 2.1. A derivation of a graph of bandwidth ≤ 2 in G_3 (for $k = 2$).

Since it has been shown in Rozenberg & Welzl (1984, Theorem 6.6) that for an integer d and a u-BNLC language L, "$X \in L$?" can be decided in polynomial time for connected graphs X of maximal degree $\leq d$, Example 3 proves that for a natural number k, it can be decided in polynomial time whether a graph is of bandwidth $\leq k$. (Note, that a graph of bandwidth $\leq k$ is of maximal degree $\leq 2k$, and that a graph is of bandwidth $\leq k$ if and only if each of its connected components is of bandwidth $\leq k$). Although, this has already been demonstrated in Saxe (1980), it is interesting to note that this result can be obtained via BNLC grammars. Of course, we cannot deduce the concrete $O(n^k)$ time bound for the recognition of graphs with bandwidth $\leq k$ (see Gurari & Sudborough, 1982).

For more examples of BNLC grammars and languages we refer to Rozenberg & Welzl (1984), where also a number of basic properties of BNLC grammars have been elaborated. We recall here three of these properties, as they are often implicitely used in many proofs concerning BNLC grammars.

PROPOSITION 2.1. Let $G = (\Sigma,\Delta,P,\underline{conn},Z_{ax})$ be a BNLC grammar. Then every graph in $S(G)$ is a Γ-boundary graph. □

PROPOSITION 2.2. Let G be a BNLC grammar. Let $X_0 \in S(G)$, let $x,y \in V_{X_0}$ and let Y_1,Y_2,X_1,X_2 be graphs such that

$$X_0 \underset{G}{\Rightarrow} (x,Y_1)\ X_1 \underset{G}{\Rightarrow} (y,Y_2)\ X_2$$

holds. If X_1' and X_2' are the graphs, such that

$$X_0 \underset{G}{\Rightarrow} (y,Y_2)\ X_1' \underset{G}{\Rightarrow} (x,Y_1)\ X_2'$$

holds, then $X_2' = X_2$. □

We use the following normal forms for BNLC grammars. Let $G = (\Sigma,\Delta,P,\underline{conn},Z)$ be a BNLC grammar.

G is normalized, if (1) for all $(A,Y) \in P$, $\#Y \geq 1$, (2) $\#Z = 1$, and (3) for all $d \in \Sigma$, $\underline{conn}(d) \subseteq \Delta$.

G is context consistent, if there is a function η from Γ into 2^{Σ} with the following property: for every graph $X \in S(G)$ and for every nonterminal node $x \in V_X$, $\underline{cont}_X(x) = \eta(\varphi_X(x))$ holds. The function η satisfying the above is called the context describing function of G.

G is chain-free, if, for all $(A,Y) \in P$ with $V_Y = \{y\}$ (i.e., $\#Y = 1$), y is a terminal node.

PROPOSITION 2.3. For every BNLC language L there is a normalized, context-consistent, and chain-free BNLC grammar G such that $L(G) = L - \{\lambda\}$. □

In what follows we consider two graph languages to be equal if they coincide up to the empty graph λ.

We conclude this section by providing a technical tool which will be needed in forthcoming proofs.

Concrete derivations

Let $G = (\Sigma,\Delta,P,conn,Z_{ax})$ be an NLC grammar. If a graph X concretely derives a graph Z in G, replacing a node x by a graph Y, then, somewhat informally, we refer to the construct $X \Rightarrow_{(x,Y)} Z$ as a concrete derivation step in G (from X to Z).

A sequence of "successive" concrete derivation steps in G

$$D: X_0 \Rightarrow_{(x_0,Y_1)} X_1 \Rightarrow_{(x_1,Y_2)} X_2 \cdots \Rightarrow_{(x_{n-1},Y_n)} X_n,$$

where $n \geq 0$ and the sets V_{X_0}, V_{Y_i}, $1 \leq i \leq n$, are pairwise disjoint, is referred to as a concrete derivation in G (from X_0 to X_n).

The node set of D is $V_D = \bigcup_{i=0}^{n} V_{X_i}$. The edge set of D is $E_D = \bigcup_{i=0}^{n} E_{X_i}$. The labeling function φ_D of D is defined by $\varphi_D(x) = \varphi_{X_0}(x)$ if $x \in V_{X_0}$ and $\varphi_D(x) = \varphi_{Y_i}(x)$ if $x \in V_{Y_i}$ for some i, $1 \leq i \leq n$. Note that $V_D = V_{X_0} \cup \bigcup_{i=1}^{n} V_{Y_i}$, hence φ_D is defined on the whole set V_D. Moreover, if $x \in V_{X_i}$ for some i, $0 \leq i \leq n$, then $\varphi_{X_i}(x) = \varphi_D(x)$. Thus every concrete derivation D defines naturally a graph with set of nodes V_D, set of edges E_D and labeling function φ_D; this justifies our abuse of notation in using V_D, E_D, and φ_D when referring to various elements of a concrete derivation D. Note that this "graph" D is a Γ-boundary graph whenever X_0 is a Γ-boundary graph and G is a BNLC grammar.

Let $\mathcal{O}_D$ be a distinguished element not in V_D which is called the <u>origin of</u> D. The <u>predecessor mapping</u> $\underline{pred}_D$ <u>of</u> D is a function from V_D into $V_D \cup \{\mathcal{O}_D\}$ such that for $x \in V_D$

$$\underline{pred}_D(x) = \begin{cases} \mathcal{O}_D & \text{if } x \in V_{X_0}\text{, and} \\ x_i & \text{if } x \in V_{Y_{i+1}} \text{ for an } i,\ 0 \le i \le n-1. \end{cases}$$

Hence $\underline{pred}_D$ maps every node x in V_D to the node from which x is directly derived (or to $\mathcal{O}_D$ if x was already present in X_0).

The <u>history</u> $\underline{hist}_D(x)$ <u>of a node</u> $x \in V_D$ <u>in</u> D is the sequence $(y_0, y_1, \ldots, y_m)$, $m \ge 1$, $y_i \in V_D$ for all i, $1 \le i \le m$, such that $y_0 = \mathcal{O}_D$, $y_m = x$, and $y_i = \underline{pred}_D(y_{i+1})$ for all i, $0 \le i \le m-1$.

Finally, we denote the set of nodes in V_{X_n} which are derived from a node $x \in V_D$ by $\underline{targ}_D(x)$, i.e., $\underline{targ}_D(x) = \{y \in V_{X_n} \mid x \in \underline{hist}_D(y)\}$. (For a sequence $s = (y_0, y_1, \ldots, y_m)$ we write $x \in s$ if there is an i, $0 \le i \le m$, such that $x = y_i$.)

For basic properties of concrete derivations we refer the reader to Rozenberg & Welzl (1984).

3. SUBGRAPHS

Given a graph language L, one often considers (various types of) subgraphs of graphs from L. In this section we consider "subgraph taking operations" for the case of BNLC languages. In particular, we demonstrate that the set of all induced subgraphs of the graphs from a BNLC language is again a BNLC language, while this is not the case when one considers the set of (arbitrary) subgraphs.

THEOREM 3.1. The set of all induced subgraphs of the graphs from a BNLC language is again a BNLC language.

Proof. Let $G = (\Sigma,\Delta,P,\underline{conn},Z)$ be a normalized BNLC grammar. Consider now the BNLC grammar $G' = (\Sigma,\Delta,P',\underline{conn},Z)$, where

$$P' = \{(d,Y') \mid Y' \text{ is an induced subgraph of some } Y \text{ with } (d,Y) \in P\}.$$

It is easily seen that L(G') is the set of all induced subgraphs of graphs in L(G). Since each BNLC language can be generated by a normalized BNLC grammar, the result follows. □

THEOREM 3.2. There exists a BNLC language L such that the set of all subgraphs of the graphs in L is not a BNLC language.

Proof. By Example 1, the set of all complete graphs with all nodes labeled by b, say, is a BNLC language. Obviously, the set of all subgraphs of these graphs is the set of all graphs labeled by b. However, it follows from Rozenberg & Welzl (1984, Theorem 5.2) that this is not a BNLC language. □

Let Δ be a set of labels. The Δ-projection, $\underline{proj}_\Delta(X)$, of a graph X is the subgraph of X which is induced by nodes of X with labels from Δ. (For example if $\underline{lab}(X) \cap \Delta = \emptyset$, then $\underline{proj}_{\Delta}(X) = \lambda$ and if $\underline{lab}(X) \subseteq \Delta$, then $\underline{proj}_{\Delta}(X) = X$). For a graph language L, we define $\underline{proj}_{\Delta}(L) = \{\underline{proj}_{\Delta}(X) \mid X \in L\}$.

THEOREM 3.3. For a set of labels Δ and for a BNLC language L, the language $\underline{proj}_{\Delta}(L)$ is a BNLC language.

Proof. Let $G' = (\Sigma', \Delta', P', \underline{conn}', Z')$ be a BNLC grammar such that $L = L(G')$ We construct a BNLC grammar $G = (\Sigma, \Delta, P, \underline{conn}, Z)$ with $L(G) = \underline{proj}_{\Delta}(L(G')) = \underline{proj}_{\Delta}(L)$ as follows (without loss of generality we assume that $\Delta \cap \Gamma' = \emptyset$).

Let $\Sigma = \Delta \cup \Gamma'$. For $d \in \Sigma$, we set $\underline{conn}(d) = \underline{conn}'(d) \cap \Sigma$. $Z = \underline{proj}_{\Sigma}(Z')$. Finally, P is the set $\{(A, \underline{proj}_{\Sigma}(Y)) \mid (A,Y) \in P'\}$. It is easily seen that indeed $L(G) = \underline{proj}_{\Delta}(L)$ and so the theorem holds. □

4. K-COLORABLE GRAPHS

In this section we prove that for every positive integer k, the set of all k-colorable graphs from a BNLC language is again a BNLC language. For technical reasons we start by considering relabelings of BNLC languages.

In some of the subsequent proofs, when we want to show that a language L is a BNLC language, we construct first a BNLC grammar G and give a relabeling ρ, such that $L = \rho(L(G))$. However, in general it may happen that $\rho(L(G))$ is not even an NLC language.

THEOREM 4.1. There exists a BNLC language L and a relabeling ρ such that $\rho(L)$ is not an NLC language.

Proof. In Ehrenfeucht et al. (1984, Corollary 3.3) it has been demonstrated that for a label b, no infinite subset of the set L_0 of all cycles labeled by b is an NLC language. Consider now the BNLC language $L(G_2)$, where G_2 is the grammar of Example 2 from Section 2. Let ρ be the relabeling from $\{b,c\}$ to $\{b\}$ defined by $\rho(c) = \rho(b) = b$. Then $\rho(L(G_2)) = L_0$ and so $\rho(L(G_2))$ is not an NLC language. □

To cope with the situation indicated above we proceed as follows.

Let $G = (\Sigma,\Delta,P,\underline{conn},Z)$ be a BNLC grammar. A relabeling ρ from Σ is called G-applicable, if $\rho(\Gamma) \cap \rho(\Delta) = \emptyset$ and if, for all $d,d' \in \Sigma$, (i) $\rho(d) \in \rho(\underline{conn}(d'))$ implies $d \in \underline{conn}(d')$ and (ii) $\rho(d) = \rho(d')$ implies $\rho(\underline{conn}(d)) = \rho(\underline{conn}(d'))$.

If ρ is G-applicable, then the ρ-image of G, denoted by $\rho(G)$, is the grammar $(\rho(\Sigma),\rho(\Delta),\rho(P),\underline{conn}_\rho,\rho(Z))$, where $\rho(P) = \{(\rho(d),\rho(Y)) \mid (d,Y) \in P\}$ and, for all $d \in \Sigma$, $\underline{conn}_\rho(\rho(d)) = \rho(\underline{conn}(d))$.

Note that $\underline{conn}_\rho$ is well-defined on the whole set $\rho(\Sigma)$, because of condition (ii) above on a G-applicable relabeling. Moreover, note that $\rho(G)$ is a BNLC grammar, because G is a BNLC grammar and because $\rho(\Gamma) \cap \rho(\Delta) = \emptyset$.

LEMMA 4.2. Let G be a BNLC grammar and let ρ be a G-applicable relabeling. Then $\rho(L(G)) \subseteq L(\rho(G))$.

Proof. Let $G = (\Sigma,\Delta,P,\underline{conn},Z_{ax})$. Consider a concrete derivation step $X \Rightarrow_{(x,Y)} Z$ in G, where X is a Γ-boundary graph. We will show that $\rho(X) \Rightarrow_{(x,\rho(Y))} \rho(Z)$ is a concrete derivation step in $\rho(G)$. This proves that $\rho(S(G)) \subseteq S(\rho(G))$ and, consequently, $\rho(L(G)) \subseteq L(\rho(G))$.

Since $\varphi_{\rho(X)}(x) = \rho(\varphi_X(x))$, we have $(\varphi_{\rho(X)}(x),\rho(Y)) \in [\rho(P)]$. The crucial point is to prove that

$$(*) \quad \{\{z,y\} | z \in \underline{neigh}_X(x), y \in V_Y, \text{ and } \varphi_X(z) \in \underline{conn}(\varphi_Y(y))\} =$$
$$\{\{z,y\} | z \in \underline{neigh}_{\rho(X)}(x), y \in V_{\rho(Y)} \text{ and } \varphi_{\rho(X)}(z) \in \underline{conn}_\rho(\varphi_{\rho(Y)}(y))\}.$$

Observe that (i) $\underline{neigh}_{\rho(X)}(x) = \underline{neigh}_X(x)$, (ii) $V_Y = V_{\rho(Y)}$, and (iii), for all d, $d' \in \Sigma$, $d \in \underline{conn}(d')$ if and only if $\rho(d) \in \underline{conn}_\rho(\rho(d')) = \rho(\underline{conn}(d'))$. Since $\varphi_{\rho(X)}(z) = \rho(\varphi_X(z))$ and $\varphi_{\rho(Y)}(y) = \rho(\varphi_Y(y))$ this proves the equality (*) and so the lemma holds. □

LEMMA 4.3. Let $G = (\Sigma,\Delta,P,\underline{conn},Z_{ax})$ be a BNLC grammar and let ρ be a G-applicable relabeling of Σ, such that ρ is the identity on Γ. Then $\rho(L(G)) = L(\rho(G))$.

Proof. Let $X' \Rightarrow_{(x,Y')} Z'$ be a concrete derivation step in $\rho(G)$. Consider now a graph X over Σ with $\rho(X) = X'$ and let $(d,Y) \in [P]$ be such that $\rho(d) = \varphi_{X'}(x)$ and $\rho(Y) = Y'$. Since $\rho(d) = d$, we have $\varphi_X(x) = d$ and so there is a graph Z such that $X \Rightarrow_{(x,Y)} Z$ is a concrete derivation step in G. Since Z is unique, it follows from the previous proof that $\rho(Z) = Z'$. Consequently, for every concrete derivation of a graph Z' from $\rho(Z_{ax})$ in $\rho(G)$ there exists a concrete derivation of a graph Z from Z_{ax} in G such that $\rho(Z) = Z'$. This proves that $L(\rho(G)) \subseteq \rho(L(G))$. This,in turn, implies (by Lemma 4.2) the equality of $L(\rho(G))$ and $\rho(L(G))$. □

For a natural number k, a k-coloring of a graph X is a function from V_X into $\{1,2,\dots,k\}$. A k-coloring of X is proper, if it assignes different "colors" to adjacent nodes in X. A graph X is k-colorable, if there is a proper k-coloring of X.

THEOREM 4.4. Let k be a natural number and let L be a BNLC language. The set of all k-colorable graphs from L is again a BNLC language.

Proof. Let $G = (\Sigma,\Delta,P,\text{conn},Z)$ be a normalized BNLC grammar with $L = L(G)$. (Such a normalized grammar always exists.)

The basic idea behind our construction is that a coloring is "guessed" during a derivation and during the same derivation it is checked whether the coloring is proper.

Let L_0 be the set of k-colorable graphs from L. First we define a BNLC grammar $G' = (\Sigma',\Delta',P',\text{conn}',Z')$ and a relabeling ρ such that $\rho(L(G')) = L_0$.

Let $\Delta' = \Delta \times \{1,2,\dots,k\}$, $\Gamma' = \{A_r \mid A \in \Gamma, r \subset \Delta'\}$ and let $\Sigma' = \Gamma' \cup \Delta'$. We use the relabeling ρ from Σ' to Σ defined by: $\rho(A_r) = A$ for $A_r \in \Gamma'$ and $\rho((a,i)) = a$ for $(a,i) \in \Delta'$.

The connection function conn' is defined by $\text{conn}'(d) = \rho^{-1}(\text{conn}(\rho(d)))$ for all $d \in \Sigma'$. If A is the label of the unique node of Z, then Z' is a one node graph labeled by $A_\emptyset$.

P' consists of all productions $(A_r,Y), A_r \in \Gamma'$ and $Y \in G_{\Sigma'}$, satisfying the following conditions:

(i) $(\rho(A_r),\rho(Y)) \in P$.

(ii) Let y be a nonterminal node in Y with $\varphi_Y(y) = B_s \in \Gamma'$. Then $s = (r \cap \text{conn}'(B_s)) \cup \text{cont}_Y(y)$.

(iii) If $\varphi_Y(y_1) = (a,i) \in \Delta'$ and $\varphi_Y(y_2) = (b,j) \in \Delta'$ for two adjacent nodes y_1 and y_2 of Y, then $i \neq j$.

(iv) If, for some $y \in V_Y$, $\varphi_Y(y) = (a,i) \in \Delta'$, then $\text{conn}'((a,i)) \cap r \cap \{(b,i) \mid b \in \Delta\} = \emptyset$.

From condition (ii) it follows that G' is context consistent with context describing function η', where $\eta'(A_r) = r$ for all $A_r \in \Gamma'$. Hence, it is easily seen that conditions (iii) and (iv) together imply that if (a,i) and (b,j) appear as labels of two adjacent nodes of a graph from S(G'), then $i \neq j$. That is, all graphs in L(G') are k-colorable.

It is straightforward to see, that ρ is G'-applicable. Consequently, $\rho(G')$ is defined and (by Lemma 4.2) $\rho(L(G')) \subseteq L(\rho(G'))$. Consider $\rho(G') = (\rho(\Sigma'), \rho(\Delta'), \rho(P'), \underline{\text{conn}}_\rho, \rho(Z'))$. Then $\rho(\Sigma') = \Sigma$, $\rho(\Delta') = \Delta$, $\rho(P') \subseteq P$, $\underline{\text{conn}}'_\rho = \underline{\text{conn}}$, and $\rho(Z') = Z$. Thus it is easily seen that $L(\rho(G')) \subseteq L(G)$, that is we have shown that $\rho(L(G')) \subseteq L(G)$.

It is left to show that $\rho(L(G'))$ contains <u>all</u> k-colorable graphs in L(G). Let X be a graph from L(G) for which there exists a proper k-coloring α. Consider a concrete derivation in G of X from $X_0 \in [Z]$:

$$D: X_0 \Rightarrow_{(x_0,Y_1)} X_1 \Rightarrow_{(x_1,Y_2)} X_2 \cdots \Rightarrow_{(x_{n-1},Y_n)} X_n = X.$$

We define a labeling φ' of V_D. For $x \in V_D$ with $\varphi_D(x) = a \in \Delta$, $\varphi'(x) = (a,\alpha(x))$, and for $x \in V_D$ with $\varphi_D(x) = A \in \Gamma$, $\varphi'(x) = A_r$, where $r = \{\varphi'(y) \mid \{x,y\} \in E_D\}$. This labeling φ' induces a concrete derivation D' in G' in the obvious way. Note in particular that, for $x,y \in V_D = V_{D'}$, $\varphi'(x) \in \underline{\text{conn}}'(\varphi'(y))$ if and only if $\varphi_D(x) = \rho(\varphi'(x)) \in \rho(\underline{\text{conn}}'(\varphi'(y)) = \underline{\text{conn}}(\rho(\varphi'(y))) = \underline{\text{conn}}(\varphi_D(y))$. Obviously, for the resulting graph X'_n of D', $\rho(X'_n) = X_n = X$, which implies that $\rho(L(G'))$ contains all k-colorable graphs from L(G). Thus we have shown that $L_0 = \rho(L(G'))$.

Let now ρ' be the relabeling from Σ' to $\Gamma' \cup \Delta$, where ρ' is the identity on Γ' and ρ' equals ρ on Δ'. Then ρ' is G'-applicable, $\rho'(G')$ exists, and we can apply Lemma 4.3 to conclude that $\rho'(L(G')) = L(\rho'(G'))$. Obviously $\rho'(L(G')) = \rho(L(G'))$ and so $L_0 = L(\rho'(G'))$, which proves that L_0 is a BNLC language. □

We get now the following easy corollaries of Theorem 4.4.

COROLLARY 4.5. It is decidable whether or not L(G) contains a k-colorable graph, where k is a natural number and G is a BNLC grammar.

Proof. In the proof of Theorem 4.4, we gave an effective construction of a BNLC grammar G' which generates all k-colorable graphs in L(G). Since it is easily seen that the emptiness problem for NLC grammars is decidable, the result follows.□

COROLLARY 4.6. It is decidable whether or not L(G) contains a totally disconnected graph, where G is a BNLC grammar.

Proof. This follows from Corollary 4.5 and from the easy observation, that a graph is totally disconnected if and only if it is 1-colorable. □

It is instructive to compare the above result with the situation for general NLC grammars. It has been shown in Janssens & Rozenberg (1981, Theorem 3) that it is undecidable whether or not L(G) contains a totally disconnected graph, where G is an NLC grammar.

COROLLARY 4.7. Let d be a natural number and let L be a (u-)BNLC language. For (unlabeled) graphs $X \in L$ of maximal degree $\leq d$, and for a natural number k, it is decidable in polynomial time whether or not X is k-colorable.

Proof. We consider the labeled case, i.e., L is a BNLC language. If $k > d$, then a graph X of maximal degree $\leq d$ is always k-colorable, which makes the problem trivial. Hence we may assume that $k \leq d$ and k is fixed (because d is a constant).

A graph X is k-colorable if and only if each of its connected components is k-colorable. Let now L_0 be the set of induced subgraphs from L and let L_1 be the set of all k-colorable graphs from L_0. Now a graph X in L is k-colorable if and only if its connected components are in L_1. That is, since L_1 is a BNLC language (see Theorems 3.1 and 4.4) we have (polynomial time) reduced the problem "X k-colorable?" for $X \in L$ to the membership problem of connected graphs in a BNLC language. This has been shown to be decidable in polynomial time for graphs of bounded fixed degree in Rozenberg & Welzl (1984 , Theorem 6.3) and so the result holds. Since this membership problem is also decidable in polynomial time for a u-BNLC language, the result can be shown analogously for a u-BNLC language L. □

To put the above result in a proper perspective, let us recall that the problem whether a graph of maximal degree ≤ 4 is 3-colorable is NP-complete (see Garey et al., 1976, Theorem 2.3).

5. CONNECTED AND DISCONNECTED GRAPHS

Let X be a graph and let x and y be nodes in V_X. A walk from x to y in X is a sequence $(x_0, x_1, \ldots, x_m), m \geq 0$, of nodes in V_X such that $x_0 = x$, $x_m = y$, and $\{x_i, x_{i+1}\} \in E_X$ for all i, $0 \leq i \leq m-1$. A graph X is connected, if, for every pair of nodes x and y in V_X, there is a walk from x to y. A graph X is disconnected if it is not connected.

THEOREM 5.1. The set of all disconnected graphs from a BNLC language is again a BNLC language.

Proof. We use here the following easy observation. A graph X is disconnected if and only if there is a 2-coloring of X such that (i) 1 is assigned to at least one node and 2 is assigned to at least one node and (ii) adjacent nodes have the same color assigned to.

Let $G = (\Sigma, \Delta, P, \underline{conn}, Z)$ be a normalized BNLC grammar. First we construct a BNLC grammar $G' = (\Sigma', \Delta', P', \underline{conn}', Z')$ and we give a relabeling ρ such that $\rho(L(G'))$ consists of all disconnected graphs in L(G).

We set $\Delta' = \Delta \times \{1,2\}$, $\Gamma' = \Gamma \times 2^{\Delta'} \times 2^{\{1,2\}}$, and $\Sigma' = \Delta' \cup \Gamma'$. The relabeling ρ is defined by: $\rho((a,i)) = a$ for $(a,i) \in \Delta'$, and $\rho((A,r,t)) = A$ for $(A,r,t) \in \Gamma'$.

For $d \in \Sigma'$, $\underline{conn}'(d) = \rho^{-1}\underline{conn}(\rho(d))$. If A is the label of the unique node of Z, then Z' is a graph consisting of one node labeled by $(A, \emptyset, \{1,2\})$.

P' consists of all productions $((A,r,t),Y)$ which satisfy the following conditions.

(i) $(A, \rho(Y)) \in P$.

(ii) Let $y \in V_Y$ with $\varphi_Y(y) = (B,s,u) \in \Gamma'$. Then $s = (r \cap \underline{conn}'((B,s,u))) \cup \underline{cont}_Y(y)$.

(iii) If $\varphi_Y(y_1) = (a,i) \in \Delta'$ and $\varphi_Y(y_2) = (b,j) \in \Delta'$ for two adjacent nodes y_1 and y_2 of Y, then $i = j$.

(iv) If, for some $y \in V_Y$, $\varphi_Y(y) = (a,i) \in \Delta'$, then $\underline{conn}'((a,i)) \cap r \cap \{(b,j) \in \Delta' \mid i \neq j\} = \emptyset$.

(v) If $i \in t$, then either there is a terminal node y in Y such that $\varphi_Y(y) = (a,i)$ for some $a \in \Delta$, or there is a nonterminal node y in Y such that $\varphi_Y(y) = (B,s,u) \in \Gamma'$ and $i \in u$.

Using arguments similar to those from the proof of Theorem 4.4 one can show that (i) $\rho(L(G'))$ consists of all disconnected graphs in L and that (ii) $\rho(L(G'))$ is a BNLC language. □

THEOREM 5.2. The set of all connected graphs from a BNLC language is again a BNLC language.

Proof. The problem we have to cope with in this proof is that being connected is not a "local property" of a graph (and so "local" techniques like the one for guessing a coloring as used in the proofs of Theorems 4.4 and 5.1 do not apply directly).

Let us consider the following example. Suppose that the axiom of a BNLC grammar derives a graph X as depticted in Fig. 5.1(a) . We apply now a production to the B-labeled node which disconnects the "direct" walk from its c-labeled neighbor to its f-labeled neighbor via the B-labeled node (see Fig. 5.1(b)). Now, whether or not the graph resulting from a further derivation "continuing this step" is connected, depends on the "behaviour" of the A-labeled node. This A-labeled node "does not know", whether the B-labeled node has disconnected the c-f path, and, moreover, the B-labeled node is not able to "send a message" to the A-labeled node. We will settle this problem by a "guessing mechanism", which, however, is essentially different from the ones we use in the proofs of Theorems 4.4 and 5.1. To every nonterminal node certain "connection tasks" are (nonderterministically) conferred. In our example, this would mean that we introduce labels $A_\emptyset$ and $A_{\{\{a,d\}\}}$ instead of A, and labels $B_\emptyset$ and $B_{\{\{c,f\}\}}$ instead of B. Then, e.g., $B_{\{\{c,f\}\}}$ stands for : whatever is derived

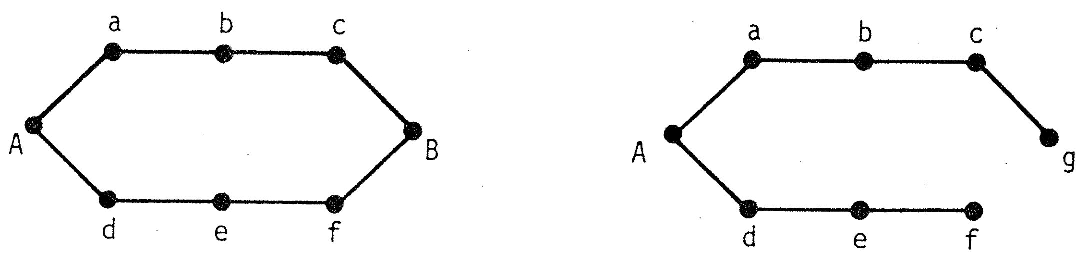

Fig. 5.1. "Breaking up" a walk in a derivation step.

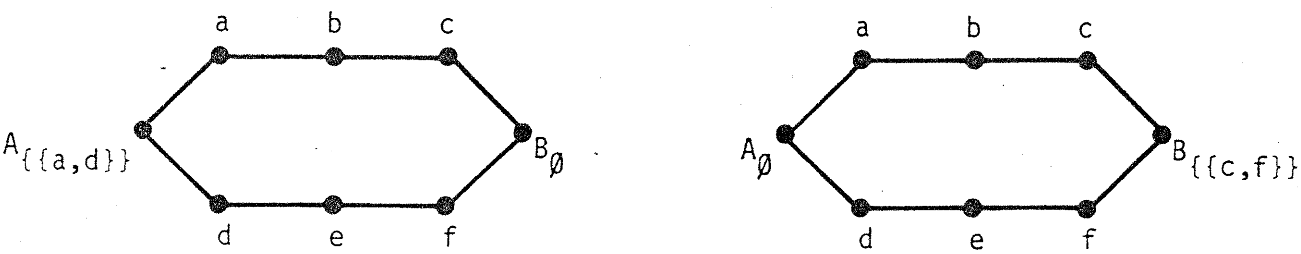

Fig. 5.2. "Secure" graphs derived instead of the graph in Fig. 5.1(a).

from a node labeled by this nonterminal, there must be a walk from each c-labeled neighbor of this node to each f-labeled neighbor of this node. This means that we actually derive two graphs X' and X" rather than X - these are shown in Figs.5.2(a) and (b).

We turn now to the formal proof. Let $G = (\Sigma,\Delta,P,\underline{conn},Z_{ax})$ be a normalized chain-free and context-consistent BNLC grammar with context describing function η (such a grammar exists for every BNLC language) and let L_c be the set of all connected graphs from L(G). The BNLC grammar $G' = (\Sigma',\Delta,P',\underline{conn'},Z'_{ax})$ such that $L(G') = L_c$ is now constructed as follows.

Let $\Gamma' = \{A_r | A \in \Gamma,\ r \subseteq \{\{a,b\} | a,b \in \eta(A)\}\}$ and let $\Sigma' = \Gamma' \cup \Delta$. (Note that for A_r from Γ', the set r might contain sets of cardinality 1.) The relabeling ρ from Σ' to Σ is defined by $\rho(A_r) = A$, for $A_r \in \Gamma'$ and $\rho(a) = a$, for $a \in \Delta$.

For all $d \in \Sigma'$, we set conn'(d) = conn(ρ(d)). (Note that conn(ρ(d)) = ρ^{-1} conn(ρ(d)), because conn(d) $\subseteq \Delta$.) If A is the label of the unique node of Z_{ax}, then Z'_{ax} is a graph consisting of one node labeled by $A_\emptyset$.

In order to define the set of productions P' we need the following notions.

Let Y be a Γ'-boundary graph. A walk $(x_1,x_2,\ldots,x_n)$ from x_1 to x_n in Y is a secure walk from x_1 to x_n in Y if, for $2 \le i \le n-1$, $\varphi_Y(x_i) = A_r \in \Gamma'$ implies that $\{\varphi_Y(x_{i-1}), \varphi_Y(x_{i+1})\} \in r$. (This intuitively means that whenever, on the walk from x_1 to x_n, we are passing a nonterminal node x_i, $2 \le i \le n-1$, then we can "rely" on x_i in the sense that if x_i is replaced by a graph, then, in the resulting graph, there will still be a possibility of getting from x_{i-1} to x_{i+1}.)

A Γ'-boundary graph Y is now called securely connected if it is connected and if there is a secure walk from x to y in Y for all pairs of terminal nodes $x,y \in V_Y$. (Note that a graph labeled by terminals only is securely connected if and only if it is connected.)

Let Y be a Γ'-boundary graph and let a and b be (not necessarily distinct) terminal labels. A walk $(x_1,x_2,\ldots,x_n)$ from x_1 to x_n is a secure (a,b)-connecting walk from x_1 to x_n in Y, if (i) it is a secure walk from x_1 to x_n, (ii) $a \in$ conn'($\varphi_Y(x_1)$) and $b \in$ conn'($\varphi_Y(x_n)$), (iii) if $\varphi_Y(x_1) = A_r \in \Gamma'$, then $\{a,\varphi_Y(x_2)\} \in r$ (or $\{a,b\} \in r$, if $n = 1$) and (iv) if $\varphi_Y(x_n) = B_s \in \Gamma'$, then $\{\varphi_Y(x_{n-1}),b\} \in s$ (or $\{a,b\} \in s$, if $n = 1$). (This intuitively means that if, in a graph X, an a-labeled node z_1 and a b-labeled node z_2 are connected via a nonterminal node z which is replaced by Y, then -in the resulting graph - there is a secure walk from z_1 to z_2.)

A Γ'-boundary graph Y is {a,b}-connecting, if there are nodes x and y in V_Y for which there is a secure (a,b)-connecting walk from x to y in Y or a (b,a)-connecting walk from x to y in Y.

Let $s \subseteq \Delta$. A Γ'-boundary graph Y is externally connected by s, if, for every node x of Y, there exists a secure walk $(x_1,x_2,\ldots,x_n)$ from some node x_1 to x in Y, such that: conn'($\varphi_Y(x_1)$) $\cap\, s \neq \emptyset$ and if $\varphi_Y(x_1) = A_r \in \Gamma'$ and $n \ge 2$, for some $a \in \underline{\text{conn}}'(\varphi_Y(x_1)) \cap s$. (This intuitively means that if a node z with context s is replaced by Y, then, in the resulting graph, there is a secure walk from every node x in Y to a neighbor of z.)

Now we are ready to define the set of productions P':

$$P' = \{(A_r, Y) \mid \eta(A) = \emptyset,\ Y \text{ is securely connected and } (A, \rho(Y)) \in P\} \cup$$
$$\cup \{(A_r, Y) \mid \eta(A) \neq \emptyset,\ Y \text{ is } \{a,b\}\text{-connecting for all } \{a,b\} \in r,\ Y \text{ is externally connected by } \eta(A), \text{ and } (A, \rho(Y)) \in P\}.$$

We will prove now that indeed $L(G') = L_c$.

Claim 1. All graphs in S(G') are securely connected.

Proof of Claim 1. Clearly the axiom of G' and every graph directly derived from the axiom is securely connected. Moreover, it is also clear, that no graph in S(G') except for the axiom has a nonterminal node labeled by A_r where $\eta(A) = \emptyset$. (Recall that G is chain-free!)

Let X be a securely connected graph in $S(G') - [Z'_{ax}]$ and let us consider a concrete derivation step $X \Rightarrow_{(x,Y)} Z$ in G'. We will show that Z is securely connected which implies the claim.

Let $\varphi_X(x) = A_r$ and let y be an arbitrary node of Y. Since Y is externally connected by $\eta(A) = \underline{\text{cont}}_X(x) \neq \emptyset$, there is a secure walk from y to a node $x' \in \underline{\text{neigh}}_X(x)$.

Now, let x_1 and x_2 be two nodes of $X-x$ such that there is a secure walk $(x_1, \bar{x}, x_2)$ in X. If $\bar{x} \neq x$, then $(x_1, \bar{x}, x_2)$ is a secure walk in Z. If $\bar{x} = x$, then x_1 and x_2 are terminal nodes, where $\{\varphi_X(x_1), \varphi_X(x_2)\} \in r$. Consequently, Y is $\{\varphi_X(x_1), \varphi_X(x_2)\}$-connecting and so there is a secure walk $(x_1, y_1, y_2, \ldots, y_k, x_2)$, $k \geq 1$, from x_1 to x_2 in Z with $y_i \in V_Y$ for all i, $1 \leq i \leq k$. This actually shows that every secure walk from a node z_1 of $X-x$ to a node z_2 of $X-x$ in X can be transformed to a secure walk from z_1 to z_2 in Z, simply by inserting a "corresponding" walk in Y wherever x occurs.

It remains to show that there is a secure walk from every y_1 of Y to every y_2 of Y in Z. Such a walk can be obtained by concatenating a secure walk from y_1 to a node $x_1 \in$ $\underline{neigh}_X(x)$ in Z with a secure walk from a node $x_2 \in$ $\underline{neigh}_X(x)$ to y_2 in Z using a secure walk from x_1 to x_2 in Z. Note that the concatenation of such secure walks always yields a secure walk in Z, because the "connecting" nodes x_1 and x_2 are terminal nodes. Thus Claim 1 holds.

Claim 2. $L(G') \subseteq L_c$.

Proof of Claim 2. Obviously, ρ is G'-applicable. It is easily seen that $L(\rho(G')) \subseteq L(G)$. By Lemma 4.2, we have $\rho(L(G')) \subseteq L(\rho(G'))$. Since $\rho(L(G')) = L(G')$, we also have $L(G') \subseteq L(G)$. Claim 1 implies that all graphs in $L(G')$ are connected. Hence $L(G') \subseteq L_c$ and the Claim holds.

Claim 3. $L_c \subseteq L(G')$.

Proof of Claim 3. Consider a concrete derivation D of a connected graph X from $X_0 \in [Z_{ax}]$ in G,

$$D: X_0 \Rightarrow_{(x_0,Y_1)} X_1 \Rightarrow_{(x_1,Y_2)} X_2 \cdots \Rightarrow_{(x_{n-1},Y_n)} X_n,$$

where $X = X_n$. We will show that there is a derivation D' of X from a graph $X_0' \in [Z_{ax}']$ in G'.

Let $x \in V_D - V_X$, where $\varphi_D(x) = A \in \Gamma$. Then we assign to x a new label $\varphi'(x) = A_r$, where r is the set of all sets $\{a,b\} \subseteq \Delta$ with the following property:

There are nodes $x_1, x_2 \in \underline{neigh}_D(x)$, (i.e., $\{x_1,x\} \in E_D$ and $\{x_2,x\} \in E_D$) with $\varphi_D(x_1) = a$, $\varphi_D(x_2) = b$ and there is a walk $(x_1,z_1,z_2,\ldots,z_m,x_2)$, $m \geq 1$, from x_1 to x_2 in X, such that $z_i \in \underline{targ}_D(x)$ for all i, $1 \leq i \leq m$.

Note that, if $x_1', x_2' \in \underline{neigh}_D(x)$ with $\varphi_D(x_1') = a$ and $\varphi_D(x_2') = b$, then $(x_1',z_1,z_2,\ldots,z_m,x_2')$ is a walk from x_1' to x_2' in X.) This mapping φ', extended by $\varphi'(x) = \varphi_D(x)$, for $x \in V_X$, is a new labeling of V_D. This labeling yields in the obvious way a "primed" sequence,

$$D': X'_0 \Rightarrow_{(x_0,Y'_1)} X'_1 \Rightarrow_{(x_1,Y'_2)} X'_2 \cdots \Rightarrow_{(x_{n-1},Y'_n)} X'_n ,$$

by setting $V_{X'_i} = V_{X_i}$, $E_{X'_i} = E_{X_i}$, and $\varphi_{X'_i}$ equal the restriction of φ' to $V_{X'_i}$ for all i, $0 \leq i \leq n$, (and setting analogously each Y'_i for i, $1 \leq i \leq n$). We will show that D' is a concrete derivation in G'. Since $X'_0 \in [Z'_{ax}]$ and $X_n = X$, this implies that $X \in L(G')$.

It is easily seen that, for all i, $0 \leq i \leq n-1$, $X'_1 \Rightarrow_{(x_i,Y'_{i+1})} X'_{i+1}$ is a concrete derivation step in G', provided that $(\varphi_{X'_{i+1}}(x_i),Y'_{i+1}) \in [P']$ (recall the proof of Lemma 4.3).

Clearly, $(\rho(\varphi_{X'_i}(x_i)),\rho(Y'_{i+1})) \in [P]$. Thus, it remains to show that

(i) Y'_1 is securely connected,

and for all i, $1 \leq i \leq n-1$, if $A_r = \varphi_{X'_i}(x_i)$, then

(ii) Y'_{i+1} is $\{a,b\}$-connecting for all $\{a,b\} \in r$, and

(iii) Y'_{i+1} is externally connected by $\eta(A)$.

(Note that, because G is chain-free and $i > 0$, the set $r \neq \emptyset$.)

To demonstrate (i), let $(y_1,y_2,\ldots,y_k)$, $k \geq 2$, be a walk from y_1 to y_k in X, such that $y_1,y_k \in V_{Y'_1}$ and $y_i \notin V_{Y'_i}$ for all i, $2 \leq i \leq k-1$. If $k = 2$, then (y_1,y_2) is a secure walk in Y'_1. If $k > 2$, then there is a node $z \in V_{Y'_1}$ with $z \in \underline{hist}_D(y_i)$ for all i, $2 \leq i \leq k-1$. (Suppose there are distinct nonterminal nodes z_1 and z_2 in Y'_1 such that $z_1 \in \underline{hist}_D(y_i)$ and $z_2 \in hist_D(y_{i+1})$ for some i, $2 \leq i \leq k-2$. Then z_1 and z_2 are not adjacent, and so y_i and y_{i+1} are not adjacent, a contradiction.) This implies that (by definition of φ'), for $\varphi_{Y'_1}(z) = A_r$, $\{\varphi_{Y'_1}(y_1), \varphi_{Y'_1}(y_k)\} \in r$ holds. Hence (y_1,z,y_k) is a secure walk in Y'_1. Thus every walk from a terminal node $y \in V_{Y'_1}$ to a terminal node $y' \in V_{Y'_1}$ in X can be transformed to a secure walk from y to y' in Y'_1, simply by replacing subsequences of nodes not in $V_{Y'_1}$ by the "corresponding" nonterminal nodes in $V_{Y'_1}$. Obviously Y'_1 must be connected. This proves that Y'_1 is securely connected.

Conditions (ii) and (iii) can be proved analogously by transforming walks in X to secure walks in Y_{i+1} for all i, $2 \leq i \leq n-1$. Hence the claim holds.

The theorem follows now from Claims 2 and 3. □

Using the reasoning analogous to the one from the proof of Corollary 4.5, we get the following decidability results.

COROLLARY 5.3. It is decidable whether of not (i) L(G) contains a connected graph, (ii) L(G) contains a disconnected graph, where G is a BNLC grammar. □

This result should be compared with the situation for general NLC grammars. In particular, it is undecidable whether or not L(G) contains a connected graph, where G is an NLC grammar (see Janssens & Rozenberg,1981, Theorem 8).

Note, however, that our (effective) closure properties for BNLC languages do not only entail a number of decidability and complexity results which differ significantly from those known for NLC grammars (as we have seen in Corollaries 4.5, 4.6, 4.7, and 5.3). We can use these results also to confer combinatorial properties of BNLC languages to subsets of BNLC languages defined by graph theoretic squeezing mechanisms. For example, (by a correspondence to context-free string grammars,) it is easily seen that an NLC language L has always a semilinear "size-set", i.e., the set $\{\#X \mid X \in L\}$ is semilinear (see e.g. Ginsburg, 1966, for the definition of a semilinear set). However, the set of connected graphs in an NLC language has possibly a size set which is not semilinear, as it can be easily deduced from a result in Janssens & Rozenberg (1980b, Theorem 9). For example, there exists an NLC language L such that $\{\#X \mid X \in L$ and X is connected$\}$ is exactly the set of prime numbers (see Janssens & Rozenberg, 1980b, Theorem 9 and Salomaa, 1973, Example I.2.5). By Theorem 5.2, the set of connected graphs in a BNLC language has a semilinear size set.

6. GRAPHS WITH SUBGRAPHS HOMEOMORHPIC TO A GIVEN GRAPH AND NONPLANAR GRAPHS

Let Z be an unlabeled graph. A subdivision of an edge $\{x,y\}$ in E_Z is the replacement of $\{x,y\}$ by a node z (not in V_Z) together with edges $\{x,z\}$ and $\{z,y\}$. An unlabeled graph X is homeomorphic to an unlabeled graph Z if X can be obtained from Z by a sequence of subdivisions of edges. A graph X is homeomorphic to an unlabeled graph Z, if und(X) is homeomorphic to Z. (Recall here the remark about our definition of "homeomorphic" given at the beginning of Section 1.)

The goal of this section is to show that, for a given graph Z, the set of all graphs from a BNLC language which have a subgraph homeomorphic to Z forms again a BNLC language. Consequently, the Kuratowsky theorem implies that the set of all nonplanar graphs from a BNLC language is again a BNLC language.

LEMMA 6.1. Let X and Z be unlabeled graphs. X has a subgraph homeomorphic to Z, if and only if there is a function f from V_X into $V_Z \cup E_Z \cup \{\$\}$ with the following properties.

(1) For every $z \in V_Z$, there is exactly one $x \in V_X$ with $f(x) = z$.

(2) For every $\{z_1,z_2\} \in E_Z$ and every $x \in V_X$ with $f(x) = z_1$, there is exactly one neighbor x' of x with $f(x') \in \{\{z_1,z_2\},z_2\}$.

(3) If, for a node $x \in V_X$, $f(x) = \{z_1,z_2\} \in E_Z$, then there are exactly two neighbors x' and x" of x in X, such that $f(x'), f(x") \in \{z_1,z_2,\{z_1,z_2\}\}$.

Proof. Let f be a function from V_X satisfying the statement of the lemma. Then it is easily seen that for all $\{z_1,z_2\} \in E_Z$, there is a unique path $x_1,x_2,\dots,x_k$, $k \geq 2$ from the unique node $x_1 \in V_X$ with $f(x_1) = z_1$ to the unique node $x_k \in V_X$ with $f(x_k) = z_2$, such that $f(x_i) = \{z_1,z_2\}$, for all i, $2 \leq i \leq k-1$. Hence, using f one can find a subgraph X' homeomorphic to Z. Note however, that not all nodes $x \in V_X$ with $f(x) \neq \$$ must appear in such a subgraph X' - there might be cycles in X with their nodes having some $\{z_1,z_2\} \in E_Z$ assigned to by f which do not belong to this subgraph X'.

On the other hand, let X' be a subgraph of X homeomorphic to Z. Without loss of generality, we may assume that X' is minimal in the sense that $\#X'' \geq \#X'$ for every subgraph X" of X homeomorphic to Z. Hence there is a function g from V_Z into $V_{X'}$ such that for all $\{z_1,z_2\} \in E_Z$ there is a corresponding path from $g(z_1)$ to $g(z_2)$ in X', and for different edges in E_Z these corresponding paths are disjoint (except perhaps from their start - and endpoints). Moreover, the minimality of X' implies that only consecutive nodes on these paths are adjacent in X.

Let us fix such a collection of (pairwise disjoint) paths. Now it is rather easy to define a function f from V_X into $V_Z \cup E_Z \cup \{\$\}$ which fulfills conditions (1) through (3) from the statement of the lemma:

- for all nodes x, for which there is a $z \in V_Z$ with $g(z) = x$ we set $f(x) = z$,
- for a node $x \in V_X$ on a path (from the chosen collection) corresponding to the edge $\{z_1,z_2\} \in E_Z$, we set $f(x) = \{z_1,z_2\}$ (unless x is an end- or start-point of this path), and, finally,
- we assign \$ to the remaining nodes in X.

Thus the lemma holds. □

THEOREM 6.2. Let Z be an unlabeled graph. The set of all graphs from a BNLC language having a subgraph homeomorphic to Z is again a BNLC language.

Proof (sketch). The proof of this theorem uses Lemma 6.1. and ideas similar to those from the proofs of Theorems 4.4 and 5.1. Let $G = (\Sigma,\Delta,P,\underline{conn},Z_{ax})$ be a normalized BNLC grammar. Then we define a new set of terminals by $\Delta' = \Delta \times (V_Z \cup E_Z \cup \{\$\})$ and a relabeling ρ from Δ' to Δ, where $\rho((a,u)) = a$ for $a \in \Delta$ and $u \in V_Z \cup E_Z \cup \{\$\}$.

Consider now $G_1 = (\Sigma_1,\Delta',P_1,\underline{conn}_1,Z_1)$, where $\Sigma_1 = \Gamma \times 2^{V_Z} \cup \Delta'$. Let ρ_1 be the relabeling from Σ_1 to Σ such that ρ_1 equals ρ on Δ' and $\rho_1((A,U)) = A$, for $A \in \Gamma$ and $U \subseteq V_Z$. For $(A,U) \in \Gamma_1$, a production $((A,U),Y) \in P_1$ if and only if:

(i) $(A,\rho_1(Y)) \in P$, and

(ii) $z \in U$ if and only if either there is exactly one node $y \in V_Y$ with $\varphi_Y(y) = (a,z)$, for some $a \in \Delta$, or there is exactly one node $y \in V_Y$ with $\varphi_Y(y) = (B,U')$ for some $B \in \Gamma$ and some $U' \subseteq V_Z$ such that $z \in U'$.

If A is the label of Z_{ax}, then Z_1 is a graph consisting of one node labeled by (A,V_Z). Finally, we set $\underline{conn}_1 = \rho_1^{-1}\underline{conn}\rho_1$. It is not too difficult to see that $L(G_1)$ consists of all graphs $X \in G_{\Delta'}$ such that $\rho(X) \in L(G)$, where for each $z \in V_Z$ there is exactly one node x in X with $\varphi_X(x) = (a,z)$ for some $a \in \Delta$. Thus we have guessed a function f and then we have "filtered out" all graphs satisfying condition (1) from the statement of Lemma 6.1.

The next step is to filter out all those graphs from $L(G_1)$ which satisfy condition (2) from the statement of Lemma 6.1. To this aim we choose an edge $\{z_1,z_2\} \in E_Z$ and an element of $\{z_1,z_2\}$, say z_1.

Now, rather than to provide tedious formal details we give the intuition underlying our construction. To every nonterminal in Γ_1 we append a second component which contains the information about the labels in the neighborhood and, moreover, this component "notes" whether a label occurs once, twice or more often. In addition, a flag (0 or 1) is added which says whether this nonterminal is responsible for establishing the unique edge between the unique node labeled by (a,z_1) for some $a \in \Delta$, and a node labeled by (b,u) for some $b \in \Delta$ and $u \in \{z_2,\{z_1,z_2\}\}$. Obviously, this information can be kept finite and it is sufficient for choosing those productions which fulfill the appropriate task. Although this description is not very detailed we think that it provides enough "hints" for the construction of a grammar which generates all those graphs in $L(G_1)$, where the unique terminal node with z_1 in the second component of its label has a unique neighbor with z_2 or $\{z_1,z_2\}$ in the second component of its label.

The same procedure has to be performed successively for all $\{z',z''\} \in E_Z$ and $z \in \{z',z''\}$.

Analogous techniques can be used for assuring condition (3) from the statement of Lemma 6.1. Thus we obtain a BNLC grammar $G' = (\Sigma',\Delta',P',\underline{conn}',Z')$ such that $\rho(L(G'))$consists of all graphs in L(G) having a subgraph homeomorphic to Z.

Let ρ' be the relabeling from Σ' to $\Gamma' \cup \Delta$ such that ρ' is the identidy on Γ' and ρ' equals ρ on Δ'. If one follows the construction of G', then it appears that Lemma 4.3 can be applied, i.e.,$\rho'(G')$ generates exactly $\rho'(L(G')) = \rho(L(G'))$ which completes this informal reasoning behind a possible proof of the theorem. □

The following corollaries follow now easily from the above theorem.

COROLLARY 6.3. The set of all nonplanar graphs from a BNLC language is again a BNLC language. □

COROLLARY 6.4. It is decidable whether or not L(G) contains a nonplanar graph, where G is a BNLC grammar. □

COROLLARY 6.5. Let Z be an unlabeled graph, let k be a positive integer and let L be a BNLC language. Then for a graph of maximal degree ≤ k in L it is decidable in polynomial time whether it has a subgraph homeomorphic to Z. □

7. GRAPHS WITH FORBIDDEN COMPLETE SUBGRAPHS

In the preceding section, we considered graphs (from a BNLC language) which have certain substructures. In this section we consider the opposite case, i.e., we are interested in graphs which do not contain certain subgraphs. In particular, we show that, for a given k, the set of all graphs from a BNLC language that do not contain a complete subgraph on k nodes is again a BNLC language. The more general problem, i.e., "Given a graph Z, is the set of all graphs from a BNLC language that do not contain an induced subgraph isomorphic to Z again a BNLC language?" is open.

THEOREM 7.1. Let k be a positive integer. The set of all graphs from a BNLC language which have no complete subgraphs on k nodes is again a BNLC language.

Proof (sketch). Let $G = (\Sigma, \Delta, P, \underline{conn}, Z)$ be a normalized BNLC grammar.

Let D be a concrete derivation in G of a graph $X \in L(G)$ from a graph $X_0 \in [Z]$ and let x and y be two nonterminal nodes in an intermediate graph X' in D. Since x and y are not adjacent, a complete subgraph of X cannot have nodes both in $\underline{targ}_D(x)$ and in $\underline{targ}_D(y)$. That is, if a complete subgraph contains a node in $\underline{targ}_D(x)$, then it is built up entirely from nodes in $\underline{targ}_D(x) \cup \underline{neigh}_{X'}(x)$. This observation allows one to construct a BNLC grammar $G' = (\Sigma', \Delta, P', \underline{conn}', Z')$ which generates the set of all graphs from L(G) with no complete subgraph on k nodes, using a "local control" of nonterminal nodes over their neighborhoods.

To this end, let $\Delta = \{a_1, a_2, \ldots, a_n\}$, $n = \#\Delta$, be an enumeration of Δ, let $K = \{0,1,\ldots,k-1\}$, and let $\Gamma' = \Gamma \times 2^{(K^n)}$. Intuitively, for $(A,r) \in \Gamma'$, $(i_1, i_2, \ldots, i_n) \in r$ stands for: there is a complete subgraph on $(i_1+i_2+\ldots+i_n)$ nodes in the graph induced by the neighborhood of the (A,r) labeled node, where i_1 nodes from this complete subgraph are a_1-labeled, i_2 are a_2-labeled, ..., and i_n are a_n-labeled.

Consider now a production $(A,Y) \in P$ and a set $r \subseteq K^n$. Although rather tedious to define formally down to the last detail, it is now intuitively easy to see how a corresponding production $((A,r),Y')$ in P' looks like. (i) One has to verify that no complete graph on k nodes is produced by the neighborhood of the replaced node (which is sufficiently described by r) or by the terminal labeled nodes in Y (otherwise there is no corresponding production $((A,r),Y')$ in P'). (ii) Once (i) is satisfied, for every nonterminal node from Y the second component of its label in Y' has to be calculated from r and from its neighborhood in Y (the first component is the "old" label in Y).

Z' and conn' have to be chosen in the obvious way. This completes the informal sketch of the construction underlying the proof of the theorem. □

8. DISCUSSION

This paper has continued a systematic investigation of BNLC grammars and languages initiated in Rozenberg & Welzl (1984). We believe that these two papers together indicate that the family of BNLC languages is an attractive subfamily of the family of NLC languages. In particular, in this paper we have demonstrated that very often considerations of the from - "Consider only those graphs from a BNLC language that satisfy a particular property" - do not lead out of the BNLC family. This certainly makes the family of BNLC grammars mathematically attractive to work with and allows one to prove a number of properties which are either not true or (at this stage) not known for the general family of NLC languages.

Clearly, a number of open problems and problem areas should still be investigated before one gets good insight into the basic properties of BNLC grammars. For example,

(1) We have shown that the set of all nonplanar graphs from a BNLC language is again a BNLC language. What about planar graphs?

(2) Is the set of hamiltonian (nonhamiltonian) graphs from a BNLC language again a BNLC language? In general, it would be interesting to find nontrivial graph properties which - applied to BNLC languages as squeezing mechanisms - lead out of the family of BNLC languages.

In addition to concrete open problems as outlined above, one should treat a number of topics not "touched" until now. In particular, one should investigate combinatorial properties of BNLC languages. A typical question we have in mind is of the following type: "If a BNLC language satisfies a combinatorial property P_1 then it must satisfy also property P_2", where P_1 and P_2 should be "interesting" graph theoretic properties.

We are currently investigating a number of problems of this type and we hope to report on this research in the near future.

Aknowledgements: The first author acknowledges the support by NSF Grant MCS 83-05245. Research of the second author has been partially supported by the Austrian "Fonds zur Förderung der wissenschaftlichen Forschung".

REFERENCES

Bondy, J.A. & Murty, U.S.R. (1976), "Graph Theory with Applications", Macmillan & Co., London.

Brandenburg, F.J. (1983), On the complexity of the membership problem for graph grammars, in "Proceedings of the WG 83" (Nagl, M. & Perl, J., eds.), Universitätsverlag Trauner, Linz, pp. 40-49.

Ehrenfeucht, A., Main, M.G. & Rozenberg, G. (1984), Restrictions on NLC grammars, Theoret. Comput. Sci. 31, pp. 211-223.

Ehrig, H. (1979), Introduction to the algebraic theory of graph grammars, Lecture Notes in Computer Science 73, pp. 1-69.

Garey, M.R. & Johnson, D.S. (1979), "Computers and Intractability - A Guide to the Theory of NP-completeness", Freeman, San Francisco.

Garey, M.R., Johnson, D.S. & Stockmeyer, L. (1976), Some simplified NP-complete graph problems, Theoret. Comput. Sci. 1, pp. 237-267.

Ginsburg, S. (1966) "The Mathematical Theory of Context-Free languages", McGraw-Hill, Inc., New York.

Gurari, E.H. & Sudborough, I.H. (1982), Improved dynamic programming algorithms for bandwidth minimization and the min-cut arrangement problem, to appear in J. Algorithms.

Harary, F. (1969), "Graph Theory", Addison Wesley, Reading, Massachusetts.

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, pp. 191-216.

Janssens, D. & Rozenberg, G. (1980b), Restrictions, extensions, and variations of NLC grammars, Inform. Sci. 20, pp. 217-244.

Janssens, D. & Rozenberg, G. (1981), Decision problems for node label controlled graph grammars, J. Comp. System Sci. 22, pp. 144-174.

Janssens, D., Rozenberg, G. & Welzl, E. (1984), The bounded degree problem for NLC grammars is decidable, Institute of Applied Mathematics and Computer Science, University of Leiden, Report no. 84-05.

Nagl, M. (1979), A tutorial and bibliographical survey of graph grammars, Lecture Notes in Computer Science 73, pp. 70-126.

Rozenberg, G. & Welzl., E. (1984), Boundary NLC graph grammars - basic definitions, normal forms and complexity, Institute of Applied Mathematics and

Computer Science, University of Leiden, Report no. 84-29

Salomaa, A. (1973), "Formal Languages", Academic Press, London.

Saxe, J.B. (1980), Dynamic programming algorithms for recognizing small band-width graphs in polynomial time, SIAM J. Algebraic Discrete Methods 1, pp. 363-369.

Turán, Gy. (1983), On the complexity of graph grammars, Acta Cybernet. 6, 271-281.