INFORMATION AND COMPUTATION **120**, 1–10 (1995)

# A Normal Form Problem for Unlabeled Boundary NLC Graph Languages

KOICHI YAMAZAKI

*Department of Information Sciences, Tokyo Denki University, Hatoyama-Machi, Hiki-Gun, Saitama 350-03, Japan*

We consider an existential problem of a Chomsky-type normal form for unlabeled boundary NLC graph languages. Let $G$ be a boundary NLC graph grammar, $maxr(G)$ be the maximum for the sizes of graphs which are the start graph (axiom) and the right-hand sides of production rules in $G$, and $und(L(G))$ is the set of underlying unlabeled graphs which are obtained from graphs in the generated language $L(G)$ by removing the labels. Then it is an open question (Rozenberg and Welzl, 1986, *Inform. and Control* **69**, 136–167) whether there exists a positive integer $k_0$ such that there is a boundary NLC graph grammar $G$ with $maxr(G) \leqslant k_0$ and $L = und(L(G))$ for every unlabeled boundary NLC graph language $L$. We will show that there is an infinite hierarchy in the class of the underlying unlabeled boundary NLC graph languages with respect to the sizes of the graphs appearing in the production rules of the boundary NLC graph grammars that generate the underlying unlabeled boundary NLC graph languages. Finally, we will show that there is no integer $k_0$ satisfying the above conditions, using a pumping lemma for boundary NLC graph languages. © 1995 Academic Press, Inc.

## 1. INTRODUCTION

NLC (node-label-controlled) graph grammars were introduced by Janssens and Rozenberg (1980) as a framework for mathematical investigation of graph grammars.

Since then, NLC graph grammars have been intensively investigated by several authors, Ehrenfeucht *et al.* (1984) and Janssens *et al.* (1986), among others. Boundary Node Label Controlled (BNLC) graph grammars were introduced and investigated in (Rozenberg and Welzl, 1986a, b). The BNLC graph grammars are NLC graph grammars such that no two nonterminal nodes of any graph in the production rules and the start graph are adjacent. The BNLC graph languages are an interesting subfamily of the NLC graph languages, because the BNLC graph grammars have attractive properties such as the Church–Rosser property, a neighborhood-preserving normal form, and for connected graphs of bounded degree the membership problem for BNLC graph languages is solvable in polynomial time (see Rozenberg and Welzl, 1986a). In (Rozenberg and Welzl, 1986a), the Chomsky-type normal form problem for BNLC graph grammars was investigated and it was shown that there exists no Chomsky-type normal form for BNLC graph grammars. An unlabeled BNLC (u-BNLC) graph language $L_u$ is the set of underlying unlabeled graphs which are obtained from the graphs in a BNLC graph language $L$ by taking off the labels.

In this paper, we will consider the Chomsky-type normal form problem for the u-BNLC graph languages. In (Rozenberg and Welzl, 1986a), it remains an open question whether there is the Chomsky-type normal form for unlabeled BNLC graph languages, i.e., "whether there is a positive integer $k_0$, such that for every unlabeled BNLC graph language $L$ there is a BNLC graph grammar $G$ with $maxr(G) \leqslant k_0$ and $L = und(L(G))$," where $und(L(G))$ is the set of underlying unlabeled graphs of $L(G)$. It will be proved here that there exists no positive integer $k_0$ for this normal form problem.

As a result, we provide an infinite hierarchy in the class of the underlying unlabeled BNLC graph languages, that is, $\mathscr{L}_k \subsetneq \mathscr{L}_{k+1}$ ($k \geqslant 1$), where $\mathscr{L}_k$ is the class of unlabeled BNLC graph languages that are generated by BNLC graph grammars $G$ with $maxr(G) \leqslant k$. Accordingly, we show that there exists no Chomsky-type normal form for unlabeled BNLC graph languages.

This paper is organized as follows. In Section 2, we introduce definitions, basic notions, and a pumping lemma for BNLC graph grammars. In Section 3, we provide a proof for a hierarchy theorem, that is, $\mathscr{L}_k \subsetneq \mathscr{L}_{k+1}$ ($k \geqslant 1$), using the pumping lemma reviewed in Section 2, and we provide the negative solution to the Chomsky-type normal form for unlabeled BNLC graph languages as a corollary to the hierarchy theorem.

## 2. PRELIMINARIES

We start with basic notations concerning graphs, graph grammars, concrete derivations, and a pumping lemma for BNLC graph languages. We assume familiarity with elementary graph theory (Harary, 1969) and formal language theory (Hopcroft and Ullman, 1979).

### 2.1. *Graphs*

We consider *finite undirected node labeled graphs* without *loops* and *multiple edges*.



0890-5401/95 $12.00
Copyright © 1995 by Academic Press, Inc.
All rights of reproduction in any form reserved.

DEFINITION 2.1. For a set of *labels* $\Sigma$, a *graph* $X$ (*over* $\Sigma$) is specified by $V_X$, $E_X$, and $\varphi_X$ where

- $V_X$ is a finite nonempty set of *nodes*,
- $E_X \subseteq V_X \times V_X$ is a set of *edges*, and
- $\varphi_X$ is a function from $V_X$ into $\Sigma$, called the *labeling function*.

Let $\Sigma$ and $\Phi$ be sets of labels. The set of all graphs over $\Sigma$ is denoted by $G_\Sigma$. The graph $X-x$ is the *subgraph of* $X$ *induced by* $V_X - \{x\}$. The *neighbor of* $x$ *in* $X$ is the set $\{y \in V_X \mid \{x, y\} \in E_X\}$. A graph $X'$ is *isomorphic* to $X$, if there is a bijection from $V_{X'}$ to $V_X$ which preserves labeling and adjacency. The *size of* $X$, denoted by $\sharp X$, is the number of the nodes in $X$.

## 2.2. *BNLC Graph Grammars*

DEFINITION 2.2. A *boundary node label controlled* (*BNLC*) *graph grammar* is a system $G = (\Sigma, \Delta, P, conn, Z_{ax})$, where

- $\Sigma$ is a finite nonempty set of *labels*,
- $\Delta$ is a nonempty subset of $\Sigma$ (the set of *terminals*),
- $P$ is a finite set of pairs $(d, Y)$ where $d$ is in $\Sigma - \Delta$ and $Y$ is a graph over $\Sigma$ such that no two nodes in $Y$ labeled by elements of $\Sigma - \Delta$ are adjacent (the set of *productions*),
- *conn* is a function from $\Sigma$ into $2^\Sigma$ (the *connection function*), and
- $Z_{ax}$ is a graph over $\Sigma$ such that no two nodes in $Z$ labeled by elements of $\Sigma - \Delta$ are adjacent (the *axiom*).

The set $\Sigma - \Delta$ is referred to as the set of *nonterminals*. A node $x$ is a *terminal* (respectively *nonterminal*) *node*, if $x$ is labeled by elements of $\Delta$ (respectively $\Sigma - \Delta$). A production $(A, Y) \in P$ is called a *chain-rule* if $V_Y = \{y\}$ and $y$ is a nonterminal node. A production $(A, Y) \in P$ is called an *empty-rule* if $Y$ is empty graph. A BNLC graph grammar $G$ is called *proper* if $G$ has no chain-rule and no empty-rule. Without loss of generality, we can assume that BNLC graph grammars which are considered in this paper are proper (i.e., an arbitrary BNLC graph grammar can be transformed into a proper BNLC graph language with preserving $L(G)$ and $maxr(G)$; see Rozenberg and Welzl, 1986a).

Let $G = (\Sigma, \Delta, P, conn, Z_{ax})$ be a BNLC grammar, $(d, W)$ be a production in $P$, $X$ and $Y$ be graphs over $\Sigma$ such that $V_X \cap V_Y = \varnothing$ and $Y$ be isomorphic to $W$, and $x$ be a node labeled by $d$ in graph $X$. Then graph $Z$ is derived from graph $X$ by the production $(d, Y)$ in the following way:

Step 1: Delete the node $x$ (and the edges which are incident with $x$) from the graph $X$. (Note that $x$ is labeled by $d$.)

Step 2: Replace $x$ with $Y$. (Note that $Y$ is a copy of $W$.)

Step 3: Connect a node $y$ in $Y$ to the neighbor $x'$ of $x$ by an edge if and only if $\varphi_X(x') \in conn(\varphi_Y(y))$ holds.

Consequently, the graph $Z$ is obtained, where $V_Z = V_{X-x} \cup V_Y$, $E_Z = E_{X-x} \cup E_Y \cup \{\{x', y\} \mid x'$ is a neighbor of $x$, $y \in V_Y$, $\varphi_X(x') \in conn(\varphi_Y(y))\}$, and

$$\varphi_Z(u) = \begin{cases} \varphi_{X-x}(u) & \text{if } u \in V_{X-x}, \text{ and} \\ \varphi_Y(u) & \text{if } u \in V_Y. \end{cases}$$

Then we say that "*$X$ concretely derives $Z$ (in $G$, replacing $x$ by $Y$)*," denoted by $X \Rightarrow_{G(x,Y)} Z$ or simply by $X \Rightarrow_{(x,Y)} Z$.

The *language generated by* $G$, denoted by $L(G)$, is the set $\{X \in G_\Delta \mid Z_{ax} \overset{*}{\Rightarrow}_G X\}$, where the relation $X \Rightarrow_G Z$ means that there is a graph $Z'$ which is isomorphic to $Z$ such that $X$ concretely derives $Z'$ in $G$, and $\overset{*}{\Rightarrow}_G$ denotes the reflexive and transitive closure of $\Rightarrow_G$. A set $L$ of graphs is *a BNLC language* if there is a BNLC graph grammar $G$ such that $L = L(G)$.

## 2.3. *Concrete Derivations*

In this paper, we need the notion of concrete derivation which are introduced in (Rozenberg and Welzl, 1986a) to handle a "concrete" derivation ("concrete" means that the node $x_i$ which is replaced in the graph $X_i$ and the graph $Y_{i+1}$ which substitute the node $x_i$ are written expressly in the derivation). Let $G = (\Sigma, \Delta, P, conn, Z_{ax})$ be a BNLC graph grammar. If a graph $X$ concretely derives a graph $Z$ in $G$, replacing a node $x$ by a graph $Y$, then 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 $G$

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

where $n \geq 1$, is referred to as *a concrete derivation in $G$* (*from $X_0$ to $X_n$*). For derivations, the set of nodes and the function $\varphi$ are extended, that is, $V_D = V_{X_0} \cup \bigcup_{1 \leq i \leq n} V_{Y_i}$, $\varphi_D(x) = \varphi_{Y_i}(x)$, if $x \in V_{Y_i}$ for some $i$ ($1 \leq i \leq n$). The extension is useful when we refer a node of a derivation.

Let

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

be a concrete derivation. Let $\mathcal{O}_D$, which we call the *origin* of $D$, be a distinguished element not in $V_D$. The *predecessor mapping $pred_D$ of $D$* is a function from $V_D$ into $V_D \cup \{\mathcal{O}_D\}$ such that for $x \in V_D$,

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

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

The *history* $hist_D(x)$ *of a node* $x \in V_D$ in $D$ is the sequence $(y_0, y_1, ..., y_m)$, $m \geqslant 1$, $y_i \in V_D$ for all $i$ $(1 \leqslant i \leqslant m)$, such that $y_0 = \mathcal{C}_D$, $y_m = x$, and $y_i = pred_D(y_{i+1})$ for all $i$ $(0 \leqslant i \leqslant m-1)$. Let $(y_0, y_1, ..., y_m)$ be a sequence such that $hist_D(x) = (y_0, y_1, ..., y_m)$, and let $0 \leqslant i < j \leqslant m$. Then we denote the sequence $(y_i, y_{i+1}, ..., y_j)$ by $hist_D(y_i, y_j)$. Note that we can define $hist_D(x, y)$ only when $x$ is an ancestor of $y$. We denote by $targ_D(x)$ the set $\{y \in V_{X_n} \mid x \in hist_D(y)\}$. We denote by $C_D$ the set $\{x_0, x_1, ..., x_{n-1}\}$ of rewritten nonterminal nodes in the concrete derivation $D$. We call the graph $X_n$ *the result of* the concrete derivation $D$ and denote it by $result(D)$.

### 2.4. *Pumping Lemma for BNLC Graph Languages*

We need the notation on iteration of a derivation in order to prove the Chomsky-type normal form in unlabeled boundary NLC graph languages. We will explain the notation by means of an example to lightly grasp the notation (for strict discussion, see Yamazaki and Yaku, to appear).

We consider a derivation $D$ which has nodes $x_p, x_q$ $(x_p \neq x_q) \in C_D$ such that $x_p \in hist_D(x_q)$ and $\varphi_D(x_p) = \varphi_D(x_q)$. Let $t$ be the derivation tree of $D$, $t_1$ be the tree which is obtained by taking away the subtree of $t$ at $x_p$ from $t$, $t_3$ be the subtree of $t$ at $x_q$, and $t_2$ be the tree which is obtained by taking away $t_3$ from the subtree of $t$ at $x_p$. Then we can construct a derivation $D'$ which has following property by rearrangement of derivation steps in $D$: $x_p = x'_s$, $x_q = x'_t$, $result(D) = result(D')$, and $D'$ can be divided into three subderivations, the first, the middle, and the last subderivation, corresponding to $t_1$, $t_2$, and $t_3$, respectively. We call $D'$ rearranged derivation for $D$ with respect to $x_p$ and $x_q$. We construct a "pumped" derivation by iterating the middle subderivation in the rearranged derivation $D'$ with respect to $x_p = x'_s$ and $x_q = x'_t$. We denote $m$ times pumped derivation by $pump(D', x'_s, x'_t, m)$.

EXAMPLE (An Iteration of Derivation $D$). Let

$$D: X_0 \Rightarrow_{(x_0, Y_1)} X_1 \Rightarrow_{(x_1, Y_2)} \cdots \Rightarrow_{(x_6, Y_7)} X_7$$

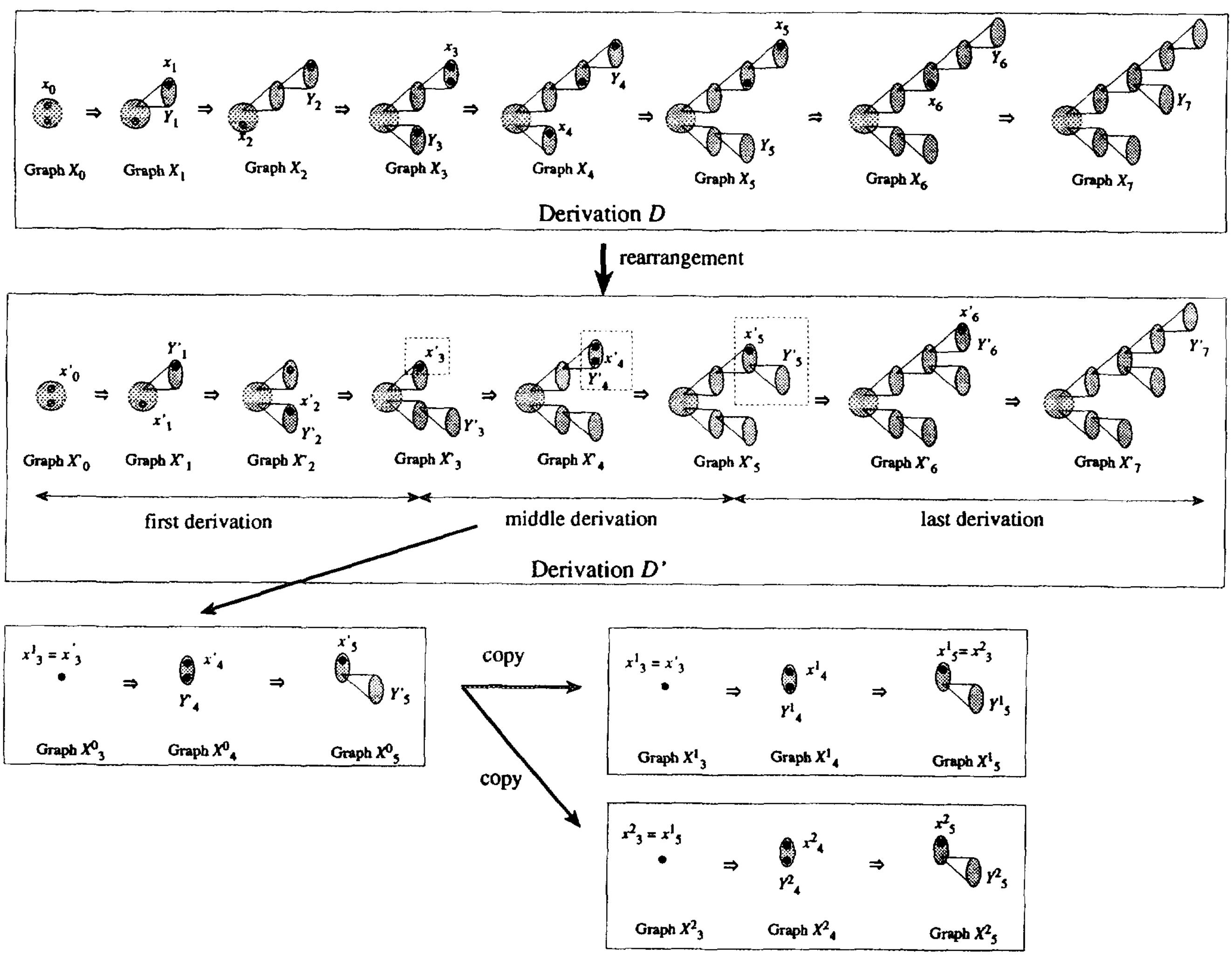



FIG. 2.1. The derivations $D$ and $D'$.

be a derivation such that $x_1 \neq x_3$, $x_1 \in hist_D(x_3)$, and $\varphi_D(x_1) = \varphi_D(x_3)$. Then we can cite the following derivation as a rearranged derivation $D'$ with respect to $x_1$ and $x_3$:

$$D': X'_0 \Rightarrow_{(x'_0,\, Y'_1)} X'_1 \Rightarrow_{(x'_1,\, Y'_2)} \cdots \Rightarrow_{(x'_6,\, Y'_7)} X'_7.$$

Here $x'_0 = x_0$, $x'_1 = x_2$, $x'_2 = x_4$, $x'_3 = x_1$, $x'_4 = x_6$, $x'_5 = x_3$, $x'_6 = x_5$, $Y'_1 = Y_1$, $Y'_2 = Y_3$, $Y'_3 = Y_5$, $Y'_4 = Y_2$, $Y'_5 = Y_7$, $Y'_6 = Y_4$, $Y'_7 = Y_6$ (see Figs. 2.1 and 2.2). The "first subderivation"

$$X'_0 \Rightarrow_{(x'_0,\, Y'_1)} X'_1 \Rightarrow_{(x'_1,\, Y'_2)} X'_2 \Rightarrow_{(x'_2,\, Y'_3)} X'_3$$

constructs $t_1$, the "middle subderivation"

$$X'_3 \Rightarrow_{(x'_3,\, Y'_4)} X'_4 \Rightarrow_{(x'_4,\, Y'_5)} X'_5$$

constructs $t_2$, and the "last subderivation"

$$X'_5 \Rightarrow_{(x'_5,\, Y'_6)} X'_6 \Rightarrow_{(x'_6,\, Y'_7)} X'_7$$

constructs $t_3$ (see Fig. 2.2).

Let

$$X^1_3 \Rightarrow_{(x^1_3,\, Y^1_4)} X^1_4 \Rightarrow_{(x^1_4,\, Y^1_5)} X^1_5,$$
$$X^2_3 \Rightarrow_{(x^2_3,\, Y^2_4)} X^2_4 \Rightarrow_{(x^2_4,\, Y^2_5)} X^2_5$$

be derivations which are isomorphic to the subderivation

$$X^0_3 \Rightarrow_{(x'_3,\, Y'_4)} X^0_4 \Rightarrow_{(x'_4,\, Y'_5)} X^0_5,$$

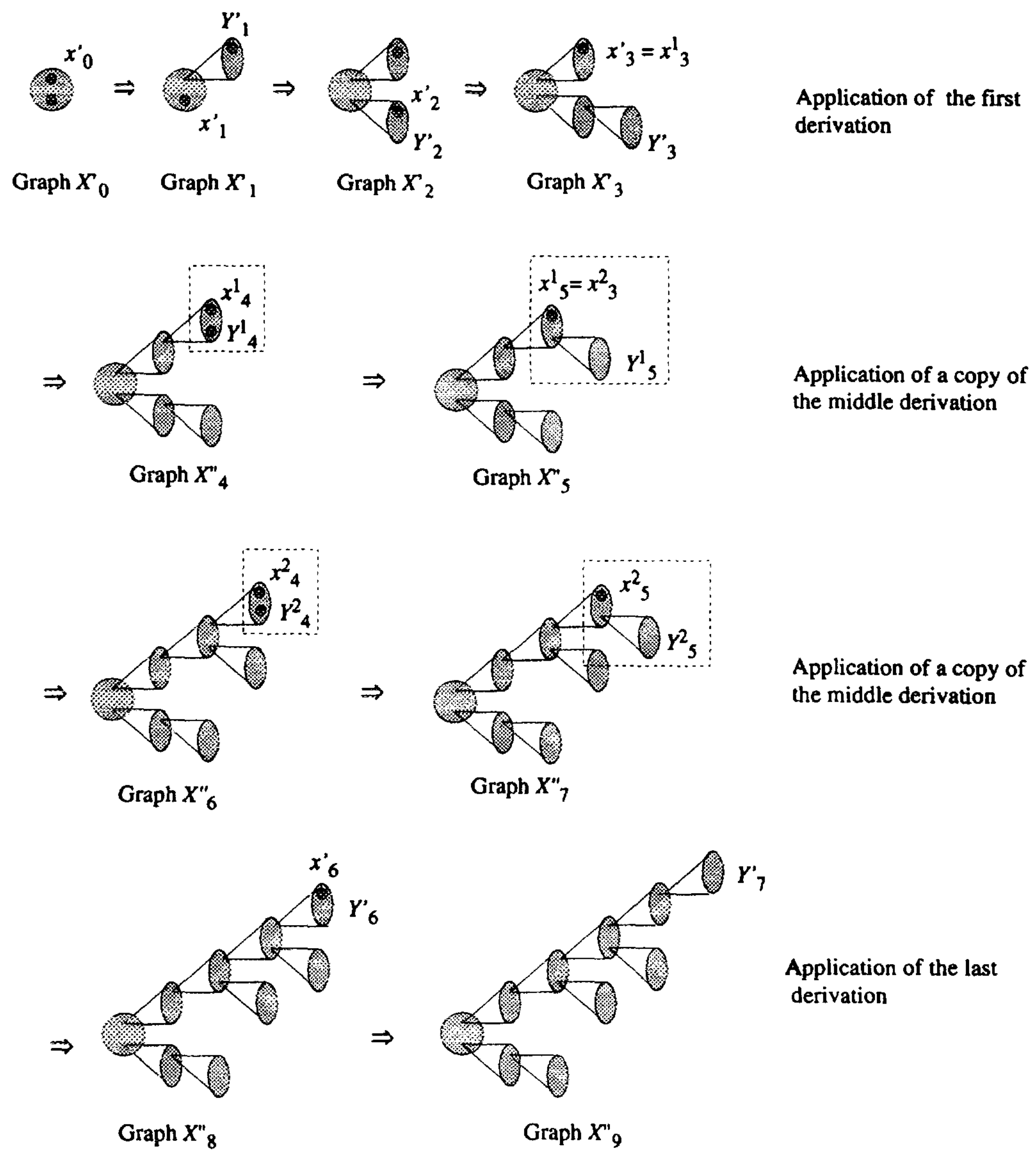
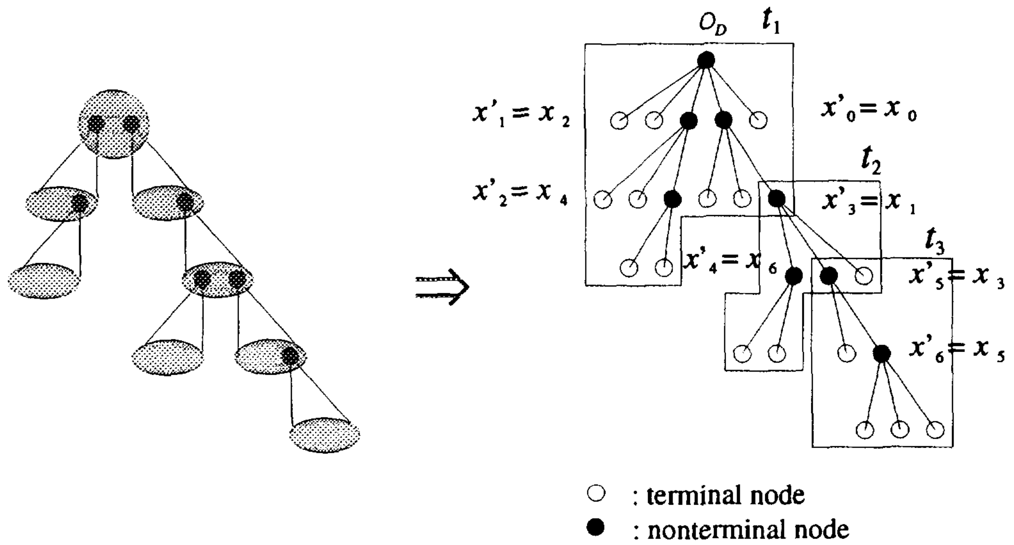



**FIG. 2.2.** The derivation tree of $X'_7$.




Graph $X_7$ Derivation tree of $X'_7$

**FIG. 2.3.** The derivation $D''$.

where $X_3^0$, $X_3^1$, $X_3^2$ are graphs such that $V_{X_3^0} = \{x'_3\}$, $V_{X_3^1} = \{x_3^1\}$, $V_{X_3^2} = \{x_3^2\}$, and $x_3^1 = x'_3$, $x_5^1 = x_3^2$. Then, $D'' = pump(D', x'_3, x'_5, 2)$ is the following derivation (see Figs. 2.3 and 2.4):

$$D'': X'_0 \Rightarrow_{(x'_0,\, Y'_1)} \cdots \Rightarrow_{(x'_2,\, Y'_3)} X'_3$$

(application of the first derivation)

$$\Rightarrow_{(x_3^1,\, Y_4^1)} X''_4 \Rightarrow_{(x_4^1,\, Y_5^1)} X''_5$$

(application of a copy of the middle derivation)

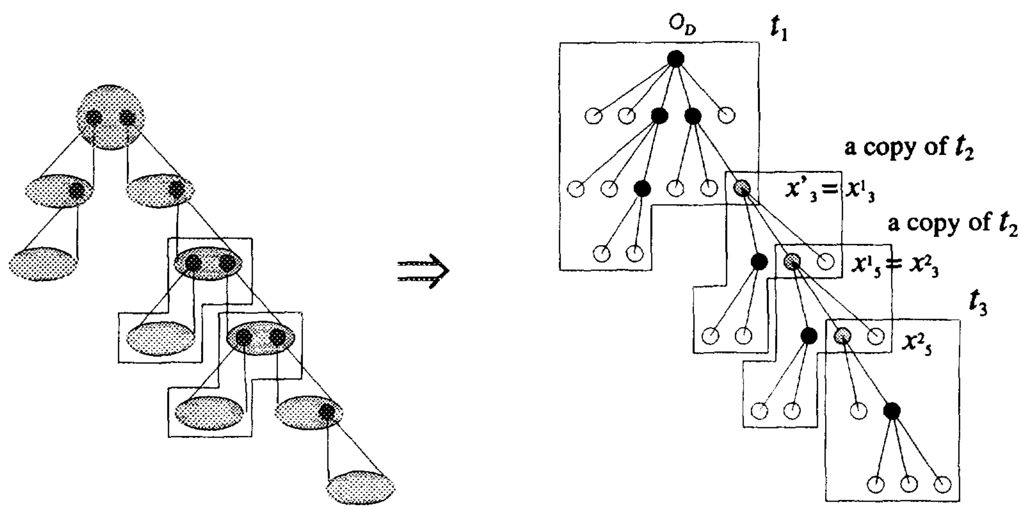
$$\Rightarrow_{(x_3^2,\, Y_4^2)} X''_6 \Rightarrow_{(x_4^2,\, Y_5^2)} X''_7$$

(application of a copy of the middle derivation)

$$\Rightarrow_{(x_5^2,\, Y'_6)} X''_8 \Rightarrow_{(x'_6,\, Y'_7)} X''_9$$

(application of the last derivation).

PROPOSITION 2.1 (Yamazaki and Yaku, 1993). *Let G be a BNLC grammar and D be a derivation which has nodes $x_p, x_q$ $(x_p \neq x_q) \in C_D$ such that $x_p \in hist_D(x_q)$, and $\varphi_D(x_p) = \varphi_D(x_q)$. Let D' be the rearranged derivation for D*




Graph $X''_9$ Derivation tree of $X''_9$

**FIG. 2.4.** The derivation tree of $X''_9$.

*with respect to* $x_p$ *and* $x_q$, *and* $x_p = x'_s$, $x_q = x'_t$. *If* $result(D) \in L(G)$, *then* $result(pump(D', x'_s, x'_t, m)) \in L(G)$ *for each non-negative integer m.*

### 3. A HIERARCHY IN THE u-BNLC GRAPH LANGUAGES

In this section, the following theorem is shown.

THEOREM 3.1. *Let* $\mathscr{L}_k$ *be the class of u-BNLC graph languages derived from BNLC graph grammars with* $maxr(G) \leqslant k$ $(1 \leqslant k)$. *Then* $\mathscr{L}_k \subsetneqq \mathscr{L}_{k+1}$.

To demonstrate this, we will show that there exists a u-BNLC graph language $L_k$ in $\mathscr{L}_{k+1}$ such that for every BNLC graph grammar $G$ with $maxr(G) \leqslant k$, $L_k \neq und(L(G))$. An $L_k$ such as above is constructed by the following method: For each integer $k$ $(1 \leqslant k)$, we consider a BNLC graph grammar $G_k = (\Sigma_k, \Delta_k, P_k, conn_k, Z_{axk})$, where $\Sigma_k = \{a_1, a_2, ..., a_k, s\}$, $\Delta_k = \{a_1, a_2, ..., a_k\}$, $conn_k(a_i) = a_i$ for all $i$ $(1 \leqslant i \leqslant k)$, $conn_k(s) = \Delta_k$, $Z_{axk}$ is a single node with label $s$, $P_k = \{(s, Y_{1k}), (s, Y_{2k})\}$, where $Y_{1k}$ is the complete graph with set of nodes $\{u_1, u_2, ..., u_k, u_{k+1}\}$, where $\varphi_{Y_{1k}}(u_i) = a_i$ for all $i$ $(1 \leqslant i \leqslant k)$, $\varphi_{Y_{1k}}(u_{k+1}) = s$, $Y_{2k}$ is the complete graph with set of nodes $\{v_1, v_2, ..., v_k\}$, where $\varphi_{Y_{2k}}(v_i) = a_i$ for all $i$ $(1 \leqslant i \leqslant k)$. We define an unlabeled graph language $L_k$ by $L_k = und(L(G_k))$.

PROPOSITION 3.2. *Let* $H_u$ *be an underlying unlabeled graph in* $L_k$. *If* $H_u$ *has size* $k \cdot e$ *for some integer e, then,*

(1) *Every maximal clique of* $H_u$ *has order k or e,*

(2) *Every node in* $H_u$ *is included in exactly two maximal cliques, one has order k and the other has order e,*

(3) *Every edge* $\{x, y\}$ *in* $H_u$ *is included in a unique maximal clique.*

*Proof.* Straightforward. ∎

DEFINITION 3.3. Let $k$, $e$ be integers such that $1 \leqslant k < e$ and let $H_u$ be an underlying unlabeled graph with $k \cdot e$ nodes in $L_k$. Then DL (*Different Label*) *group* denotes every maximal clique of order $k$ in $H_u$ and SL (*Same Label*) *group* denotes every maximal clique of order $e$ in $H_u$ (see Fig. 3.1).

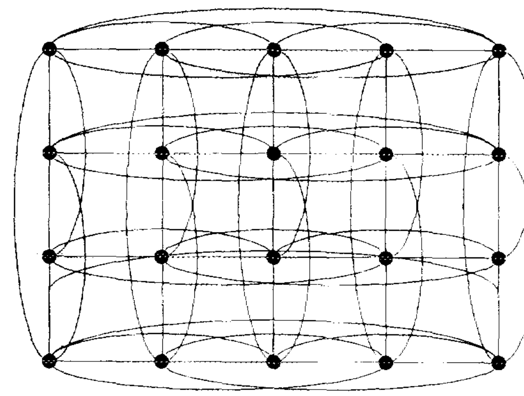



**FIG. 3.1.** An unlabeled graph in $L_4$.

COROLLARY 3.4. *Let* $H_u$ *be an underlying unlabeled graph with more than* $k^2 (1 \leqslant k)$ *nodes in* $L_k$, *and let F be a complete subgraph with more than k nodes in* $H_u$. *Then for all nodes x and y in F, x and y belong to the same SL group.*

*Proof.* Straightforward. ∎

Let $G$ be a BNLC graph grammar such that $und(L(G)) = L_k$. Then we consider a pumped derivation $D''$ which is obtained from a derivation $D$ in $G$. In the rest of this section, such a grammar $G$ and derivation $D$, $D''$ are fixed. Let

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

be a concrete derivation with following properties: (1) $und(result(D)) \in L_k$; (2) $D$ has nodes $x_p, x_q$ $(x_p \neq x_q) \in C_D$ such that $x_p \in hist_D(x_q)$ and $\varphi_D(x_p) = \varphi_D(x_q)$. Let

$$D'\colon X'_0 \Rightarrow_{(x'_0, Y'_1)} X'_1 \Rightarrow_{(x'_1, Y'_2)} \cdots \Rightarrow_{(x'_{n-1}, Y'_n)} X'_n$$

be the rearranged derivation for $D$ with respect to $x_p$ and $x_q$. Let $s$ and $t$ be integers such that $x_p = x'_s$, $x_q = x'_t$.

For $1 \leqslant j \leqslant 2$, let

$$D_j^{copy}\colon X_s^j \Rightarrow_{(x_s^j, Y_{s+1}^j)} \Rightarrow_{(x_{s+1}^j, Y_{s+2}^j)} X_{s+2}^j \cdots \Rightarrow_{(x_{t-2}^j, Y_{t-1}^j)} X_{t-1}^j \Rightarrow_{(x_{t-1}^j, Y_t^j)} X_t^j$$

be the derivations which are isomorphic to the middle derivation,

$$X'_s \Rightarrow_{(x'_s, Y'_{s+1})} X'_{s+1} \cdots \Rightarrow_{(x'_{t-1}, Y'_t)} X'_t,$$

and $h_j$ be an isomorphism from the middle derivation to $D_j^{copy}$. (For convenience, $D_1^{copy}$ is the middle derivation, i.e., $x_i^1 = x'_i$ and $Y_i^1 = Y'_i$ for all $i$ $(s \leqslant i \leqslant t)$.) Then

$$\begin{aligned} D''\colon X'_0 &\Rightarrow_{(x'_0, Y'_1)} X'_1 \cdots \Rightarrow_{(x'_{s-1}, Y'_s)} X'_s \\ &\Rightarrow_{(x_s^1, Y_{s+1}^1)} X''_{s+1} \cdots \Rightarrow_{(x_{t-1}^1, Y_t^1)} X''_{s+(t-s)} \\ &\Rightarrow_{(x_s^2, Y_{s+1}^2)} X''_{s+(t-s)+1} \cdots \Rightarrow_{(x_{t-1}^2, Y_t^2)} X''_{s+2(t-s)} \\ &\Rightarrow_{(x'_t, Y'_{t+1})} X''_{s+1+2(t-s)} \cdots \Rightarrow_{(x'_{n-1}, Y'_n)} X''_{n+(t-s)} \end{aligned}$$

is pumped derivation $D'' = pump(D', x'_s, x'_t, 2)$.

For the derivation $D''$, the following lemmas hold.

LEMMA 3.5. *For each nonterminal node* $y \in hist_{D'}(x'_s, x'_t)$, *if a graph X in the derivation D' has the node y, then y and all terminal nodes of the graph X are adjacent.*

*Proof.* In order to prove this lemma, it suffices to show that $x'_t$ and every terminal node in $X''_{s+(t-s)}$ are adjacent in the graph $X''_{s+(t-s)}$. Suppose, to the contrary, that there exists a node $w$ in $X''_{s+(t-s)}$ such that $x'_t$ and $w$ are not adjacent in the graph $X''_{s+(t-s)}$. Then, there exists no node $u \in targ_{D''}(x'_t)$ such that $u$ and $w$ belong to the same SL group. For the reason that the graph $X'_s$ and $X''_{s+(t-s)}$ have exactly one nonterminal node, $x'_s$ and $x'_t$ (see Yamazaki and Yaku, 1993), this means that the cardinality of the SL group to which $w$ belongs is not influenced by pumping of the derivation $D'$ in a pumped graph (see Fig. 3.2). However, this contradicts the structure of graphs in $L_k$ (i.e., SL groups of a graph in $L_k$ have same size one another). As a result, $x'_t$ and every terminal node in $X''_{s+(t-s)}$ are adjacent in the graph $X''_{s+(t-s)}$. ∎

LEMMA 3.6. *Let $y$ be a terminal node such that $x'_s \in hist_{D''}(y)$ and $x'_t \notin hist_{D''}(y)$ and $z$ be a node such that $h_2(y) = z$. And, let $S$ be the SL group in result($D''$) such that $y \in S$. Then, if $\sharp(V_{X'_s} \cap V_S)$ is greater than $k$, $z$ belongs also to the SL group $S$ in result($D''$).*

*Proof.* If a subgraph $H$ in *result*($D''$) is a complete graph and $\sharp V_H > k$, then $H$ is a subgraph of an SL group, since for every DL group, its size is $k$. Thus, if the subgraph whose node set is $(V_{X'_s} \cap V_S) \cup \{z\}$ is a complete graph, then $z$ belongs also to $S$ in *result*($D''$) by Lemma 3.4 (see Fig. 3.3). Hence, in order to prove this lemma, it suffices to show that the subgraph whose node set is $(V_{X'_s} \cap V_S) \cup \{z\}$ is a complete subgraph in *result*($D''$).

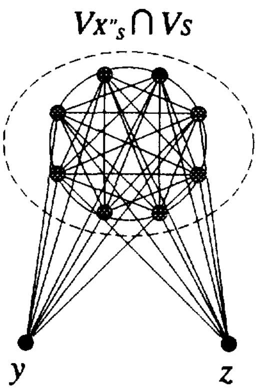


FIG. 3.3. The relation between $y$ and $z$.

Let $(u_0, u_1, ..., u_m)$ and $(v_0, v_1, ..., v_m)$ be the sequences $hist_{D''}(x'_s, y)$ and $hist_{D''}(x'_t, z)$, respectively (see Fig. 3.4). From Lemma 3.5, each node $u$ in $V_{X'_s} \cap V_S$ and $x'_t$ are adjacent in the graph $X''_{s+(t-s)}$. Since $u$ and $y$ are adjacent, $\varphi_{D''}(u) \in conn(\varphi_{D''}(u_i))$ for all $i$ $(1 \leqslant i \leqslant m)$. Thus, $\varphi_{D''}(u) \in conn(\varphi_{D''}(v_i))$ for all $i$ $(1 \leqslant i \leqslant m)$. Hence $z$ and $u$ are adjacent in the graph $X''_{s+1+(t-s)}$. As a result, the subgraph whose node set is $(V_{X'_s} \cap V_S) \cup \{z\}$ is a complete subgraph in *result*($D''$). ∎

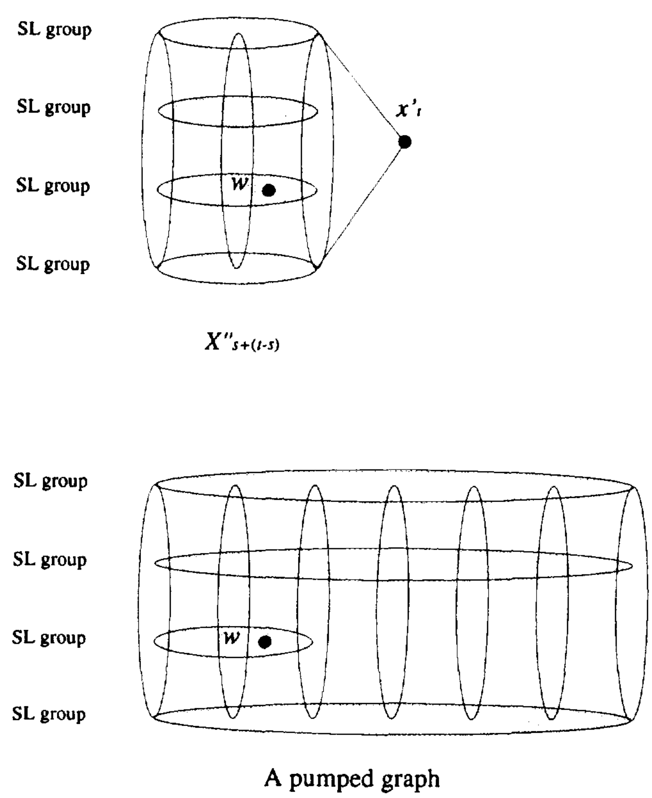


FIG. 3.2. The graph $X''_{s+(t-s)}$ and a pumped graph.

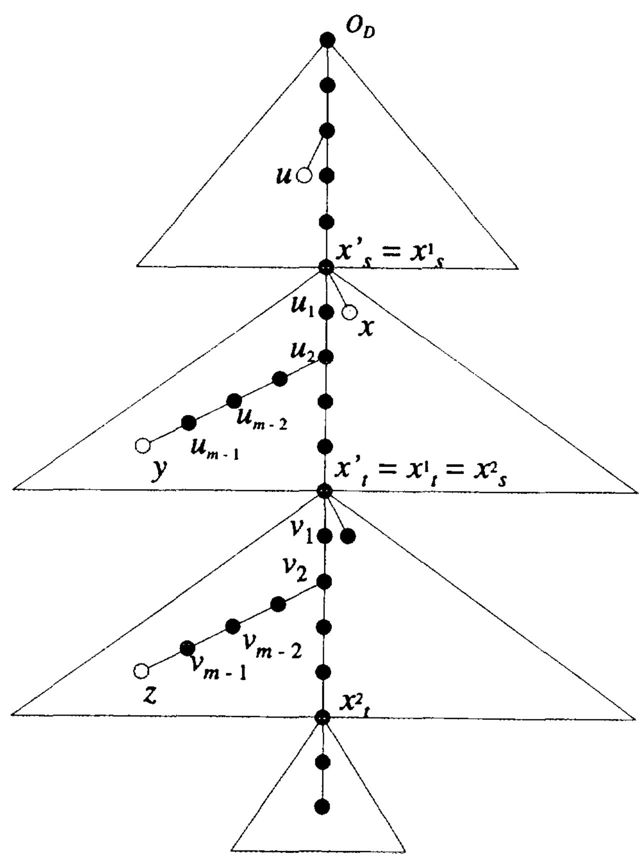


FIG. 3.4. The derivation tree of $D''$.

*Remark* 3.7. We can assume that there exists a derivation $D$ such that for every SL group $S$ in *result*$(D'')$, $\sharp(V_{X'_s} \cap V_S)$ is greater than $k$, because, if initially $D$ is not such a derivation, then we can obtain such a derivation using the change $D$, $x_p$, and $x_q$ as *pump*$(D', x'_s, x'_t, m)$, $h_m(x'_s)$, and $h_m(x'_t)$, respectively, for large enough $m$ (see Fig. 3.5). Hence, without loss of generality, we can assume that $y$ and $z$ belong to the same SL group in *result*$(D'')$ for a terminal node $y$ such that $x'_s \in hist_{D''}(y)$ and $x'_t \notin hist_{D''}(y)$, and $z$ be the terminal node such that $h_2(y) = z$. Moreover, Lemma 3.6 holds for $h_m(y) = z$, $m \geqslant 2$. However, the case of $m = 2$ suffices to solve the Chomsky-type normal form problem for u-BNLC graph languages.

LEMMA 3.8. *For the BNLC graph grammar $G$ which derives $D''$, $k < maxr(G)$.*

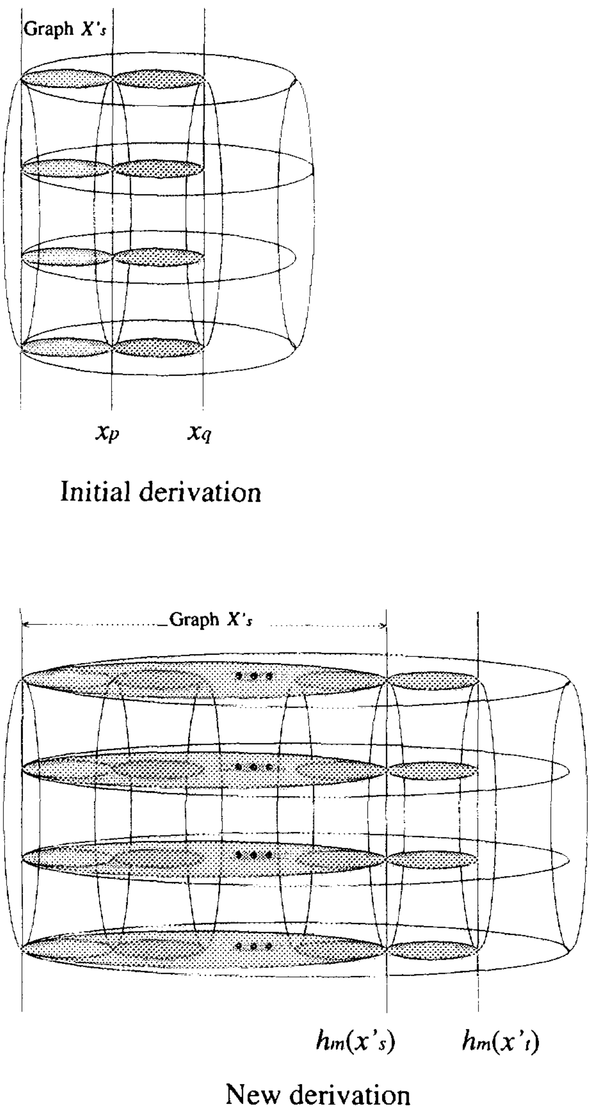



**FIG. 3.5.** A construction a derivation satisfied the assumption.

*Proof.* Suppose $maxr(G) \leqslant k$, then we will show the contrary that $und(result(D'')) \notin L_k$. Note that $und(result(D'')) \in L_k$ by Proposition 2.1. Let $x$ be a terminal node such that $pred_{D''}(x) = x'_s$ (since $G$ is proper, there exists such a terminal node $x$), and let $F$ be a DL group such that $x \in F$ in *result*$(D'')$. Since $maxr(G) \leqslant k$ and the fact that $x'_s$ yield at least one nonterminal node in $D''$, it is impossible to derive $k$ terminal nodes by one derivation step. Hence at least two derivation steps are necessary to derive $k$ terminal nodes in $D''$. Thus there exists some node $y \in F$ such that $pred_{D''}(x) \neq pred_{D''}(y)$. We consider the following three cases into which the time for yielding node $y$ in $D''$ is categorized:

Case 1: $x'_s \notin hist_{D''}(y)$.

Case 2: $x'_s \in hist_{D''}(y)$ and $x'_t \notin hist_{D''}(y)$.

Case 3: $x'_t \in hist_{D''}(y)$.

Case 1 means that $y$ is yielded earlier than $x'_s$ in $D''$, or $y$ and $x'_s$ is yielded at the same time in $D''$ (i.e., $pred_{D''}(y) = pred_{D''}(x'_s)$). Case 2 means that $y$ is yielded after $x'_s$ is yielded and $y$ is yielded earlier than $x'_t$ in $D''$, or $y$ is yielded after $x'_s$ is yielded and $x'_t$ and $y$ are yielded at the same time in $D''$. Case 3 means that $y$ is yielded after $x'_t$ is yielded in $D''$. There exists no case outside the above three cases.

The principle of the proof is to show that there exists $u$, $v$, and $w \in V_{\text{result}(D'')}$ such that:

(p1) $u$, $v$, and $w$ are pairwise adjacent;

(p2) $u$ and $v$ belong to the same SL group in *result*$(D'')$; and

(p3) $u$ and $w$ don't belong to the same SL group in *result*$(D'')$.

The existence of such nodes contradicts Proposition 3.2, thus showing $und(result(D'')) \notin L_k$.

*Case* 1. Let $z \in D_2^{\text{copy}}$ be a node such that $h_2(x) = z$ (see Fig. 3.6). Then, $x'_t = pred_{D''}(z)$ holds.

By hypothesis, $x$ and $y$ are adjacent in *result*$(D'')$. By Lemma 3.6 and Remark 3.7, without loss of generality, we can assume that $x$ and $z$ belong to the same SL group. Thus $x$ and $z$ are adjacent. By Lemma 3.5, $x'_t = pred_{D''}(z)$ and all terminal nodes are adjacent in the graph $X''_{s+(t-s)}$. Hence, as $x'_t = pred_{D''}(z)$, $pred_{D''}(z)$ and $y$ are adjacent. Since $x$ and $y$ are adjacent in *result*$(D'')$, $\varphi_{D''}(y) \in conn(\varphi_{D''}(x))$. Thus $\varphi_{D''}(y) \in conn(\varphi_{D''}(z))$. Therefore $y$ and $z$ are adjacent. Accordingly, $x$, $y$, and $z$ are pairwise adjacent.

Since $x$ and $y$ belong to the same DL group in *result*$(D'')$, they cannot belong to the same SL group in *result*$(D'')$. As we have seen, we show that there exist such nodes $u$, $v$, and $w$ as $x$, $z$, and $y$, respectively.

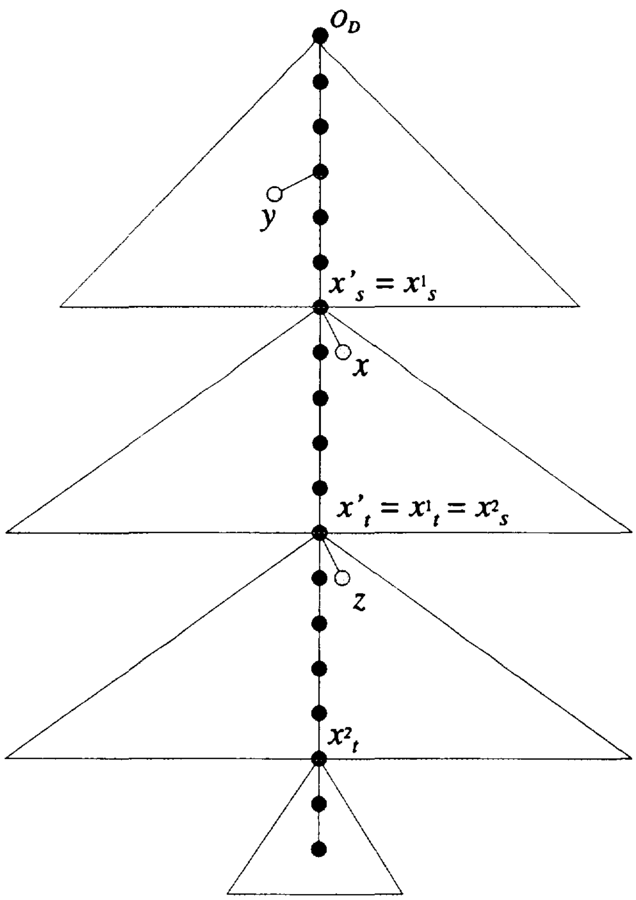



FIG. 3.6. The derivation tree of $D''$ in Case 1.

*Case* 2. Let $z \in D_2^{\text{copy}}$ be the node such that $h_2(y) = z$ and let us denote $hist_{D''}(x'_s, y)$, $hist_{D''}(x'_t, z)$ by $(u_0, u_1, ..., u_m)$, $(v_0, v_1, ..., v_m)$ respectively (Note that $h_2(u_i) = v_i$ for all $i$ $(0 \leqslant i \leqslant m)$) (see Fig. 3.7).

By hypothesis, $x$ and $y$ are adjacent in *result*($D''$). By Lemma 3.6 and Remark 3.7, $y$ and $z$ belong to the same SL group. Thus, $y$ and $z$ are adjacent. By Lemma 3.5, $x'_t$ and $x$ are adjacent. Since $x$ and $y$ are adjacent in *result*($D''$), $\varphi_{D''}(x) \in conn(\varphi_{D''}(u_i))$ for all $i$ $(2 \leqslant i \leqslant m)$. (Note that it is possible to hold $\varphi_{D''}(x) \notin conn(\varphi_{D''}(u_1))$ and $x$ and $u_1$ are adjacent in the graph $Y^1_{s+1}$.) $\varphi_{D''}(x) \in conn(\varphi_{D''}(u_1))$ is guaranteed from adjacency between $x$ and $x^2_t$ in $X''_{s+2(t-s)}$. (It is not difficult to see that $x$ and $x^2_t$ are adjacent in $X''_{s+2(t-s)}$.) Thus $\varphi_{D''}(x) \in conn(\varphi_{D''}(vi))$ for all $i$ $(1 \leqslant i \leqslant m)$. Hence $x$ and $z$ are adjacent in *result*($D''$). Accordingly, $x$, $y$, and $z$ are pairwise adjacent.

In the same way as in Case 1, it is shown that $x$ and $y$ (or $z$) do not belong to the same SL group in *result*($D''$). As we have seen, we show that there are exist such nodes $u$, $v$, and $w$ as $y$, $z$, and $x$, respectively.

*Case* 3. Let $z \in D_2^{\text{copy}}$ be a node such that $h_2(x) = z$ (see Fig. 3.8). By Lemma 3.6 and Remark 3.7, $x$ and $z$ belong to the same SL group. Thus, $x$ and $z$ are adjacent. The fact that $x$ and $y$ adjacent in *result*($D''$) guarantees that $x$ and $z$ are adjacent to $y$ in *result*($D''$). Accordingly, $x$, $y$, and $z$ are pairwise adjacent. In the same way as in Case 1, it is shown that $y$ and $x$ (or $z$) do not belong to the same SL group in *result*($D''$). As we have seen, we show that there are exist such nodes $u$, $v$, and $w$ as $x$, $z$, and $y$, respectively.

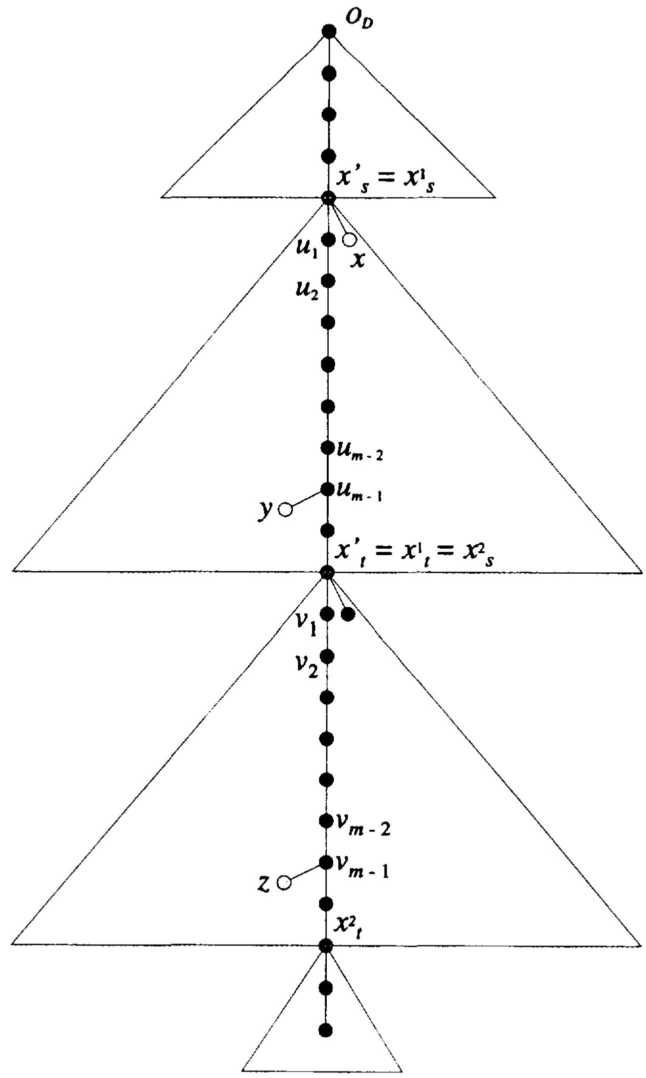



FIG. 3.7. The derivation tree of $D''$ in Case 2.

As a result, *und*(*result*($D''$)) $\notin L_k$ in all cases. ▮

The proof of Theorem 3.1 is clear from the above lemmas. Hence, we have shown that there exists no Chomsky-type normal form for the underlying unlabeled BNLC graph languages.

COROLLARY 3.10. *There is no fixed positive integer k such that for an arbitrary underlying unlabeled BNLC graph language L, there is a BNLC graph grammar G with* $maxr(G) \leqslant k$ *and* $L = und(L(G))$.

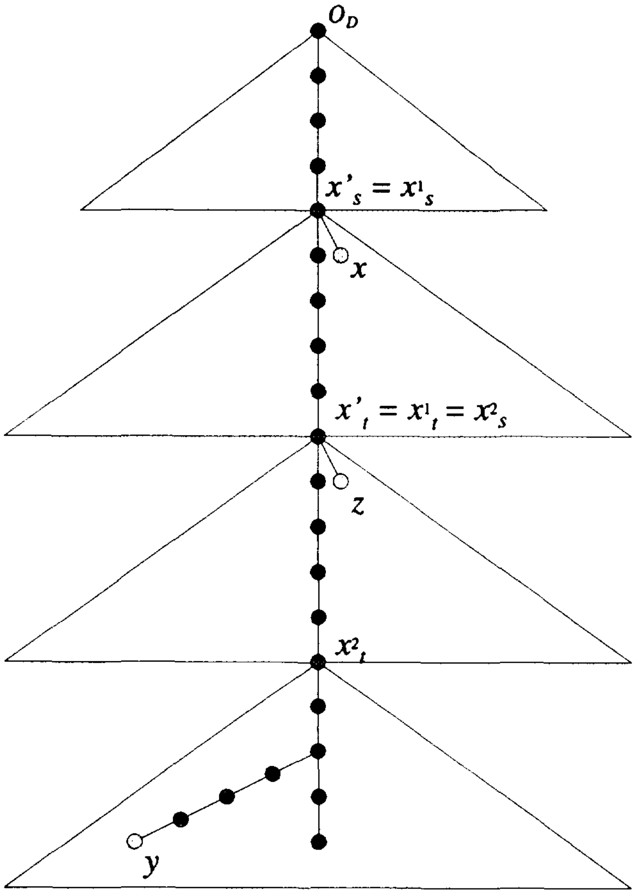



**FIG. 3.8.** The derivation tree of $D''$ in Case 3.

## ACKNOWLEDGMENTS

The author thanks Professor Takeo Yaku of Nihon University for his supervision and valuable suggestions during this research. He is also indebted to Mr. Tetsuro Nishino of the Japan Advanced Institute of Science and Technology and to Professor Shin Hitotsumatsu and Professor Lewis A. Davis of Tokyo Denki University for their careful reading of the manuscript. The author thanks the anonymous referees for many helpful comments and suggestions regarding the presentation of the results and proofs.

Received November 1, 1991; final manuscript received March 8, 1994

## REFERENCES

Ehrenfeucht, A., Main, M., and Rozenberg, G. (1984), Restrictions on NLC grammars, *Theoret. Comput. Sci.* **31**, 211–223.

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

Hopcroft, J. E., and Ullman, J. D. (1979), "Introduction to Automata Theory, Languages and Computation," Addison-Wesley, Reading, MA.

Janssens, D., and Rozenberg, G. (1980), On the structure of node label controlled graph languages, *Inform. Sci.* **20**, 191–216.

Janssens, D., Rozenberg, G., and Welzl, E. (1986), The bounded degree problem for NLC grammars is decidable, *J. Comput. System. Sci.* **33**, 415–422.

Rozenberg, G., and Welzl, E. (1986a), Boundary NLC graph grammars—basic definitions, normal forms, and complexity, *Inform. and Control* **69**, 136–167.

Rozenberg, G., and Welzl, E. (1986b), Graph theoretic closure properties of boundary NLC graph languages, *Acta Inform.* **23**, 289–309.

Yamazaki, K., and Yaku, T. (1993), A pumping lemma and the structure of derivations in the boundary NLC graph languages, *Inform. Sci.* **75**, 81–97.