COMBINATORIAL PROPERTIES OF BOUNDARY NLC GRAPH LANGUAGES

by

G. Rozenberg and E. Welzl

CU-CS-295-85 April, 1985

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

Rep.no. 84-31

COMBINATORIAL PROPERTIES OF BOUNDARY NLC GRAPH LANGUAGES

by

Grzegorz Rozenberg & 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.

In this paper we investigate the behaviour of graph invariants within BNLC languages. First we demonstrate that there is a dependency between the chromatic number and the clique number of graphs in BNLC langugages (while this is well-known not to be true for arbitrary graph languages). Secondly, we introduce a new graph invariant, the so-called index of a graph which seems to be very suitable for describing the adjacency structure of a graph. Then we prove that every BNLC language is of bounded index (which is shown not to be true for arbitrary graph languages). Thus we exhibit properties (concerning graph invariants) of BNLC languages which are intrinsic to this class. We use them to demonstrate that certain graph languages are not BNLC languages.

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, 1984a). 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 (1984a,b) 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). Moreover, the family of BNLC languages turns out to be closed under many of the operations of taking of a BNLC language all graph satisfying a certain graph property (e.g. being k-colorable, being connected, or being nonplanar).

In this paper we continue this research and we concentrate on combinatorial properties of BNLC languages. In particular, we investigate the behaviour of graph invariants within BNLC languages. First we demonstrate that there is a dependency between the chromatic number and the clique number of graphs in BNLC languages (while this is well-known not to be true for arbitrary graph languages). For example, we show that a BNLC language is of bounded chromatic number if and only if it is of bounded clique number. Secondly, we introduce a new graph invariant, the so-called <u>index</u> of a graph which seems to be very suitable for describing the adjacency structure of a graph. Then we prove that every BNLC language is of bounded index (which is shown not to be true for arbitrary graph languages). Thus we exhibit properties (concerning graph invariants) of BNLC languages which are intrinsic to this class. We use them to demonstrate that certain graph languages are not BNLC languages. For example, we prove that among all graphs in a BNLC languages (i) there is only a finite number of Mycielski graphs (see Mycielski, 1955) and (ii) there is only a finite number of square grid graphs.

The paper is organized as follows. General notions concerning graphs and graph grammars are recalled in Section 1. In Section 2 we recall basic notions and properties concerning BNLC grammars. In Section 3 we consider the relationship between the chromatic number and the clique number of graphs in a BNLC language. In Section 4 we introduce the notion of the index of a graph and we demonstrate that every BNLC language is of bounded index. Finally, a discussion in Section 5 concludes this 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, e.g. in the scope of Harary (1969).

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 $\mathcal{G}_\Sigma$.

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, $\text{neigh}_X(x)$, is the set $\{y \in V_X \mid \{x,y\} \in E_X\}$. 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., $\#X = \#V_X$. 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 $\{\text{und}(X) \mid X \in L\}$. For a label a, $\text{proj}_a(X)$ denotes the subgraph of X which is induced by the a-labeled nodes in X. For a set of graphs L, $\text{proj}_a(L) = \{\text{proj}_a(X) \mid X \in L\}$.

Graph Grammars

A node label controlled (NLC) grammar is a system $G = (\Sigma, \Delta, P, \text{conn}, Z_{ax})$, 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 $Y \in \mathcal{G}_\Sigma$

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

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

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

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 $\varphi_X(x') \in conn(\varphi_Y(y))$.)

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

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

A graph language L is an NLC language 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 an unlabeled BNLC (u-BNLC) language, if there is a BNLC language L' such that $L = \underline{und}(L')$.(Recall that we set implicitely $\Sigma - \Delta = \Gamma$.)

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

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

PROPOSITION 2.3. For every BNLC language L there is a normalized BNLC grammar G such that $L(G) = L - \{\lambda\}$, where λ is the empty graph.

In what follows we consider two graph languages to be <u>equal</u> if they coincide up to the empty graph.

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

<u>Concrete derivations</u>

Let $G = (\Sigma,\Delta,P,\underline{conn},Z_{ax})$ 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 $X \Rrightarrow_{(x,Y)} Z$ as a <u>concrete derivation step in</u> G (<u>from</u> X <u>to</u> Z).

A sequence of "successive" concrete derivation steps in G

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

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 <u>concrete derivation in</u> G (<u>from</u> X_0 <u>to</u> X_n).

The <u>node set</u> of D is $V_D = \bigcup_{i=0}^{n} V_{X_i}$. The <u>edge set</u> of D is $E_D = \bigcup_{i=0}^{n} E_{X_i}$. The <u>labeling function</u> 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 $x \in V_D$

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

Hence $\underline{\text{pred}}_D$ 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> $\underline{\text{hist}}_D(x)$ <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 $y_i = \underline{\text{pred}}_D(y_{i+1})$ 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 $\underline{\text{targ}}_D(x)$, i.e., $\underline{\text{targ}}_D(x) = \{y \in V_{X_n} \mid x \in \underline{\text{hist}}_D(y)\}$. (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 $x = y_i$.)

If D is understood, then we omit the inscription D in $\underline{\text{pred}}_D$, $\underline{\text{hist}}_D$, and $\underline{\text{targ}}_D$.

We recall now some basic properties of concrete derivations in BNLC grammars.

PROPOSITION 2.4. Let $G = (\Sigma,\Delta,P,\underline{\text{conn}},Z)$ be a BNLC grammar and let D be a concrete derivation in G from a Γ-boundary graph X_0 to a graph X. Then

(1) If $\{x,y\} \in E_D$, then at least one of the relations $\underline{\text{pred}}(x) \in \underline{\text{hist}}(y)$ or $\underline{\text{pred}}(y) \in \underline{\text{hist}}(x)$ holds.

(2) Let $\underline{\text{hist}}(y) = (y_0,y_1,\ldots,y_k)$, $k \ge 1$, and let $x \in V_D$ be such that $\underline{\text{pred}}(x) = y_\ell$, for some ℓ, $0 \le \ell \le k-1$. If $\{x,y\} \in E_D$ then $\{x,y_{\ell+1}\} \in E_D$ and $\varphi_D(x) \in \underline{\text{conn}}(\varphi_D(y_i))$ for all i, $\ell+2 \le i \le k$.

(3) Let $\{x,y\} \notin E_D$ and let $x' \in \underline{\text{targ}}(x)$ and $y' \in \underline{\text{targ}}(y)$. Then $\{x',y'\} \notin E_D$. □

3. CHROMATIC NUMBER VERSUS CLIQUE NUMBER IN BNLC LANGUAGES

Let X be a graph and let n be a positive integer. An n-coloring of X is a function from V_X into $\{1,2,\ldots,n\}$. An n-coloring of X is called proper, if it assignes different "colors" to adjacent nodes in X. The chromatic number, $\chi(X)$, of X is the minimum n for which there exists a proper n-coloring of X. The clique number, $\omega(X)$, of X is the maximum n such that there is a complete subgraph of X with n nodes.

A graph language L is of bounded chromatic number (of bounded clique number) if there is a positive integer k such that $\chi(X) \le k$ ($\omega(X) \le k$, respectively) for all $X \in L$.

On the one hand, it is clear that $\omega(X) \le \chi(X)$ holds for every graph X. Hence, a graph language of bounded chromatic number is also of bounded clique number. On the other hand, there are graphs with "arbitrary small" clique number and "arbitrary large" chromatic number - this result was proved in Mycielski (1955) and it is formally stated as follows.

PROPOSITION 3.1. For every pair of integers n and m with $2 \le n \le m$, there is a graph X such that $\omega(X) \le n$ and $\chi(X) \ge m$.

For example, triangle-free unlabeled graphs with arbitrary high chromatic number can be constructed as follows. Let M_3 be the cycle of length 5. For $i \ge 4$, M_i is obtained from M_{i-1} by (i) adding to every node x in M_i a node x' which is adjacent to all neighbors of x and (ii) adding an additional node y which is adjacent to all "new" nodes. Then $\omega(M_i) = 2$ and $\chi(M_i) = i$ for all i, $i \ge 3$ (see Mycielski, 1955, or also Bondy & Murty, 1976). The graphs M_i, $i \ge 3$, are called Mycielski graphs.

In this section we show that such an independence between chromatic number and clique number cannot exist within a BNLC language. More precisely, we will demonstrate that for every BNLC language L and every positive integer n, there is an integer m, such that, for all $X \in L$, $\omega(X) \leq n$ implies $\chi(X) \leq m$. This shows, e.g. that a u-BNLC language cannot contain an infinite number of Mycielski graphs.

First we state a lemma which is easy to prove.

LEMMA 3.2. A graph language $L \subseteq G_\Delta$ (where Δ is a finite set of labels) is of bounded chromatic number (of bounded clique number) if and only if, for all $a \in \Delta$, $\underline{proj}_a(L)$ is of bounded chromatic number (of bounded clique number, respectively). □

Now we are ready to prove the key theorem of the section.

THEOREM 3.3. A BNLC language is of bounded chromatic number if and only if it is of bounded clique number.

Proof. Since $\omega(X) \leq \chi(X)$ for every graph X, the "only if part" of the theorem holds.

To prove the "if part" we proceed as follows. Let L be a BNLC language of bounded clique number over a set of labels Δ. By Lemma 3.2 it suffices to show that, for all $a \in \Delta$, $\underline{proj}_a(L)$ is of bounded chromatic number. Let a be an arbitrary but fixed label from Δ. By Rozenberg & Welzl (1984b, Theorem 3.3), there is a normalized BNLC grammar $G = (\Sigma,\{a\},P,\underline{conn},Z_{ax})$ such that $L(G) = \underline{proj}_a(L)$. Clearly, we may assume that G is reduced, i.e., for each label $A \in \Gamma$ there are graphs X and X' such that $Z_{ax} \overset{*}{\Rightarrow} X' \overset{*}{\Rightarrow} X$, $X \in L(G)$ and $A \in \underline{lab}(X')$. We consider now seperately two cases.

Case 1: $a \notin \underline{conn}(a)$.

Then every graph in L(G) consists of connected components each having no more than $\underline{maxr}(G)$ nodes. Hence, $\chi(X) \leq \underline{maxr}(G)$ holds for all $X \in L(G)$. This settles the first case.

Case 2: $a \in \underline{conn}(a)$.

Here we proceed as follows. First we prove a number of consequences (claims) of the fact that L(G) is of bounded clique number. These properties allow us to define for each graph $X \in L(G)$ a $2\cdot\#\Gamma$-coloring (based on a derivation of X in G) which is "almost" proper. Finally, we point out how this coloring of X can be extended to a proper $2\cdot\#\Gamma\cdot\underline{maxr}(G)$-coloring of X.

Let $C(\Gamma) = \{A \in \Gamma \mid a \in \underline{conn}(A)\}$. For $A,B \in \Gamma$, we write $A \sim B$ if $A,B \in C(\Gamma)$ and there is a production $p = (A,Y) \in P$ such that $B \in \underline{lab}(Y)$; we say then that p transfers A to B. $\overset{*}{\sim}$ is the reflexive and transitive closure of $\sim$. We say that a sequence of productions $p_1,p_2,\ldots,p_k$, $k \geq 0$, gradually transfers A to B, if either $k = 0$ and $A = B$ or $k \geq 1$ and there are labels $A_0,A_1,\ldots,A_k$ such that $A = A_0$, $B = A_k$ and p_i transfers A_{i-1} to A_i for all i, $1 \leq i \leq k$.

Moreover, we write $A \overset{\approx}{\sim} B$, if $A,B \in C(\Gamma)$ and there is a production $p = (A,Y) \in P$ such that there are nodes $x,y \in V_Y$ with $\varphi_Y(x) = a$, $\varphi_Y(y) = B$, and $\{x,y\} \in E_Y$; we say then that p productively transfers A to B.

Obviously, $A \overset{\approx}{\sim} B$ implies $A \sim B$. We will demonstrate that $A \overset{\approx}{\sim} B$ excludes $B \overset{*}{\sim} A$.

Claim 1. If $A \overset{\approx}{\sim} B$, then $B \overset{*}{\sim} A$ does not hold.

Proof of Claim 1. Assume to the contrary that $A \overset{\approx}{\sim} B$ and $B \overset{*}{\sim} A$ hold for some $A,B \in C(\Gamma)$. Let m be an arbitrary positive integer. Then the following procedure leads to a graph whose clique number exceeds m.

(1) Derive from Z_{ax} a graph containing an A-labeled node z_0. Let $i = 1$.

(2) Apply to z_{i-1} a production which productively transfers A to B. Let x_i, y_i be two (fixed) nodes derived in this step such that x_i is adjacent to y_i, x_i is labeled by a, and y_i is labeled by B.
(Note that x_i and y_i are adjacent to all nodes x_j, $j = 1,2,\ldots,i-1$.)

(3) Apply now to y_i a sequence of productions which gradually transfers B to A (in such a way that always a node derived in the previous step is replaced in the next one). Let z_i be an A-labeled node obtained in the last step. If the applied sequence was empty, let $z_i = y_i$.
(Note that z_i is adjacent to all x_j, $j = 1,2,\ldots,i$.)

(4) If $i \le m$, then let $i = i+1$ and go back to step (2).

(5) Apply "terminating sequences" of productions to all nonterminal nodes (i.e., derive a terminally labeled graph from every nonterminal node).

It is easily seen that $\{x_1,x_2,\ldots,x_{m+1}\}$ induces a complete subgraph of the graph obtained by the above procedure. Since m was chosen arbitrarily, this contradicts the fact that L(G) is of bounded clique number. Hence the assumption that $A \overset{*}{\sim} B$ and $B \overset{*}{\sim} A$ hold is false and the claim follows.

Consider now a concrete derivation D of a graph $X \in L(G)$ from a graph $X_0 \in [Z_{aX}]$. For $x,y \in V_X$, we write $x \to y$, if $\{x,y\} \in E_D$, pred(x) $\in$ hist(y) and pred(y) $\notin$ hist(x). We write $x \leftrightarrow y$ if $\{x,y\} \in E_D$ and pred(x) = pred(y) (i.e., pred(x) $\in$ hist(y) and pred(y) $\in$ hist(x)). Note that for all $\{x,y\} \in E_X$, exactly one of the relations $x \to y$, $y \to x$ or $x \leftrightarrow y$ holds (see Proposition 2.4(1)).

Claim 2. Let $x,y,z \in V_X$ be such that $x \to z$, $y \to z$ and φ_D(pred(x)) = φ_D(pred(y)) = φ_D(pred_D(z)). Then pred(x) = pred(y).

Proof of Claim 2. Consider hist(z) = $(z_0,z_1,\ldots,z_k)$, $k \ge 1$. There must be indices i,j such that pred(x) = z_i and pred(y) = z_j.

(i) Assume that $i < j$. Clearly, $a \in$ conn($\varphi_D(z_\ell)$) for all ℓ, $i+2 \le \ell \le k$ (see Proposition 2.4(2)). Thus, in particular, $a \in$ conn($\varphi_D(z_{j+1})$). Moreover,

$a \in \underline{\text{conn}}(\varphi_D(z_j))$. If $j \geq i+2$, then this is straightforward. If $j = i+1$, then, since $z_j = \underline{\text{pred}}(y)$, we have $\varphi_D(z_j) = \varphi_D(z_{k-1})$ and so indeed $a \in \underline{\text{conn}}(\varphi_D(z_j))$. Hence, all labels $\varphi_D(z_j), \varphi_D(z_{j+1}), \ldots, \varphi_D(z_{k-1})$ are from $C(\Gamma)$. Since y and z_{j+1} are produced in the same derivation step (i.e., $\underline{\text{pred}}(y) = \underline{\text{pred}}(z_{j+1})$) and since $\{y, z_{j+1}\} \in E_D$ (otherwise $\{y,z\} \notin E_D$ which contradicts $y \to z$), we have $\varphi_D(z_j) \gtrsim \varphi_D(z_{j+1})$. But $\varphi_D(z_{j+1}) \overset{*}{\sim} \varphi_D(z_{k-1}) = \varphi_D(z_j)$ which is a contradiction to Claim 1.

(ii) Analogously, assuming that $i > j$ leads to a contradiction. Consequently, $i = j$ and so Claim 2 holds.

Claim 3. Let $x,y,z \in V_X$, $x \to y$, $\underline{\text{pred}}(y) = \underline{\text{pred}}(z)$. Then $x \to z$ holds.

Proof of Claim 3. Since y and z have the same label and both are derived in the same derivation step, the claim holds.

For $A \in \Gamma$, let $V_X^A = \{x \in V_X \mid \varphi_D(\underline{\text{pred}}(x)) = A\}$. The function α_A from V_X^A to $\{0,1\}$ is inductively defined for $x \in V_X^A$ as follows:

$$\alpha_A(x) = \begin{cases} 0 & \text{if there is no node } y \in V_X^A \text{ with } y \to x \text{ or if, for all nodes} \\ & y \in V_X^A \text{ with } y \to x,\ \alpha_A(y) = 1 \text{ is already defined,} \\ 1 & \text{if, there is at least one node } y \in V_X^A \text{ with } y \to x \text{ and, for} \\ & \text{all nodes } y \in V_X^A \text{ with } y \to x,\ \alpha_A(y) = 0 \text{ is already defined.} \end{cases}$$

Claim 4. α_A is well-defined on the whole set V_X^A.

Proof of Claim 4. Assume that α_A is not defined on the whole set V_X^A. Then there are nodes $x,y,z \in V_X^A$ such that $y \to x$, $z \to x$, and $\alpha_A(y) = 1$ and $\alpha_A(z) = 0$ are defined. Recall that $\varphi_D(\underline{\text{pred}}(x)) = \varphi_D(\underline{\text{pred}}(y)) = \varphi_D(\underline{\text{pred}}(z)) = A$ and hence it follows by Claim 2 that $\underline{\text{pred}}(y) = \underline{\text{pred}}(z)$. Since $\alpha_A(y) = 1$, there is a node $x' \in V_X^A$ such that $x' \to y$ and $\alpha_A(x') = 0$ is already defined. However, by Claim 3 also $x' \to z$ holds which is a contradiction to the fact that $\alpha_A(z) = 0$. Hence the claim holds.

Claim 5. Let $x,y \in V_X^A$. If $\alpha_A(x) = \alpha_A(y)$ and $\{x,y\} \in E_X$, then $x \leftrightarrow y$.

Proof of Claim 5. If $x \leftrightarrow y$ does not hold, then either $x \to y$ or $y \to x$ holds. In either case this would imply $\alpha_A(x) \neq \alpha_A(y)$. Hence the claim follows.

Let $\{A_1,A_2,\dots,A_s\}$, $s = \#\Gamma$, be an arbitrary but fixed enumeration of all elements from Γ. We define a 2s-coloring α of X as follows

$$\alpha(x) = \alpha_{A_i}(x) \cdot s + i, \text{ if } x \in V_X^{A_i} .$$

It is obvious from Claim 5 that if $\alpha(x) = \alpha(y)$ and $\{x,y\} \in E_X$, then $x \leftrightarrow y$, that is, pred(x) = pred(y). Hence, for all j, $1 \le j \le 2s$, the subgraph of X induced by the nodes x with $\alpha(x) = j$ has connected components of maximal size maxr(G) (this was meant by "almost" proper). Consequently, the coloring α can be easily extended to a proper $2s\cdot$maxr(G)-coloring of X. Hence the theorem holds. □

As a matter of fact one gets the following functional dependency between the clique number and the chromatic number of graphs from a BNLC language.

THEOREM 3.4. For very BNLC language L there is a positive integer function f_L such that $\chi(X) \le f_L(\omega(X))$ for all $X \in L$.

Proof. For a positive integer n, let $L^{(n)} = \{X \in L \mid \omega(X) \le n\}$. Then $L^{(n)}$ is a BNLC language (see Rozenberg & Welzl, 1984b, Theorem 7.1). Hence, by Theorem 3.3, $L^{(n)}$ is of bounded chromatic number. Thus, if we set $f_L(n) = \max(\{1\} \cup \{\chi(X) \mid X \in L^{(n)}\})$ for all n, $n \ge 1$, then f_L is a well defined positive integer function which satisfies the statement of the theorem. □

Thoerems 3.3 and 3.4 can be used to prove that certain graph languages are not BNLC languages.

COROLLARY 3.5. The set of triangle-free unlabeled graphs (i.e, graphs with clique number at most 2) is not a u-BNLC language. □

Even stronger, we can conclude that there are sets of graphs which have a finite intersection with every BNLC language.

COROLLARY 3.6. Every u-BNLC language contains only a finite number of Mycielski-graphs. □

We conclude this section with a decidability result which can be proved using Theorem 3.3: it is decidable whether or not L(G) is of bounded chromatic number for an arbitrary BNLC grammar G. First we need the following lemma.

LEMMA 3.7. There exists an algorithm which, given an arbitrary BNLC grammar G, yields a BNLC grammar G_c such that $L(G_c)$ is the set of all complete subgraphs of graphs from L(G).

Proof. Let L = L(G), where G is an arbitrary BNLC grammar. Then we can effectively construct a BNLC grammar $G' = (\Sigma', \Delta', P', \underline{conn}', Z')$ which generates the set of induced subgraphs of graphs from L(G) (see Rozenberg & Welzl, 1984b, Theorem 3.1).

A BNLC grammar $G'' = (\Sigma'', \Delta'', P'', \underline{conn}'', Z'')$ is called context consistent, if there is a function η from Γ'' into $2^{\Delta''}$ such that, for every $X \in S(G'')$ and every nonterminal node $x \in V_X$, $\eta(\varphi_X(x)) = \{\varphi_X(y) \mid y \in \underline{neigh}_X(x)\}$ holds. That is, for $A \in \Gamma''$, $\eta(A)$ is the set of labels which occur in the neighborhood of a node (in any graph from S(G'')) labeled by A. From Rozenberg & Welzl (1984a, Theorem 3.2) it follows that we can construct a normalized context consistent BNLC grammar $G'' = (\Sigma'', \Delta'', P'', \underline{conn}'', Z'')$ with $L(G'') = L(G')$. Let η be the "context describing" function of G".

Obviously, the set of all complete subgraphs of graphs from L is exactly the set of all complete graphs from L(G").

Now it is not too difficult to see that the set of all complete graphs from L(G") is generated by the BNLC grammar $G_c = (\Sigma_c, \Delta_c, P_c, \underline{conn}_c, Z_c)$, where $\Sigma_c = \Sigma''$, $\Delta_c = \Delta''$, $\underline{conn}_c = \underline{conn}''$, $Z_c = Z''$, and

$P_c = \{(A,Y) \in P'' \mid \eta(A) \subseteq \underline{\text{conn}}''(d)$, for all $d \in \underline{\text{lab}}(Y)$, and Y is a complete graph$\}$.

Hence the lemma holds. □

THEOREM 3.8. It is decidable whether or not (i) L(G) is of bounded clique number, (ii) L(G) is of bounded chromatic number, where G is an arbitrary BNLC grammar.

Proof. Let G be an BNLC grammar. L(G) is of bounded clique number if and only if the set of complete subgraphs of graphs from L(G) is finite. By Lemma 3.7, a BNLC grammar G_c generating this set can be effectively constructed. Since the finiteness problem is decidable even for NLC grammars, assertion (i) follows. By Theorem 3.3, assertion (ii) follows directly from (i). □

4. INDEX IN BNLC LANGUAGES

In the previous section we considered the relation between two well-known graph invariants in a BNLC language. In this section we introduce a new graph invariant, the so-called index of a graph, which describes a significant part of the restriction put on BNLC languages (by their generating grammars).

Let X be a graph and let $U \subseteq V_X$. Two nodes x and y in $V_X - U$ are U-equivalent, written $x \sim_U y$, if they have the same neighborhood in U, i.e., $\underline{neigh}_X(x) \cap U = \underline{neigh}_X(y) \cap U$. Clearly, $\sim_U$ is an equivalence relation.

The index of X relative to U, denoted by $\underline{index}_U(X)$, is the number of equivalence classes of $\sim_U$ on $V_X - U$.

The index of X, denoted by $\underline{index}(X)$, is defined by

$$\underline{index}(X) = \underline{min}\ \{\underline{index}_U(X) \mid U \subseteq V_X,\ \text{where } \#X/4 < \#U \leq \lceil \#X/2 \rceil\}.$$

The sub-index of X, denoted by $\underline{sindex}(X)$, is defined by

$$\underline{sindex}(X) = \underline{max}\{\underline{index}(Y) \mid Y \text{ is an induced subgraph of } X\}.$$

Remark 4.1. In order to indicate the bounds involved in the definition of index, we could have used the terminology $(1/4,1/2)$-$\underline{index}(X)$ rather than $\underline{index}(X)$. However, since these are the only bounds we consider, we omit this additional "prefix" in the notation. □

A graph language L is of bounded index (of bounded sub-index), if there is a positive integer k such that $\underline{index}(X) \leq k$ ($\underline{sindex}(X) \leq k$, respectively) for all $X \in L$.

We will show that every BNLC language is of bounded sub-index. First of all, we observe that "being of bounded sub-index" is a "nontrivial" property.

Example. Let k be a positive integer. The square grid graph S_k is the unlabeled graph defined by $V_{S_k} = \{(i,j) \mid 0 \leq i \leq k,\ 0 \leq j \leq k\}$ and

$E_{S_k} = \{\{(i,j),(i',j')\} | 1 = |i - i'| + |j - j'|\}$.

We claim that for $k \geq 1$ and $n = \#S_k$ $(=(k+1)^2)$, $\underline{\text{index}}(S_k) \geq \lfloor \sqrt{n}/4 \rfloor$. This can be shown as follows.

Let $U \subseteq V_{S_k}$ be such that $n/4 \leq \#U \leq \lceil n/2 \rceil$, and let $\bar{U} = V_{S_k} - U$. Then there are at least $\lfloor \sqrt{n} \rfloor$ nodes on the frontier of the subgraph induced by $\bar{U}$ (a node in $\bar{U}$ is on its frontier, if it has a neighbor in U). At most four of the nodes on the frontier of $\bar{U}$ can be U-equivalent (note that every node in S_k has at most four neighbors). Consequently, $\underline{\text{index}}_U(S_k) \geq \lfloor \sqrt{n} \rfloor /4 \geq \lfloor \sqrt{n}/4 \rfloor$ and the above assertion holds. Of course,the bound is far from "optimal"; however, it suffices for our purpose. □

Next we prove a basic property (as regards index) of derivations in normalized BNLC grammars.

LEMMA 4.1. Let $G = (\Sigma,\Delta,P,\underline{\text{conn}},Z)$ be a normalized BNLC grammar and let D be a concrete derivation in G of a graph $X \in L(G)$ from a graph $X_0 \in [Z]$. Then there are nodes $y_1,y_2,\ldots,y_k$, $k \geq 1$, in V_D, such that for $U = \bigcup_{i=1}^{k} \underline{\text{targ}}(y_i)$, (a) $\#X/4 < \#U \leq \lceil \#X/2 \rceil$, and (b) $\underline{\text{index}}_U(X) \leq \#\Delta + \underline{\text{maxr}}(G)$.

Proof. Clearly the assertion holds for X with $\#X = 1$. Let $\#X \geq 2$. Let $\bar{x}$ be a node in V_D, such that $\#\underline{\text{targ}}(\bar{x}) > \lceil \#X/2 \rceil$ and if $\underline{\text{pred}}(y) = \bar{x}$ for a $y \in V_D$, then $\#\underline{\text{targ}}(y) \leq \lceil \#X/2 \rceil$. Obviously, such a node exists.

Let now $y_1,y_2,\ldots,y_\ell$, $\ell \geq 1$ be an enumeration of all nodes y with $\underline{\text{pred}}(y) = \bar{x}$, where (i) $\#\underline{\text{targ}}(y_i) \geq \#\underline{\text{targ}}(y_{i+1})$, for all i, $1 \leq i \leq \ell-1$, and (ii) there is an ℓ_0, $0 \leq \ell_0 \leq \ell$ such that, for all i, $1 \leq i \leq \ell_0$, $\varphi_D(y_i) \in \Gamma$ and, for all i, $\ell_0 < i \leq \ell$, $\varphi_D(y_i) \in \Delta$. Then there is a k, $1 \leq k \leq \ell$, such that the union U of all sets $\underline{\text{targ}}(y_i)$, $1 \leq i \leq k$, satisfies the following two conditions:

(C1) $\#X/4 < \#U \leq \lceil \#X/2 \rceil$, and

(C2) either all nodes y_i, $1 \leq i \leq k$, are nonterminal nodes or all nodes y_i, $k < i \leq \ell$, are terminal nodes.

We show now that $\underline{\text{index}}_U(X) \leq \#\Delta + \underline{\text{maxr}}(G)$.

Let $W_0 = \{y \in V_X - \underline{\text{targ}}(\bar{x}) | \{\bar{x},y\} \notin E_D\}$. For all $y \in W_0$ and $z \in U$, $\{y,z\} \notin E_X$ (see Proposition 2.4(3)); consequently, $\underline{\text{neigh}}_X(y) \cap U = \emptyset$, and so all nodes in W_0 are U-equivalent.

For each $a \in \Delta$, let $W_a = \{y \in V_X - \underline{\text{targ}}(\bar{x}) | \varphi_X(y) = a, \{\bar{x},y\} \in E_D\}$. Then for all nodes $y,y' \in W_a$, $z \in U$, we have $\{y,z\} \in E_X$ if and only if $\{y',z\} \in E_X$. Consequently, for each $a \in \Delta$, all nodes in W_a are U-equivalent.

Let $W_0' = \{y \in \underline{\text{targ}}(\bar{x}) - U | \underline{\text{pred}}(y) \neq \bar{x}\}$. That is, W_0' contains the terminal nodes derived from nonterminal nodes in $T = \{y_i | k < i \leq \ell\}$. If T contains no nonterminal node, then W_0' is empty. If this is not the case, then no nonterminal node in T is adjacent to one of the nodes y_i, $1 \leq i \leq k$ (recall that in this case condition (C2) above implies that all nodes y_i, $1 \leq i \leq k$, are nonterminal nodes). Consequently, we have $\underline{\text{neigh}}_X(y) \cap U = \emptyset$ for $y \in W_0'$ and so all nodes in $W_0 \cup W_0'$ are U-equivalent.

For each terminal node y in T, let $W_y = \{y\}$. Note that there are at most $\underline{\text{maxr}}(G) - 1$ different sets W_y of this type.

Clearly, the sets $W_0 \cup W_0'$, W_a (for each $a \in \Delta$), and W_y (for each $y \in T \cap V_X$) cover the complement of U in V_X. The above reasoning shows that all nodes within each of these sets are U-equivalent. Since there are at most $\#\Delta + \underline{\text{maxr}}(G)$ different sets (one for $W_0 \cup W_0'$, $\#\Delta$ for all W_a with $a \in \Delta$, and $\underline{\text{maxr}}(G) - 1$ for all W_y with $y \in T \cap V_X$), we have shown that $\underline{\text{index}}_U(X) \leq \#\Delta + \underline{\text{maxr}}(G)$. Hence the statement of the lemma holds for our choice of $y_1, y_2, \ldots, y_k$. □

Lemma 4.1. immediately implies that every BNLC language is of bounded index. This result can be extended to sub-index in the following way.

THEOREM 4.2. Every BNLC language is of bounded sub-index.

Proof. It is easily seen that a graph language L is of bounded sub-index, if and only if the set L' of induced subgraphs of graphs from L is of bounded index. It is known that the set of all induced subgraphs of graphs from a BNLC language is again a BNLC language (see Rozenberg & Welzl, 1984b, Theorem 3.1). Since, by Lemma 4.1, every BNLC language is of bounded index, the theorem holds. □

It is instructive to notice that it is the "boundary" restriction on NLC grammars that yields the above property. It is well-known that, for a set of labels Δ, G_Δ is an NLC language - hence the above property does not hold for NLC languages. For BNLC languages, we get the following easy corollaries.

COROLLARY 4.3. Every u-BNLC language contains only a finite number of square grid graphs. □

Clearly, every square grid graph has chromatic number 2 and, moreover, it is planar.

COROLLARY 4.4. The set of planar graphs is not a u-BNLC language. □

COROLLARY 4.5. For each integer k, $k \geq 2$, the set of all unlabeled graphs X with $\chi(X) \leq k$ is not a u-BNLC language. □

5. DISCUSSION

The present paper concludes the series of three papers investigating basic properties of BNLC grammars and languages. The class of BNLC languages is certainly a mathematically natural subclass of the class of NLC languages - it can be defined either by requiring a simple property of all graphs involved in an NLC grammar (i.e., axiom and right-hand sides of productions) or by requiring a simple property of the connection function. We believe that the presented results (here and in Rozenberg & Welzl, 1984a,b) demonstrate that the class of BNLC grammars (and languages) is an interesting class to investigate and that it can play a role in the theory of graph grammars.

Clearly, until now we have considered only the most basic problems concerning BNLC grammars. Many questions about this class still have to be asked (and answered!) in order to get a better understanding of BNLC grammars and languages. We mention here three possible problem areas.

(1) Relationships of the class of BNLC languages to various other classes of graph languages considered in the literature.

(2) Complexity of various standard graph problems but considered within the class of BNLC languages.

(3) Combinatorial properties of BNLC languages, in particular search for more graph invariants which describe properties of BNLC languages (as opposed to arbitrary graph languages or languages from different families).

Acknowledgements: The first author acknowledges the support by NSF Grant MCS 83-05245. Research of the second author has been partially supported by the Austrain "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), Restriction 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.

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

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 Nr. 84-05.

Mycielski, J. (1955), Sur le coloriage des graphs, Colloq. Math. 3, pp. 161-162.

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

Rozenberg, G. & Welzl, E. (1984a), Boundary NLC graph grammars - basic definitions, normal forms and complexity, Institute of Applied Mathematics and Computer Science, University of Leiden, Rep.no. 84-29.

Rozenberg, G. & Welzl, E. (1984b), Graph theoretic closure properties of the family of boundary NLC languages, Institute of Applied Mathematics and Computer Science, University of Leiden, Report no. 84-30.

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