UNIVERSITY OF WATERLOO
COMPUTER SCIENCE DEPARTMENT

UNIVERSITY OF WATERLOO
COMPUTER SCIENCE DEPARTMENT

Defining Families of Trees with E0L Grammars

Thomas Ottmann
and
Derick Wood

Data Structuring Group
Research Report
CS-89-39

August, 1989

Defining Families of Trees with E0L Grammars *

Thomas Ottmann† Derick Wood‡

Abstract

We consider E0L grammars as tree generating mechanisms. This leads to questions of height, weight, and structural equivalence of E0L grammars. Height equivalence is solved completely, weight equivalence remains open, and structural equivalence is solved for two special cases. We characterize E0L grammars with two nonterminals which generate exactly the sets of 1-2 and 2-3 trees.

1 Introduction

We initiate the study of "context-free" rewriting systems that define well known families of trees such as 1-2 trees, 2-3 trees, brother trees, etc. Our motivation is that rewriting systems provide a precise and familiar means of defining trees, so their study from this point of view is long overdue. A second and fundamental language-theoretic motivation for our investigation is the notion of structural equivalence. This concept is well known for context-free grammars (see [11], for example), but for other "context-free" rewriting systems it has not been considered except for E0L systems in [8]. Two rewriting systems of the same type are structurally equivalent if for every sentential terminating syntax tree in the first system there is a sentential terminating syntax tree in the second system that is identical except for the labeling of internal nodes, and vice versa. The importance of this concept for context-free grammars is that structural equivalence is decidable (see [5]), while language equivalence is undecidable. It is not yet known whether structural equivalence is decidable for other "context-free" rewriting systems.

*This work was supported under a Natural Sciences and Engineering Research Council of Canada Grant No. A-5692 and under a grant from the Information Technology Research Centre.

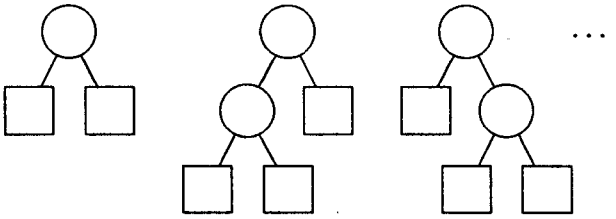
†Institut für Informatik, Universität Freiburg, Rheinstraße 10-12, D-7800 Freiburg, West Germany.

‡Data Structuring Group, Department of Computer Science, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada.

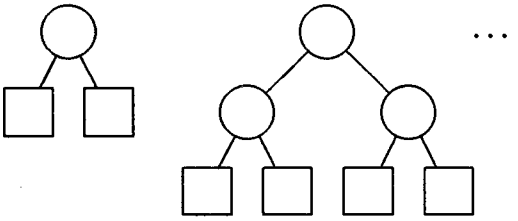
For structural equivalence only labels of internal nodes are ignored; here we ignore all labels; that is, we have only one terminal symbol. For example, the context-free grammar

$$S \rightarrow aa \mid SS$$

generates the set of extended binary trees



However, when we view this as an EOL grammar, it generates all perfect binary trees



In this paper, we consider EOL grammars, rather than context-free grammars, since they are powerful enough to describe 2-3 trees, brother trees, and stratified trees. Specifically, we characterize in Section 5 those EOL grammars with at most two nonterminals that generate all 2-3 trees. In Section 3 we look briefly at the heights of trees generated by EOL grammars. Given two EOL grammars it is decidable if they generate trees of the same height. Finally, in Section 4 we consider the weights of trees generated by EOL grammars. Apart from a reduction result no further results have been obtained in this case. Whether or not an EOL grammar generates trees with each possible weight—weight universality—appears to be a difficult question. The only positive result in this area is the decidability of almost-weight universality (all but finitely many weights) for unary OL grammars found in [2].

To ensure that the paper is selfcontained we provide the necessary definitions in the following section.

2 Definitions

We begin our exploration by defining sets of trees—the (a, b) trees. This is followed by the definition of (unary) EOL grammars and their associated

trees.

Let a and b be two integers that satisfy $1 \leq a \leq b$. An (a,b) *tree* t of n internal nodes either is *empty* and consists of an external node, if $n = 0$, or consists of a *root node* u together with r subtrees $t_1, \ldots, t_r$ of u having $n_1, \ldots, n_r$ internal nodes, respectively. We also require, in this latter case, that $a \leq r \leq b$ and $1 + n_1 + \cdots + n_r = n$.

Given an (a,b) tree t its *height* is recursively defined by

$$height(t) = \begin{cases} 0 & \text{if } t \text{ is empty} \\ 1 + max(\{height(t_i) : 1 \leq i \leq r\}) & \text{if } t \text{ is nonempty and its root} \\ \text{has } r \text{ subtrees } t_1, \ldots, t_r. & \end{cases}$$

and its *weight* is defined recursively by

$$weight(t) = \begin{cases} 1 & \text{if } t \text{ is empty} \\ \sum_{i=1}^{r} weight(t_i) & \text{if } t \text{ is nonempty and its root} \\ \text{has } r \text{ subtrees } t_1, \ldots, t_r. & \end{cases}$$

We say that an (a,b) tree has *uniform depth* if its external nodes are all at the same distance from the root. More formally, an empty (a,b) tree has uniform depth, and a nonempty (a,b) tree has uniform depth if its root has exactly r subtrees $t_1, \ldots, t_r$, these have uniform depth, and $height(t_1) = \cdots = height(t_r)$.

Remark: From hereon in we are only concerned with uniform depth trees so we call them, simply, trees.

A $(2,2)$ tree is called a *binary tree*, a $(1,2)$ tree is called a *unary-binary tree*, and a $(2,3)$ tree is called a *binary-ternary tree*. Note that $(2,3)$ trees are the well known 2-3 trees[1], while brother trees, neighbor trees, and son trees are all $(1,2)$ trees[7, 3, 4, 6].

For our purposes an E0L grammar is defined as follows. Let a be the *universal terminal symbol* throughout this paper. An *E0L grammar* G is a triple (N, P, S), where

N is an alphabet of *nonterminals*,

$P \subseteq N \times (N^+ \cup a^+)$ is a finite set of *productions*, and

$S \subseteq N$ is a nonempty set of *sentence symbols*.

This definition differs from the traditional one, see [9] for example, in four respects. First, we only have one terminal symbol; they are really *unary* E0L grammars. Second, only nonterminals have productions; the grammar is *synchronized*, see [9]. Third, the productions cannot have an empty right hand side; the grammar is *propagating*. Fourth, there can be more than one sentence symbol. Each of these modifications, apart from the first, does not

affect the languages generated by E0L grammars. (Apart from the loss of the empty word.)

Rewriting is defined in the usual way. Let α be a nonempty word over N; that is, $\alpha = A_1 \cdots A_n$, where A_i is in N, $1 \leq i \leq n$, and $n = |\ \alpha\ |$. Then, α can be rewritten as $\beta = \beta_1 \cdots \beta_n$, for some β_i in $(N \cup \{a\})^+$, $1 \leq i \leq n$, if $A_i \rightarrow \beta_i$ is in P, $1 \leq i \leq n$. We usually denote this by

$$\alpha \Rightarrow \beta$$

We write $\alpha \Rightarrow^d \beta$ to denote that α gives β in d steps, for $d \geq 1$, if either $d = 1$ and $\alpha \Rightarrow \beta$, or $d > 1$ and there exists γ in N^+ such that $\alpha \Rightarrow \gamma$ and $\gamma \Rightarrow^{d-1} \beta$. We write $\alpha \Rightarrow^+ \beta$ if $\alpha \Rightarrow^d \beta$, for some $d \geq 1$, and we write $\alpha \Rightarrow^* \beta$ if either $\alpha = \beta$ or $\alpha \Rightarrow^+ \beta$. We say that $\alpha \Rightarrow^+ \beta$ and $\alpha \Rightarrow^* \beta$ are *derivations.* A derivation $\sigma \Rightarrow^* \beta$, for some σ in S, is called a *sentential derivation.* Note that only purely nonterminal words can be rewritten. This is the reason for only allowing right hand sides of productions to be either completely nonterminal or completely terminal. We say that a word α is *d-generable* if there is a σ in S such that $\sigma \Rightarrow^d \alpha$ or $d = 0$ and $\sigma = \alpha$.

The *language generated by* G is denoted by $L(G)$ and is defined by

$$L(G) = \{x : x \text{ is in } a^+ \text{and } \sigma \Rightarrow^+ x, \text{for some } \sigma \text{ in } S\}$$

We say that an E0L grammar $G = (N, P, S)$ is *reduced* if each sentence symbol generates a terminal word and each nonterminal appears in at least one sentential derivation of a terminal word. A reduced grammar does not contain any useless nonterminals.

With each derivation of a terminal word in G we can associate a *syntax tree.* This is a uniform depth tree that has internal nodes labeled with nonterminal symbols and external nodes labeled with a. It also satisfies the following condition:

> For all internal nodes u, if u has r children $u_1, \ldots, u_r$ for some $r \geq 1$, then $L(u) \rightarrow L(u_1) \cdots L(u_r)$ is in P, where $L(v)$ denotes the label of node v.

We are particularly interested in syntax trees that have a root labeled with a sentence symbol; we call these *sentential syntax trees.* If we remove the labels from a sentential syntax tree we obtain a *stripped sentential syntax tree.* We denote by $T(G)$ the *set of stripped sentential syntax trees of* G. Note that a derivation $\sigma \Rightarrow^d x$, for σ in S and x in a^+, yields a sentential syntax tree of height d.

We close this section with two examples.

Example 2.1: Let G be given by

$$\begin{aligned} B &\rightarrow aa \mid BU \mid UB \mid BB \\ U &\rightarrow a \mid B, \end{aligned}$$

where B is the only sentence symbol. Then, $T(G)$ is the set of all nonempty brother trees[7]. □

Example 2.2: Let G be given by

$$S \rightarrow aa \mid aaa \mid SS \mid SSS.$$

Then, $T(G)$ is the set of all nonempty 2-3 trees[1]. □

3 Height

Given a set of trees T its *height set* is denoted by $H(T)$ and is defined as $H(T) = \{height(t) : t \text{ is in } T\}$. We say two sets of trees T_1 and T_2 are *height equivalent* if $H(T_1) = H(T_2)$.

It is well known that for an E0L grammar G, the corresponding set of heights, $H(T(G))$, is an ultimately periodic set; see [12] for example. Moreover, $H(T(G))$ can be computed effectively, see [12], for example. These results lead immediately to the following theorem.

Theorem 3.1 *The height equivalence problem for E0L grammars is decidable.*

Let $h \geq 0$ be a given height and T be a set of trees. Then, $H(T, h)$ denotes *the height set of T modulo h* and it is defined as $H(T, h) = \{h' : h' = height(t), \text{ for some } t \in T \text{ and } h' \geq h\}$. Clearly, $H(T) = H(T, 0)$. We say that two sets of trees T_1 and T_2 are *ultimately height equivalent* if there exists $h \geq 0$ such that $H(T_1, h) = H(T_2, h)$. Clearly this holds if and only if $H(T_1) - H(T_2)$ and $H(T_2) - H(T_1)$ are both finite. Because the height sets of E0L grammars are ultimately periodic, the difference of two such sets is also ultimately periodic. This yields our second theorem.

Theorem 3.2 *The ultimate height equivalence problem for E0L grammars is decidable.*

4 Weight

Given a set of trees T its *weight set* is denoted by $W(T)$ and is defined as $W(T) = \{weight(t) : t \text{ is in } T\}$. (This is usually called the length set of the language.) We say two sets of trees T_1 and T_2 are *weight equivalent* if $W(T_1) = W(T_2)$. We say that a set of trees T is *weight universal* if $W(T)$ equals the natural numbers and *almost weight universal* if $W(T)$ is cofinite with respect to the natural numbers.

Our main result is that we only need consider these questions for the so called UB grammars. An E0L grammar $G = (N, P, S)$ is a *unary-binary grammar* or UB *grammar* if, for all productions $A \rightarrow \alpha$ in P, we have $| \alpha | \leq 2$.

Theorem 4.1 *Let $G = (N, P, S)$ be an EOL grammar. Then, a weight equivalent UB grammar G' can be effectively constructed from G.*

Proof: Let $m = maxr(G) = max(\{| \alpha |: A \to \alpha \text{ is in } P\})$. If $m \leq 2$, then G is the required UB grammar already. Therefore assume $m \geq 3$. We stretch each production $A \to \alpha$ in P into a derivation sequence of length $m-1$ as follows, where $\alpha = \alpha_1 \cdots \alpha_n$, for α_i in $N \cup \{a\}$, $1 \leq i \leq n$.

(1) $| \alpha | = 1$. Add productions $A \to A_1$; $A_1 \to A_2$; $\cdots$; $A_{m-2} \to \alpha$, where the A_i are new nonterminals with respect to $A \to \alpha$.

(2) $| \alpha | \geq 2$. Add productions $A \to A_1$; $A_1 \to A_2$; $\cdots$; $A_{k-1} \to A_k$; $A_k \to A_{k+1}B_{k+1,k+1}$; $A_{k+1} \to A_{k+2}B_{k+2,k+2}$; $B_{k+1,k+1} \to B_{k+1,k+2}$; $\cdots$; $A_{m-2} \to \alpha_1\alpha_2$; $B_{m-2,m-2} \to \alpha_3$; $\cdots$; $B_{k+1,m-2} \to \alpha_n$, where the A_i and $B_{i,j}$ are new nonterminals with respect to $A \to \alpha$ and $k = m - n + 1$.

Clearly a single derivation step in G is simulated by $m-1$ derivation steps in G' and vice versa. Hence, not only are G and G' weight equivalent, they are also equivalent. □

Unfortunately questions concerning the weight of a UB grammar, even weight universality, appear to be very hard. Ruohonen [10] has shown that weight equivalence is undecidable for DT0L grammars.

5 Structure

In Sections 3 and 4 two coarse measures of structure have been examined, namely height and weight. In the present section we wish to investigate, in finer detail, the set of trees generated by an E0L grammar. The specific question we consider is the following. Given a set of trees T and an E0L grammar G is $T(G) = T$? In other words is the grammar *T-universal*?

To make our investigation more concrete we consider three example sets. These are B, the set of all binary trees, UB the set of all unary-binary trees, and BT the set of all binary-ternary trees.

First, we obtain a reduction theorem along the lines of Theorem 4.1.

Theorem 5.1 *Let G_1 and G_2 be two EOL grammars such that $maxr(G_1) = maxr(G_2) = m$, for some $m \geq 1$. Then, two UB grammars G_1' and G_2' can be effectively constructed from G_1 and G_2 such that*

(i) G_i and G_i' are equivalent, for $i = 1, 2$, and

(ii) G_1 and G_2 are structurally equivalent if and only if G_1' and G_2' are.

Proof: Carry out the construction of Theorem 4.1 on both G_1 and G_2, noting that the maximum length of right hand sides of productions in G_1 must equal that of G_2 if G_1 and G_2 are structurally equivalent. The construction replaces height one subtrees everywhere by height $m-1$ subtrees. As each replacement is uniquely determined by the length of the corresponding right hand side, condition (ii) holds. □

Given an E0L grammar $G = (N, P, S)$, can we decide if it is B-universal, UB-universal or BT-universal? We consider these three decision problems one at a time.

Theorem 5.2 *B-universality of E0L-grammars is decidable.*

Proof: Consider an arbitrary E0L grammar $G = (N, P, S)$. Assume G is reduced (for if it is not, then it can be reduced effectively.) Now if $T(G) \subseteq B$, every production in P must have the form

$$A \rightarrow \alpha$$

where $\mid \alpha \mid = 2$. This is the first necessary condition for $T(G)$ and B to be equal.

Second, if $T(G) \supseteq B$, then $H(T(G))$ is the set of natural numbers; that is, G is height-universal. This is the second necessary condition for $T(G)$ and B to be equal.

We claim that these two conditions are also sufficient. For the first condition implies that G only generates binary trees, while the second condition implies that a binary tree of each height is generated. Since there is only one binary tree of each height, this implies that $T(G) = B$.

To complete the theorem, observe that both conditions are decidable; the second by way of Theorem 3.1. □

We have characterized B-universality; however, UB- and BT-universality are more difficult. We need to consider the structural properties of our grammars in more depth. To this end we say that an E0L grammar $G = (N, P, S)$ is *invertible* if no two productions in P have the same right hand side. This implies that each tree in $T(G)$, for such a grammar G, corresponds to exactly one syntax tree.

Theorem 5.3 *Let $G = (N, P, S)$ be an E0L grammar. Then, an invertible structurally equivalent E0L grammar $G' = (N', P', S')$ can be effectively constructed from G.*

Proof: Define N' to be the set $\{X \subseteq N : X \neq \emptyset\}$ and S' the set $\{X : X \subseteq N \text{ and } X \cap S \neq \emptyset\}$. Given a word α' over N' we say a word α over N

corresponds to α' if $|\ \alpha\ | = |\ \alpha'\ |$ and each nonterminal symbol in α belongs to the set of nonterminal symbols appearing at the same position in α'.

The set P' of productions is defined as follows.

(i) P' contains a production $X \to \alpha'$, for $\alpha' \in N'^{+}$, if and only if

$$X = \{A : A \in N, A \to \alpha \in P \text{ and } \alpha \text{ corresponds to } \alpha'\}.$$

(ii) P' contains a production $X \to a^i$, for $i \geq 1$, if and only if

$$X = \{A : A \to a^i \in P\}.$$

The right hand side of each production in P' uniquely determines its left hand side. Thus, it is clear that G' is invertible.

Next, we have to show that G and G' are structurally equivalent. We prove that for each syntax tree of G which generates a terminal word there is a syntax tree of G' of exactly the same structure generating the same word and vice versa.

First, consider a syntax tree for $x \in a^+$, $x \in L(G)$. We construct a syntax tree in G' for x bottom up as follows. Each subword a^i of x generated by a production $A \to a^i$ in G is generated by the production $X \to a^i$, where $X = \{A : A \to a^i \in P\}$. Thus the nonterminals occurring in the syntax tree in G for x on the first level above the terminal level correspond to the nonterminals of G' on this level in the obvious way. Now assume that we know already that all nonterminals on level $l+1$ in the syntax tree in G for x correspond to the nonterminals of G' appearing at that level. Let $A_1 \cdots A_k$ on level $l+1$ be generated by a production $A \to A_1 \cdots A_k$ in P. By the assumption, in the syntax tree in G' we have $X_1, \ldots, X_k$ occurring at the same positions where the variables $A_1, \ldots, A_k$ occur, and $A_i \in X_i$, for $1 \leq i \leq k$. Now we have a (unique) production $X \to X_1 \cdots X_k$ in P' such that $A \in X$. In this way we obtain a uniquely determined sequence of nonterminals of G' such that each nonterminal in G on level l corresponds to the nonterminal of G' at that level at the same position. Finally, we obtain a set $X \subseteq N$ containing the sentence symbol which occurs at the root of the syntax tree for x in G. By definition, $X \in S'$. Therefore, we have obtained a syntax tree for x in G' of the same structure.

Conversely, consider a syntax tree for x in G'. We construct a syntax tree of the same structure for x in G top-down as follows. If at the root of the syntax tree in G' a production $X \to X_1 \cdots X_k$ was applied, we know $X \cap S \neq \emptyset$. Choose $\sigma \in X \cap S$ and a production $\sigma \to A_1 \cdots A_k \in P$ such that $A_i \in X_i$, for $1 \leq i \leq k$. Because $A_1 \cdots A_k$ corresponds to $X_1 \ldots X_k$ we must have such a production. By similar arguments we may conclude that each derivation step in G' can be mimicked by a derivation step in G

leading to a syntax tree for x of exactly the same structure. □

In what follows we will make frequent use of the following fact which is an immediate consequence of invertibility.

> If $G = (N, P, S)$ is an invertible E0L grammar and $X \Rightarrow^* \alpha$, for $X \in N$ and $\alpha \in N^+ \cup a^+$, then there is no other nonterminal $Y \neq X$, which also generates α and appears as the label of the root of a syntax tree of the same structure.

The construction of an invertible E0L grammar is also possible if the given E0L grammar has more than just one terminal symbol. Thus, Theorem 5.3 holds in this general case also.

For an E0L grammar to be UB-universal, it must be a UB grammar. Since we may assume that it is also invertible, we have the following preliminary result.

Lemma 5.4 *Let $G = (N, P, S)$ be a reduced invertible UB grammar which is UB-universal. Then, $S = N$.*

Proof: Consider an arbitrary nonterminal A in N. Since G is reduced A generates at least one UB-tree t. Because G is invertible no other nonterminal generates t. Finally, because G is UB-universal, t is in $T(G)$ and A is in S. □

The first result on UB-universality characterizes UB-grammars with a single sentence symbol.

Theorem 5.5 *Let $G = (N, P, S)$ be a reduced, invertible UB-grammar with $S = \{A\}$, for some A in N. Then, G is UB-universal if and only if $N = \{A\}$ and $P = \{A \rightarrow a \mid aa \mid A \mid AA\}$.*

Proof: Clearly G is UB-universal if it satisfies the given conditions. Therefore, assume G is UB-universal and $S = \{A\}$. By Lemma 5.4, $N = \{A\}$ and, therefore, P must have the given form. □

Theorem 5.5 and its proof obviously carry over to the case of binary-ternary trees. Therefore, we have the following result.

Corollary 5.6 *Let $G = (N, P, S)$ be a reduced, invertible BT-grammar with $S = \{A\}$, for some A in N. Then, G is BT-universal if and only if $N = \{A\}$ and $P = \{A \rightarrow aa \mid aaa \mid AA \mid AAA\}$.*

But what happens if $\#S > 1$? We completely characterize the two-letter case, the case of $\#S > 2$ is left as an open problem.

We first reduce structural equivalence of E0L grammars to equivalence via parenthesized versions of grammars. Given an E0L grammar $G = (N, P, S)$. The *parenthesized version* $G_{()}$ of G is the E0L grammar $G_{()} = (N, P_{()}, S)$ where $X \to (\alpha)$ is a production in $P_{()}$ if and only if $X \to \alpha$ is a production in P. The left and right parentheses "(" and ")" are considered to be new terminal symbols which, once generated, remain unchanged. This can be achieved by adding productions $(\to ($ and $) \to)$ to $P_{()}$. However, we usually do not mention these productions explicitly. Rewriting and other related notions from Section 2 are extended to parenthesized versions of grammars in the obvious way. Of course, if $L(G) \subseteq \{a\}^+$, then $L(G_{()}) \subseteq \{a, (,)\}^+$.

Example 5.1: The parenthesized version of the grammar of Example 2.1 is the grammar $G_{()}$ given by

$$\begin{aligned} B &\to (aa) \mid (BU) \mid (UB) \mid (BB) \\ U &\to (a) \mid (B) \end{aligned}$$

Observe that for a word $x \in L(G_{()})$ all terminal symbols a in x are surrounded by the same number of matching pairs of parentheses. □

Obviously, two E0L grammars G and G' are structurally equivalent if and only if their corresponding parenthesized versions generate the same language $L \subseteq \{(,), a\}^+$.

A *parenthesized nonterminal context* is a sentential form of the parenthesized version $G_{()}$ of an E0L grammar $G = (N, P, S)$ in which one occurrence of a nonterminal symbol is replaced by an underscore. We call such a word $\alpha \in (N \cup \{(,)\} \cup \{\_\})^+$ simply a *context*. Given a context α and a nonterminal A, $\alpha[A]$ denotes the word obtained by replacing the underscore in α with A.

Let $\alpha[A]$ be d-generable in $G_{()}$; then $\alpha[A]$ can be identified with a sentential syntax tree of height d in G from which all labels except for the labels at its frontier have been removed. We write $frontier(\alpha[A])$ to denote the sequence of labels at the frontier of this tree. Clearly, $frontier(\alpha[A])$ is d-generable in G if and only if $\alpha[A]$ is d-generable in $G_{()}$.

Example 5.1 (continued): $\alpha = (((UB)(UB))((\_\ B)))$ is a context,and $frontier(\alpha[B])$ and $frontier(\alpha[U])$ are both 3-derivable in G. The context α captures the structure of a syntax tree in G from which all labels except for the labels at the bottommost level and the label of the node with the underscore have been removed. Thus, α can be identified with the tree of Figure 1. □

Given an E0L grammar $G = (N, P, S)$ and two nonterminals A and B, we say that A and B are *d-context equivalent*, denoted by $A \equiv^d B$, if, for all contexts α, the word $\alpha[A]$ is d-generable if and only if $\alpha[B]$ is d-generable. A and B are said to be *context equivalent*, denoted by $A \equiv B$, if $A \equiv^d B$,

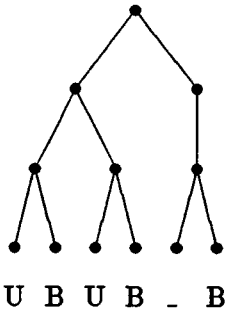


Figure 1: The syntax tree for the context α.

for all integers $d \geq 0$. We say that an EOL grammar is *context reduced*, if every pair of different nonterminals is not context equivalent.

Example 5.2: Let $G = (N, P, S)$ be given by

$$\begin{aligned} A &\rightarrow a \mid aa \mid AA \mid B \mid AB \\ B &\rightarrow A \mid BB \mid BA, \end{aligned}$$

where A and B are both sentence symbols. $A \equiv^0 B$, because $A, B \in S$. We show that $A \equiv^d B$ implies $A \equiv^{d+1} B$.

Let $A \equiv^d B$ and let α be an arbitrary context such that $\alpha[A]$ is $(d+1)$-generable. From the frontier of this tree of height $d+1$ we construct labels for all nodes at level d just above the leaves as follows. Associate label A to a node at level d, if its successors at level $d+1$ are labeled AA or B or AB; associate label B to a node at level d, if its successors are labeled A or BB or BA.

The predecessor on level d of the node representing the underscore and labeled with A on level $d+1$ is associated with either the label A or B. Replace this label with an underscore. Thus, we obtain a context β such that for either $X = A$ or $X = B$ we have: $\beta[X]$ is d-generable, $frontier(\beta[X]) \Rightarrow frontier(\alpha[A])$ in G, and X generates the occurrence of A replacing the underscore in α. Now either $frontier(\alpha[B])$ is also derivable from $frontier(\beta[X])$ in one step by replacing X by a different righthand side of a production of G or the inductive assumption $A \equiv^d B$ is applied in order to conclude that $\beta[Y]$ is d-generable, where $Y \in \{A, B\}$, and $frontier(\beta[Y]) \Rightarrow frontier(\alpha[B])$ in G. Thus, $\alpha[B]$ is $(d+1)$-generable. By symmetry we obtain in the same way that $\alpha[B]$ is $(d+1)$-generable implies that $\alpha[A]$ is $(d+1)$-generable, therefore, $A \equiv^{d+1} B$. This shows that $A \equiv B$.

If we identify the nonterminals A and B we obtain the structurally equivalent EOL grammar

$$A \rightarrow a \mid aa \mid A \mid AA.$$

□

Definition 5.1 *An EOL grammar is* simplified, *if it satisfies the three conditions:*

1. *it is reduced;*
2. *it is invertible; and*
3. *it is context reduced.*

Example 5.2: Let G be

$$\begin{aligned} A &\rightarrow a \mid aa \\ B &\rightarrow B \mid BB \mid A \mid AA, \end{aligned}$$

where A and B are both sentence symbols. Consider the context $\alpha = (\_\ A)$. Then, $\alpha[A]$ is 1-generable, but $\alpha[B]$ is not 1-generable and, therefore, A is not equivalent to B. Thus, G is simplified. □

Context equivalence partitions the set of nonterminals of a grammar into equivalence classes. The nonterminals in an equivalence class can be identified to yield a structurally equivalent grammar. For the construction of a context reduced grammar we refer to [8]. As we will see, there exist simplified, nonisomorphic, and structurally equivalent EOL grammars.

We now obtain

Theorem 5.7 *Let $G = (\{A, B\}, P, S)$ be a simplified UB grammar. Then, G is UB-universal if and only if (i) and (ii) hold.*

(i) $S = \{A, B\}$.

(ii) $P = \{A \rightarrow a \mid aa \mid B \mid BB; B \rightarrow A \mid AA\}$, $P = \{A \rightarrow a \mid aa; B \rightarrow A \mid AA \mid B \mid BB\}$, *or the roles of A and B are interchanged.*

Proof: If: Straightforward.
Only if: Condition (i) follows from Lemma 5.4.

It is clear that we must have productions with a and aa as their righthand sides. We first show that both must be generated by the *same* nonterminal. For, assume that $A \rightarrow a$ and $B \rightarrow aa$ are in P. (The case $A \rightarrow aa$ and $B \rightarrow a$ are in P is symmetric.) Consider an arbitrary context α, such that $\alpha[A]$ is d-generable. By replacing each nonterminal A and B at the frontier of $\alpha[A]$ by a and aa respectively, we obtain a terminal string in $L(G)$. Removing all labels yields a tree $t \in UB$ of height $d + 1$. Now, consider the node in t corresponding to the underscore in α. It is a unary node of depth one. Replacing this node by a binary node yields another tree $t' \in UB$ which is everywhere identical with t except for this node. Thus, t' must also belong

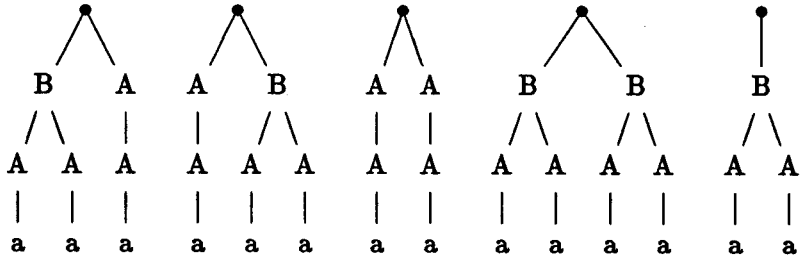


Figure 2: Reconstructing internal labels bottom-up.

to $T(G)$. Because G is invertible, this is only possible, if $\alpha[B]$ is d-generable. By symmetry we obtain in the same way that $\alpha[B]$ is d-generable implies $\alpha[A]$ is d-generable. Because d is arbitrary, we have $A \equiv B$—a contradiction.

Hence, we may assume that $A \to a \mid aa$ are in P. (The case of $B \to a \mid aa$ in P is symmetric.)

Next, we show that $B \to A$ is in P. For, assume that $B \to A$ is *not* in P. Because G has no useless nonterminals and A is the only nonterminal which generates terminal symbols, we must have $B \to AA$ in P. Furthermore, we must also have $A \to A$ in P, because otherwise arbitrarily high unary trees could not be generated by G.

Our assumptions imply that G contains the productions $A \to A \mid a \mid aa$ and $B \to AA$. This allows us to partially reconstruct the labels of syntax trees of a given structure in $T(G)$ bottom up as shown in Figure 2. Although we do not know the labels of the roots of these trees, we are able to show by induction that, for all $d \geq 0$ and for all contexts α,

$$\alpha[A] \text{ is } d\text{-generable if and only if } \alpha[B] \text{ is } d\text{-generable.}$$

The case $d = 0$ is clear, because both A and B are sentence symbols of G and the empty context α is the only context for which $\alpha[A]$ is 0-generable and $\alpha[B]$ is 0-generable, respectively.

Consider a context α such that $\alpha[A]$ is $(d+1)$-generable. Invertibility of G implies that we can uniquely infer the labels on the level d just above the leaves of $\alpha[A]$. Let X be the label of the node which generates the node with the label A at the position of the underscore in α. Replacing this label $X \in \{A, B\}$ by an underscore yields a context β such that $\beta[X]$ is d-generable and $frontier(\beta[X]) \Rightarrow frontier(\alpha[A])$ in G.

Now consider the partial reconstructions of syntax trees in G shown in Figure 2. We see that where A occurs on the second level we can have also B at the same position either alone, if A was alone, or with the same sibling

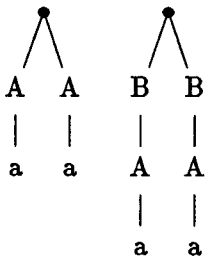


Figure 3: Partially reconstructed syntax trees.

A or B. Now either the same nonterminal X at level d which generated the A at the position of the underscore in α at level $d+1$ can also generate the B at this position in one step, or, if not, we know by the induction hypothesis that $\beta[Y]$ is d-generable and Y can generate the desired B in one step. In both cases we obtain that $\alpha[B]$ is $(d+1)$-generable. In the same way we can infer that $\alpha[A]$ is $(d+1)$-generable implies $\alpha[B]$ is $(d+1)$-generable, and the induction step is complete. However, this implies that $A \equiv B$—a contradiction. Therefore, $B \to A$ is in P.

Now consider the partially reconstructed syntax trees of Figure 3. In order to generate these trees we must have one of the following pairs of productions:

(1) $A \to AA$ and $A \to BB$;

(2) $A \to AA$ and $B \to BB$;

(3) $B \to AA$ and $A \to BB$; or

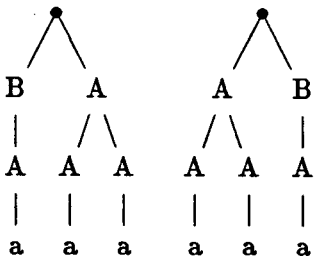
(4) $B \to AA$ and $B \to BB$.

Because G is invertible, we cannot have $A \to A$ in P. In order to generate arbitrarily high unary trees we must have either $A \to B$ or $B \to B$ in P. We consider both cases in turn.

Case 1: $A \to B$ in P.

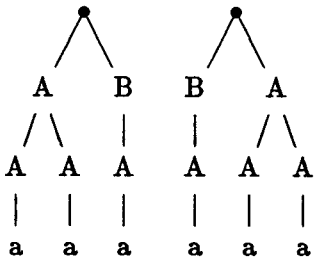
The four possibilities of productions with AA and BB as their right hand sides lead to the following four subcases.

(1.1) allows the following partial reconstructions of syntax trees:



As above we can conclude that $A \equiv B$—a contradiction.

(1.2) allows the following partial reconstructions of syntax trees:



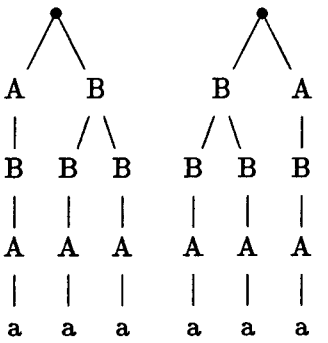
Again, we conclude $A \equiv B$—a contradiction.

(1.3) gives the following productions

$$A \rightarrow a \mid aa \mid B \mid BB$$
$$B \rightarrow A \mid AA$$

This is one of the possibilities claimed by the theorem.

(1.4) allows the following partial reconstructions of syntax trees:



Again, we conclude $A \equiv B$—a contradiction.

Case 2: $B \rightarrow B$ in P

It is again easy to see that among the four possibilities for productions with AA and BB as their right hand sides only the fourth (4) does not lead to a contradiction. In this case we obtain the following set of productions:

$$\begin{aligned} A &\rightarrow a \mid aa \\ B &\rightarrow A \mid AA \mid B \mid BB \end{aligned}$$

This is exactly the other possibility claimed in the theorem. □

The arguments which we used in the proof of Theorem 5.6 did not depend on the form of the productions but only on the number of nonterminals. So we also have:

Theorem 5.8 *Let* $G = (\{A, B\}, P, S)$ *be a simplified BT grammar. Then,* G *is BT-universal if and only if (i) and (ii) hold.*

(i) $S = \{A, B\}$

(ii) $P = \{A \rightarrow aa \mid aaa \mid BB \mid BBB; B \rightarrow AA \mid AAA\}, P = \{A \rightarrow aa \mid aaa; B \rightarrow AA \mid AAA \mid BB \mid BBB\}$, *or the roles of* A *and* B *are interchanged.*

References

[1] A.V. Aho, J.E. Hopcroft, and J.D. Ullman. *Data Structures and Algorithms.* Addison-Wesley Publishing Co., Reading, Mass., 1983.

[2] D.T. Lee, C.L. Liu, and C.K. Wong. $(g_0, \ldots, g_k)$-Trees and unary 0L systems. *Theoretical Computer Science*, 22:209–217, 1983.

[3] H.A. Maurer, Th. Ottmann, and H.-W. Six. Implementing dictionaries using binary trees of very small height. *Information Processing Letters*, 5:11–14, 1976.

[4] H.A. Maurer and D. Wood. Zur Manipulation von Zahlenmengen. *Angewandte Informatik*, 7:143–149, 1976.

[5] R. McNaughton. Parenthesis grammars. *Journal of the ACM*, 14:490–500, 1967.

[6] H. J. Olivié. *A Study of Balanced Binary Trees and Balanced One-Two Trees.* PhD thesis, Departement Wiskunde, Universiteit Antwerpen, Antwerp, Belgium, 1980.

[7] Th. Ottmann and H.-W. Six. Eine neue Klasse von ausgeglichenen Binärbäumen. *Angewandte Informatik*, 9:395–400, 1976.

[8] Th. Ottmann and D. Wood. Structural equivalence of E0L grammars. University of Waterloo, 1989.

[9] G. Rozenberg and A. Salomaa. *The Mathematical Theory of L Systems.* Academic Press, New York, 1980.

[10] K. Ruohonen. On equality of multiplicity sets of regular languages. *Theoretical Computer Science*, 36:113–117, 1895.

[11] A. Salomaa. *Formal Languages.* Academic Press, New York, 1973.

[12] D. Wood. A note on Lindenmayer systems, Szilard languages, spectra, and equivalence. *International Journal of Computer and Information Sciences*, 4:53–62, 1975.