Balanced Grammars and Their Languages

Jean Berstel[1] and Luc Boasson[2]

[1] Institut Gaspard Monge (IGM), Université Marne-la-Vallée,
77454 Marne-la-Vallée Cedex 2
berstel@univ-mlv.fr
[2] Laboratoire d'informatique algorithmique: fondements et applications (LIAFA),
Université Denis-Diderot,
75251 Paris Cedex 05
boasson@liafa.jussieu.fr

Abstract. Balanced grammars are a generalization of parenthesis grammars in two directions. First, several kind of parentheses are allowed. Next, the set of right-hand sides of productions may be an infinite regular language. XML-grammars are a special kind of balanced grammars. This paper studies balanced grammars and their languages. It is shown that there exists a unique minimal balanced grammar equivalent to a given one. Next, balanced languages are characterized through a property of their syntactic congruence. Finally, we show how this characterization is related to previous work of McNaughton and Knuth on parenthesis languages.

1 Introduction

Balanced grammars are extended context-free grammars of a special kind. They generate words over a set of parenthesis that are well-formed (i.e. Dyck words). The right-hand side of any production of a balanced grammar is well-formed in a sense to be described. Moreover, for each nonterminal, the set of right-hand sides of productions for this nonterminal is a regular set.

The motivation for studying balanced grammars is twofold. First, it appears that grammars describing XML-documents are special cases of balanced grammars. The syntactic properties of these grammars have been considered in [1]. Next, parenthesis grammars, as developed by McNaughton [8] and Knuth [6], also appear to be balanced grammars, but with finitely many productions and only one pair of parentheses.. Parenthesis grammars have many interesting syntactic and decision properties, and it is interesting to investigate whether these properties carry over to grammars with regular sets of productions and several pairs of parentheses. As we shall see, many constructs carry over, although the proofs are sometimes more involved. In the course of this investigation, we will consider how several well-known constructions for standard context-free grammars behave when the sets of productions is regular.

A context-free grammar will be called *regular* if, for each nonterminal, the set of right-hand sides of productions for this nonterminal is regular. If these

W. Brauer et al. (Eds.): Formal and Natural Computing, LNCS 2300, pp. 3–25, 2002.

sets are finite (the case of usual context-free grammars) the grammar is called finite context-free. A well-known exercise on context-free grammars shows that the language generated by a regular context-free grammar is context-free. Thus, extending the set of productions does not change the family of languages that is generated. On the contrary, questions about grammars may turn out to be more difficult in the case of regular context-free grammars. One example is given in Section 4 below, where it is shown that every grammar can be converted to a codeterministic grammar. This was proved by McNaughton in the case of parenthesis grammars, but appears to hold for general regular context-free grammars.

The paper is organized as follows. Section 2 and 3 introduce regular context-free grammars and balanced grammars. Section 4 is about codeterministic grammars. Section 5 groups elementary results, examples and undecidability results for balanced languages. In Section 6, it is shown that every codeterministic balanced grammar can be reduced to a minimal balanced grammar, and that this grammar is unique (Theorem 6.3 and 6.5). In Section 7, we show that balanced languages are closed under complement. This is a result that holds only within regular balanced grammars, and does not hold within the framework of parenthesis grammars. Section 8 presents a syntactic characterization of balanced language. These are well-formed languages such that the set of Dyck words intersects only a finite number of congruence classes for the syntactic congruence of the language. Although this property is undecidable, it is closely related to the decision procedure in Section 9 where balanced languages with bounded width are considered. Indeed, we show that this property always holds in the case of bounded width.

2 Regular Context-Free Grammars

A *regular context-free grammar* $G = (V, A, \mathcal{P})$ is defined as follows. The set V is the *finite* set of variables or non-terminals. The alphabet A is the terminal alphabet. The set $\mathcal{P}$ is the set of productions. For each variable X, the set

$$R_X = \{m \in (V \cup A)^* \mid (X \to m) \in \mathcal{P}\}$$

is a regular subset of $(V \cup A)^*$. It follows that the set $\mathcal{P}$ itself is regular. A convenient shorthand is to write

$$X \to R_X$$

The set R_X is the set of X-handles. The language generated by a variable is defined in the usual way. We consider grammars that may have several axioms.

Regular context-free grammars have been considered in particular by Conway. In his book [2], the theory of context-free languages is developed in this framework.

Example 2.1. Consider the regular grammar $G = (\{X\}, \{a, \bar{a}\}, \mathcal{P})$ where $\mathcal{P}$ is the set

$$X \to aX^*\bar{a}$$

It generates the set of *Dyck primes* over $\{a, \bar{a}\}$.

In the sequel, we simply say *context-free grammar* for a regular context-free grammar, and we say that a grammar is *finite* if it has a finite set of productions. For every (regular) context-free grammar, there exists a finite context-free grammar generating the same language. In particular, all these languages are context-free.

3 Balanced Grammars

The main purpose of this paper is to study balanced grammars, as defined below. As we shall see, these grammars are a natural extension of McNaughton's parenthesis grammars.

A context-free grammar $G = (V, T, \mathcal{P})$ is *balanced* if the two following restrictions hold. First, the terminal alphabet T has a decomposition $T = A \cup \bar{A} \cup B$, where $\bar{A} = \{\bar{a} \mid a \in A\}$ is a disjoint copy of A, and B is disjoint from A and from $\bar{A}$. Next, productions are of the form $X \longrightarrow am\bar{a}$, with $m \in (V \cup B)^*$. It follows that the regular sets R_X of X-handles admit a decomposition

$$R_X = \bigcup_{a \in A} aR_{X,a}\bar{a}$$

where

$$R_{X,a} = \{m \in (V \cup B)^* \mid X \to am\bar{a}\}$$

Of course, the sets $R_{X,a}$ are regular subsets of $(V \cup B)^*$. We write for short:

$$X \to \bigcup_{a \in A} aR_{X,a}\bar{a}$$

It appears useful to call letters in A *colors*, and to call the initial letter of the right-hand side of a production the color of the production.

If $B = \emptyset$, a balanced grammar is called *pure*. A language L is (pure) *balanced* if

$$L = \bigcup_{X \in W} L_G(X)$$

for some subset W of V.

A language over $A \cup \bar{A}$ is *well-formed* if it is a subset of the Dyck language over A. Clearly, any pure balanced language is well-formed, and the converse does not hold (see Example 5.8 below).

The set D_A of *Dyck primes* over $A \cup \bar{A}$ will play an important role. Let us recall that it is a prefix and a suffix code, and that every word $x \in D_A$ admits a unique factorization of the form $x = az_1 \cdots z_n\bar{a}$, where $a \in A$, $n \geq 0$ and $z_1, \ldots, z_n$ are Dyck primes. A *Dyck factor* of a word w is any factor x of w that is a Dyck prime.

The set D_A has strong synchronization properties. We state them in a lemma.

Lemma 3.1. (*i*) *If a Dyck prime z is a factor of a product $z_1 \cdots z_n$ of Dyck primes, then z is a factor of one of the z_i.*
(*ii*) *If a Dyck word $w \in D^*$ is a factor of a Dyck prime z, then $w = z$ or there exist Dyck words $x, y \in D^*$ and a letter $a \in A$ such that the Dyck prime $axwy\bar{a}$ is a factor of z.* □

Let us start with two simple examples of balanced languages.

Example 3.2. The language of *Dyck primes* over $\{a, \bar{a}\}$ is a pure balanced language, generated by

$$X \to aX^*\bar{a}$$

Example 3.3. The language D_A of *Dyck primes* over $T = A \cup \bar{A}$ is generated by the grammar

$$\begin{aligned} X &\to \textstyle\sum_{a\in A} X_a \\ X_a &\to aX^*\bar{a}, \qquad a \in A \end{aligned}$$

The variable X generates the language D_A which is well-formed. Although the present grammar is not balanced, the language D_A *is* a pure balanced language. Indeed, it suffices to replace X by $\sum_{a\in A} X_a$ in the second part, and to consider that every X_a is an axiom.

There exist several families of context-free grammars $G = (V, T, \mathcal{P})$ related to balanced grammars that have been studied in the past.

Parenthesis grammars have been studied in particular by McNaughton [8] and by Knuth [6]. Such a grammar is a balanced grammar where the alphabet A is a singleton (just one color), so $T = B \cup \{a, \bar{a}\}$, and with finitely many productions.

Bracketed grammars were investigated by Ginsburg and Harrison in [4]. The terminal alphabet T is the disjoint union of three alphabets A, $\bar{B}$ and C, and productions are of the form $X \longrightarrow am\bar{b}$, with $m \in (V \cup C)^*$. Moreover, there is a bijection between the set A of colors and the set of productions. Thus, in a bracketed grammar, every derivation step is marked.

Chomsky-Schützenberger grammars are used in the proof of the Chomsky-Schützenberger theorem (see e. g. [5]), even if they were never studied for their own. Here the terminal alphabet is of the form $T = A \cup \bar{A} \cup B$, and the productions are of the form $X \longrightarrow am\bar{a}$. Again, there is only one production for each color $a \in A$. So it is a special kind of balanced grammar with finite number of productions.

XML-grammars have been considered in [1]. They differ from all previous grammars by the fact that the set of productions is not necessarily finite, but regular. XML-grammars are balanced grammars. They are pure if all text elements are ignored. XML-grammars have the property that for each color $a \in A$, there is only one variable X such that the set $R_{X,a}$ is not empty. Expressed with colors, this means that all variables are monochromatic and all have different colors.

4 Codeterministic Grammars

A context-free grammar is called *codeterministic* if $X \to m$, $X' \to m$ implies $X = X'$. Codeterministic grammars are called backwards deterministic in [8].

In the next proposition, we show that codeterministic grammars can always be constructed. The main interest and use is for balanced grammars. In this case, the codeterministic grammar obtained is still balanced (Corollary 4.2). This also holds if the grammar one starts with is e.g. in Greibach Normal Form.

Proposition 4.1. *For every context-free grammar, there exists an equivalent codeterministic grammar context-free grammar that is effectively computable.*

The proof is adapted from the proof given in [8] for the case of finite context-free grammars. We give it here because it is an example of how an algorithm on finite grammars carries over to regular grammars.

Proof. Let $G = (V, A, \mathcal{P})$ be a context-free grammar. It will be convenient to denote here variables by small Greek letters such as α, β, σ because we will also deal with sets of variables. For each variable $\alpha \in V$, let R_α be the regular set of α-handles. Let $\mathcal{A}_\alpha$ be a deterministic automaton recognizing R_α. We first describe a transformation of the automaton $\mathcal{A}_\alpha$.

For any finite deterministic automaton $\mathcal{A} = (Q, q_0, F)$ over the alphabet $V \cup A$ with set of states Q, initial state q_0 and set of final states F, we define a *power automaton* $\mathcal{A}'$ recognizing words over the "big" alphabet $W = A \cup (2^V \setminus \emptyset)$ as follows. Each "big letter" B is either a nonempty subset of V, or a singleton $\{b\}$ composed of a terminal letter $b \in A$. The set of states of $\mathcal{A}'$ is 2^Q, its initial state is $\{q_0\}$, its final states are the sets $P \subset Q$ such that $P \cap Q \neq \emptyset$. The transition function is defined, for $P \subset Q$ and $B \in W$, by

$$P \cdot B = \{p \cdot b \mid p \in P, b \in B\}$$

This is quite similar to the well-known power set construction. A word $M = B_1 B_2 \cdots B_n$ over W is composed of "big letters" $B_1, \ldots B_n$. Given a word $M = B_1 B_2 \cdots B_n$ over W, we write $m \in M$ for $m \in (V \cup A)^*$ whenever $m = b_1 b_2 \cdots b_n$ with $b_i \in B_i$ for $i = 1, \ldots, n$. Observe that $B_i = \{b_i\}$ if $b_i \in A$. In other words, each $w \in A^*$ is can also be viewed as a "big" word.

For each $\alpha \in V$, let $\mathcal{A}_\alpha$ be a deterministic automaton recognizing R_α, let $\mathcal{A}'_\alpha$ be its power automaton, and let R'_α be the language (over W) recognized by $\mathcal{A}'_\alpha$. Then the following claims obviously hold.

(a) If $m \in R_\alpha$ and $m \in M$, then $M \in R'_\alpha$.

(b) Conversely, if $M \in R'_\alpha$, then there exists a word $m \in M$ such that $m \in R_\alpha$.

It follows from these claims that $M \in R'_\alpha$ if and only if there exists $m \in M$ with $m \in R_\alpha$. In other words, $M \in R'_\alpha$ if and only if $M \cap R_\alpha \neq \emptyset$.

For each word M over W, let $V(M)$ be the subset of V composed of the variables α such that M is recognized in the power automaton $\mathcal{A}'_\alpha$.

Thus

$$V(M) = \{\alpha \in V \mid M \cap R_\alpha \neq \emptyset\}$$

For each subset $U \subset V$, define the set

$$S_U = \{M \in W^* \mid U = V(M)\}$$

of words M such that $U = V(M)$. This means that $M \in S_U$ iff U is precisely the set of variable α such that $M \in R'_\alpha$ (or equivalently $M \cap R_\alpha \neq \emptyset$). The set S_U is regular, because it is indeed

$$S_U = \bigcap_{\alpha \in U} R'_\alpha \setminus \left(\bigcup_{\alpha \notin U} R'_\alpha \right) \tag{1}$$

We define now a new grammar G' as follows. Its set of variables is $\mathcal{V} = 2^V \setminus \emptyset$. The productions are

$$U \to S_U$$

The grammar is codeterministic because in a production $X \to M$, the handle M determines $V(M)$. It remains to prove that G' is equivalent to G. We prove that

$$L(G, \alpha) = \bigcup_{\alpha \in U} L(G', U) \tag{2}$$

The proof is in two parts. We first show that for $\alpha \in U$, one has

$$L(G', U) \subset L(G, \alpha)$$

Consider a word $w \in L(G', U)$ and a derivation $U \xrightarrow{k} w$ of length k.

If $k = 1$ then $w \in A^*$ and $U \longrightarrow w$. Thus w is in S_U. By Eq. 1, and because $\alpha \in U$, one has $w \in R_\alpha$. It follows that $w \in L(G, \alpha)$.

If $k > 1$, then $U \longrightarrow M \xrightarrow{k-1} t$ for some $M \in S_U$ and some terminal word t. Set $M = U_1 \cdots U_n$. Then $t = t_1 \cdots t_n$ and $U_i \xrightarrow{*} t_i$ for $i = 1, \ldots n$. By induction, one has $\alpha_i \xrightarrow{*} t_i$ for each i and for all $\alpha_i \in U_i$. Next, since $M \in S_U$, one has $M \in R'_\alpha$. Consequently there is some $m \in M \cap R_\alpha$. Setting $m = \alpha_1 \cdots \alpha_n$, one has $\alpha_i \in U_i$ and $\alpha \longrightarrow \alpha_1 \cdots \alpha_n$. It follows that $\alpha \xrightarrow{*} t$. This proves the inclusion.

Consider now the converse inclusion

$$L(G, \alpha) \subset \bigcup_{\alpha \in U} L(G', U)$$

This means that, for each word $w \in L(G, \alpha)$, there exists a set U containing α such that $w \in L(G', U)$.

We shall in fact prove the following, slightly more general property. Let $m \in (V \cup A)^*$. If $\alpha \xrightarrow{*} m$, then for every set M containing m, there exists a set U containing α such that $U \xrightarrow{*} m$.

Assume indeed that $\alpha \xrightarrow{\ell} m$. If $\ell = 1$, choose any "big word" M containing m and let $U = \{\gamma \mid M \in R'_\gamma\}$. Then $U \longrightarrow M$. Moreover α is in U because $m \in R_\alpha$. This proves the claim in this case.

Assume $\ell > 1$. Consider the last step of the derivation $\alpha \xrightarrow{\ell-1} x\beta y \longrightarrow m = xhy$, with $\beta \to h$ a production in G. Choose any "big word" M containing m.

Then $M = XHY$, where $|X| = |x|$, $|H| = |h|$, $Y| = |y|$. Then $x \in X$, $h \in H$, $y \in Y$. By the first part of the proof, there exists a set N containing β such that $N \longrightarrow H$. Consider now $Z = XNY$. This set contains $x\beta y$. By induction, there exists a set U such that $\alpha \in U$ and $U \xrightarrow{*} Z$ in the grammar G'. Consequently, $U \xrightarrow{*} M$. This finishes the proof. □

Corollary 4.2. *If a context-free grammar is balanced (pure balanced, in Greibach normal form, in two-sided Greibach normal form, is finite), there exists an equivalent codeterministic grammar that is of the same type.*

Proof. It suffices to observe that, in a "big word" constructed from a word, terminal letters remain unchanged, only variables are replaced by (finite) sets of variables. □

5 Elementary Properties and Examples

Balanced context-free grammars have some elementary features that are basic steps in proving properties of this family of grammars. Given an alphabet $A \cup \bar{A}$, we denote by D_A or by D the set of *Dyck primes* over this alphabet. Given an alphabet $A \cup \bar{A} \cup B$, where B is disjoint from $A \cup \bar{A}$, a *Motzkin word* is a word in the shuffle $D_A^* \sqcup\!\sqcup B^*$. It is not difficult to see that every Motzkin word has a unique factorization as a product of *Motzkin primes*. Motzkin primes are the words in the set

$$M = B \cup \bigcup_{a \in A} a(D_A^* \sqcup\!\sqcup B^*)\bar{a}$$

We are interested in the set

$$N = \bigcup_{a \in A} a(D_A^* \sqcup\!\sqcup B^*)\bar{a}$$

of *Motzkin-Dyck primes*

Lemma 5.1. *Let $G = (V, A \cup \bar{A} \cup B, \mathcal{P})$ be a balanced grammar. For each variable $X \in V$, the language $L(G, X)$ is a subset of N, and if G is pure, then $L(G, X)$ is a subset of D.*

Proof. The proof is straightforward by induction. □

There are only tiny differences between balanced and pure balanced grammars. Moreover, every balanced language is a homomorphic image of a pure balanced language. To get the pure language, it suffices to introduce a barred alphabet $\bar{B}$ and to replace each occurrence of a letter b by a word $b\bar{b}$. The grammar is modified by adding a new variable X_b for each b, with only the production $X_b \to b\bar{b}$. Finally, in all other productions, each b is replaced by X_b. The original language is obtained by erasing all letters in $\bar{B}$.

For this reason, we assume from now on that all balanced grammars are pure.

Lemma 5.2. *Let $G = (V, A \cup \bar{A}, \mathcal{P})$ be a balanced grammar. Assume that*

$$X \xrightarrow{*} az_1 \cdots z_n \bar{a}$$

for some letter $a \in A$ and Dyck primes $z_1, \ldots, z_n$. Then there exists a production $X \to aX_1 \cdots X_n \bar{a}$ in G such that $X_i \xrightarrow{*} z_i$ for $i = 1, \ldots, n$.

Proof. Assume $X \xrightarrow{*} az_1 \cdots z_n \bar{a}$. Then there is a production $X \to aY_1 \cdots Y_m \bar{a}$ such that $X \to aY_1 \cdots Y_m \bar{a} \xrightarrow{*} az_1 \cdots z_n \bar{a}$. Since $Y_1 \cdots Y_m \xrightarrow{*} z_1 \cdots z_n$, there exist words $y_1, \ldots, y_m$ such that $Y_i \xrightarrow{*} y_i$ and $y_1 \cdots y_m = z_1 \cdots z_n$. By Lemma 5.1, the words y_i are Dyck primes. Thus $m = n$ and $y_i = z_i$. □

Lemma 5.3. *Let L be the language generated by a balanced grammar $G = (V, A \cup \bar{A}, \mathcal{P})$. If $gud \in L$ for some words $g, d \in (A \cup \bar{A})^*$ and some Dyck prime $u \in D$, then there exists a variable X and an axiom S such that*

$$S \xrightarrow{*} gXd, \qquad X \xrightarrow{*} u$$

Moreover, if G is codeterministic, then the variable X with this property is unique.

Proof. The second part of the lemma is straightforward. If $gud \in L$, there is a left derivation $S \xrightarrow{*} gud$ for some axiom S. Let a denote the initial letter of u. Since letters in A appear only as initial letters in handles of productions, the step in the derivation where this letter is produced has the form

$$S \xrightarrow{*} gX\delta \longrightarrow ga m\bar{a}\delta \xrightarrow{*} gud$$

for some $m \in R_{X,a}$. Since $am\bar{a}\delta \xrightarrow{*} ud$, there is a factorization $ud = u'd'$ with $am\bar{a} \xrightarrow{*} u'$ and $\delta \xrightarrow{*} d'$. By Lemma 5.1, the word u' is a Dyck prime, and since $ud = u'd'$, and the set of Dyck primes is a prefix code, it follows that $u = u'$ and consequently $d = d'$. □

Lemma 5.4. *Let L be the language generated by a balanced grammar $G = (V, A \cup \bar{A}, \mathcal{P})$. If $gu_1 \cdots u_n d \in L$ for some words $g, d \in (A \cup \bar{A})^*$ and some Dyck primes $u_1, \ldots, u_n \in D$, then there exist variables $X_1, \ldots, X_n$ and an axiom S such that $S \xrightarrow{*} gX_1 \cdots X_n d$ and $X_i \xrightarrow{*} u_i$ for $i = 1, \ldots, n$.* □

Lemma 5.5. *Let $G = (V, A \cup \bar{A}, \mathcal{P})$ be a codeterministic balanced grammar. If X, Y are distinct variables, then $L(G, X)$ and $L(G, Y)$ are disjoint.*

Proof. Assume there are derivations

$$X \longrightarrow aX_1 \cdots X_n \bar{a} \xrightarrow{*} u, \quad Y \longrightarrow a'Y_1 \cdots Y_{n'} \bar{a}' \xrightarrow{*} u$$

for some word $u \in D$. The proof is by induction on the sum of the lengths of these two derivation. If $n + n' = 2$, then $n = n' = 1$, and $a = a'$. Thus $X \longrightarrow a\bar{a}$

and $Y \longrightarrow a\bar{a}$, and since G is codeterministic, $X = Y$. If $n + n' > 2$, then u has factorizations

$$u = ax_1 \cdots x_n\bar{a} = a'y_1 \cdots y_{n'}\bar{a}'$$

where $X_i \xrightarrow{*} x_i$, $Y_j \xrightarrow{*} y_j$. Clearly, $a = a'$, and because D is a prefix code, one has $n = n'$, $x_i = y_i$. By induction, if follows that $X_i = Y_i$, and by codeterminism one gets $X = Y$. □

5.1 More Examples

Example 5.6. Consider the grammars

$$\begin{array}{l} X \to aY^*\bar{a} \\ Y \to b\bar{b} \end{array} \qquad \text{and} \qquad \begin{array}{l} X \to aY \\ Y \to b\bar{b}Y \mid \bar{a} \end{array}$$

They clearly generate the same language $a(b\bar{b})^*\bar{a}$. The left grammar is infinite and balanced. Thus the language is balanced. The right grammar is finite and not balanced. It follows from a result of Knuth [6] that we will discuss later that there is no balanced grammar with a finite number of production generating this language.

Example 5.7. The language

$$L = \{b(a\bar{a})^n aa\bar{a}\bar{a}(a\bar{a})^n\bar{b} \mid n > 0\}$$

is well-formed but not balanced. Assume the contrary. Then, for each $n > 0$, there is a word $m_n \in V^*$ such that

$$S \to bm_n\bar{b} \xrightarrow{*} b(a\bar{a})^n aa\bar{a}\bar{a}(a\bar{a})^n\bar{b}$$

Moreover, the word m_n has the form

$$m_n = X_1 \cdots X_n Z Y_1 \cdots Y_n$$

where $X_i \to a\bar{a}$, $Y_i \to a\bar{a}$, $Z \xrightarrow{*} aa\bar{a}\bar{a}$. Each word m_n is in the regular language $R_{S,a}$, and a pumping argument gives the contradiction.

Example 5.8. Consider the grammar

$$\begin{array}{l} X \to aY^*\bar{a} \\ Y \to b\bar{b}Yc\bar{c} \mid \varepsilon \end{array}$$

The language is balanced if and only if $b = c$. Indeed, if $b = c$, then the language is generated by the grammar

$$\begin{array}{l} X \to a(ZZ)^*\bar{a} \\ Z \to b\bar{b} \end{array}$$

If $b \neq c$, the language is $\{a(b\bar{b})^n(c\bar{c})^n\bar{a} \mid n \geq 0\}$, and an argument similar to Example 5.7 shows that it is not balanced.

Example 5.9. The grammar

$$\begin{aligned} X_0 &\to Ya\bar{a} \\ X &\to aY\bar{a} \mid aa \\ Y &\to aX\bar{a}\bar{a}aa\bar{a} \mid aY\bar{a}\bar{a}\bar{a}aX\bar{a} \end{aligned}$$

generates a balanced language. It was used by Knuth ([6]) to demonstrate how his algorithm for the effective construction of a balanced grammar works.

5.2 Decision Problems

In this section, we state two decidability results. There are other decision problems that will be considered later. The following result was proved in [1]. It will be used later.

Theorem 5.10. *Given a context-free language L over an alphabet $A \cup \bar{A}$, it is decidable whether L is a subset of the set D_A of Dyck primes over $A \cup \bar{A}$.*

The following result is quite similar to a proposition in [1]. The proof differs slightly, and is included here for sake of completeness.

Theorem 5.11. *It is undecidable whether a language L is balanced.*

Proof. Consider the Post Correspondence Problem (PCP) for two sets of words $U = \{u_1, \ldots, u_n\}$ and $V = \{v_1, \ldots, v_n\}$ over the alphabet $C = \{a, b\}$. Consider a new alphabet $B = \{a_1, \ldots, a_n\}$ and define the sets L_U and L_V by

$$L_U = \{a_{i_1} \cdots a_{i_k} h \mid h \neq u_{i_k} \cdots u_{i_1}\} \quad L_V = \{a_{i_1} \cdots a_{i_k} h \mid h \neq v_{i_k} \cdots v_{i_1}\}$$

Recall that these are context-free, and that the set $L = L_U \cup L_V$ is regular iff $L = B^*C^*$. This holds iff the PCP has no solution.

Set $A = \{a_1, \ldots, a_n, a, b, c\}$, and define a mapping $\hat{w}$ from A^* to $(A \cup \bar{A})$ by mapping each letter d to $d\bar{d}$.

Consider words $\hat{u}_1, \ldots, \hat{u}_n, \hat{v}_1, \ldots, \hat{v}_n$ in $\{a\bar{a}, b\bar{b}\}^+$ and consider the languages

$$\hat{L}_U = \{a_{i_1}\bar{a}_{i_1} \cdots a_{i_k}\bar{a}_{i_k} h \mid h \neq \hat{u}_{i_k} \cdots \hat{u}_{i_1}\}$$

and

$$\hat{L}_V = \{a_{i_1}\bar{a}_{i_1} \cdots a_{i_k}\bar{a}_{i_k} h \mid h \neq \hat{v}_{i_k} \cdots \hat{v}_{i_1}\}$$

Set $\hat{L} = c(\hat{L}_U \cup \hat{L}_V)\bar{c}$. Assume $\hat{L}$ is a balanced language, generated by some balanced grammar with set of axioms W, and consider the set $R = \bigcup_{X \in W} R_{X,c}$. Since each word in $\hat{L}_U \cup \hat{L}_V$ is a product of two-letter Dyck primes, the set R is equal to $L_U \cup L_V$, up to a straightforward identification. Thus $L_U \cup L_V$ is regular which in turn implies that the PCP has no solution. Conversely, if the PCP has no solution, $L_U \cup L_V$ is regular which implies that $L_U \cup L_V = B^*C^*$, which implies that $\hat{L} = c\hat{B}^*\hat{C}^*\hat{c}$, showing that $\hat{L}$ is balanced. □

6 Minimal Balanced Grammars

The aim of this section is to prove the existence of a minimal balanced codeterministic grammar for every balanced context-free grammar, and moreover that this grammar is unique up to renaming. This is the extension, to regular grammars with several types of parentheses, of a theorem of McNaughton [8].

Let G be a balanced codeterministic grammar generating a language $L = L(G)$, and let H be the set of axioms, i.e. $L = \cup_{S\in H} L(G, S)$.

A *context* for the variable X is a pair (g, d) of terminal words such that $S \xrightarrow{*} gXd$ for some axiom $S \in H$. The set of contexts for X is denoted by $C_G(X)$, or $C(X)$ if the grammar is understood. The *length* of a context (g, d) is the integer $|gd|$. Two variables X and Y are *equivalent*, and we write $X \sim Y$ if and only if they have same contexts, that is if and only if $C(X) = C(Y)$.

Proposition 6.1. *Given a balanced codeterministic grammar G, there exists an integer N such that $X \sim Y$ if and only if they have same contexts of length at most N.*

The proof will be an easy consequence of the following construction.

For any pair (g, d) of terminal words, we consider the set $W = W(g, d)$ of the variables that admit (g, d) as a context. Thus $X \in W$ if and only if $(g, d) \in C(X)$.

Lemma 6.2. *Let G be a balanced codeterministic grammar G. There exists an integer N with the following property. For any pair (g, d) of terminal words, there exists a pair (g', d') of length at most N such that $W(g, d) = W(g', d')$.*

Proof of Proposition 6.1. Assume that X and Y have the same contexts of length N. Let (g, d) be any context for X, and set $W = W(g, d)$. By definition, X is in W. Next, there exists a pair (g', d') with $|g'd'| \leq N$ such that $W = W(g', d')$. Since X and Y have the same contexts of length N, and since (g', d') is a context for X, it is also a context for Y, and consequently Y is in W. This shows that every context for X is also a context for Y. □

Proof of the lemma. Consider the set $W = W(g, d)$. The construction is in three steps.

For every X in W, there is a derivation $S \xrightarrow{*} gXd$ for some axiom $S \in H$. Clearly, gd is well-formed. Moreover, since the grammar is balanced, the words g and d have the form $g = a_1 g_1 \cdots a_n g_n$, $d = d_n \bar{a}_n \cdots d_1 \bar{a}_1$, where $g_1, \ldots, g_n$, $d_1, \ldots d_n$ are (products of) Dyck words. Thus every g_i is a product of Dyck primes, and similarly for every d_j. Because G is codeterministic, there is a factorization of the derivation into $S \xrightarrow{*} a_1 M_1 \cdots a_n M_n X M'_n \bar{a}_n \cdots M'_1 a_1$ where each M_i and M'_j is a product of variables, and $M_i \xrightarrow{*} g_i$, $M'_j \xrightarrow{*} d_j$. For each of the variables appearing in these products, we choose a Dyck prime of minimal length that is generated by this variable, and we replace the corresponding factor in g and d by this word of minimal length. Denote by N_0 the greatest of these minimal lengths. Then (g, d) is replaced by pair (g', d') of the form $g' = a_1 g'_1 \cdots a_n g'_n$, $d = d'_n \bar{a}_n \cdots d'_1 \bar{a}_1$ with the property that each g'_i, d'_j, is a product of Dyck primes of length at most N_0. There may be many such Dyck primes, but they are all

small. Thus $W(g,d) = W(g',d')$, and we may assume that the initial (g,d) satisfies the property of having only small Dyck primes.

In the second step, we compute an upper bound for n. Observe that this integer is independent of the variable X chosen in W and also independent of the actual axiom. Fore each X in W, there is a path in the derivation tree from the axiom S to X. This path has $n+2$ nodes (S and X included), and each of the internal nodes of the path produces one pair $(a_i, \bar{a}_i)$ in the factorizations of g and d. Assume that there are h variables in W. Then there are h different paths. Considering all these paths, one get h-tuples of variables, which are the labels of the internal nodes at depth 1, 2,..., n for these paths. If n is greater than $h^{\|V\|+1}$ then two of these tuples are componentwise identical, and all derivation trees can be pruned simultaneously, without changing W. Thus, one may replace (g,d) by a pair such that $n \leq \|V\|^{\|V\|}$.

After these two steps, we know that $g = a_1 g_1 \cdots a_n g_n$, $d = d_n \bar{a}_n \cdots d_1 \bar{a}_1$, with n not too big and each g_i, d_j product of small Dyck primes. The number of primes in say $g_i d_i$ is exactly the number of variables minus 1 in the right-hand side of the i-th production on the path from the axiom S to the variable X. More precisely, assume that a production is $Z \to a_i \gamma Y \delta \bar{a}$, with $\gamma \xrightarrow{*} g_i$, $\delta \xrightarrow{*} d_i$. Then the number of Dyck primes in g_i is $|\gamma|$, and similarly for d_i. There may be several of these productions at level i, but for each of these productions, the handle $a_i \gamma Y \delta \bar{a}$ is the same up to possibly the variable Y. Each of these handles in in some fixed regular set, determined by the variable Z which also may change. Since there are only finitely many regular sets, it is clear that γ and δ may be chosen of small length. It follows that in each g_i, d_j the number Dyck primes they factor into may be bounded by a constant depending only on the grammar. This finishes the proof. □

A balanced codeterministic grammar is *reduced* if two equivalent variables are equal.

Theorem 6.3. *A balanced codeterministic grammar is equivalent to a balanced codeterministic reduced grammar.*

We start with a lemma of independent interest.

Lemma 6.4. *Let $X \to aX_1 \cdots X_n \bar{a}$ be a production of a balanced codeterministic grammar G. For all variables $Y_1 \sim X_1$, ..., $Y_n \sim X_n$, there exists a variable $Y \sim X$ such that $Y \to aY_1 \cdots Y_n \bar{a}$ is a production of G.*

Proof. Consider indeed a derivation

$$S \xrightarrow{*} gXd \longrightarrow gaX_1 \cdots X_n \bar{a} d \xrightarrow{*} gax_1 \cdots x_n \bar{a} d$$

where $X_i \xrightarrow{*} x_i$ for $i = 1, \ldots, n$. The pair $(ga, x_2 \cdots x_n \bar{a} d)$ is a context for X_1, thus also for Y_1. Consequently, there is a derivation

$$S_1 \xrightarrow{*} gaY_1 x_2 \cdots x_n \bar{a} d \xrightarrow{*} gay_1 x_2 \cdots x_n \bar{a} d$$

for some axiom S_1 and some word y_1 with $Y_1 \xrightarrow{*} y_1$. Since the grammar is codeterministic, it follows that $S_1 \xrightarrow{*} gay_1 X_2 x_3 \cdots x_n \bar{a} d$. Thus $(gay_1, x_3 \cdots x_n \bar{a} d)$ is

a context for X_2 (and for Y_2), and as before, there is a word y_2 with $Y_2 \xrightarrow{*} y_2$ such that, for some axiom S_2, one has

$$S_1 \xrightarrow{*} gay_1Y_2 \cdots x_n\bar{a}d \xrightarrow{*} gay_1y_2x_3 \cdots x_n\bar{a}d$$

Continuing in this way, we get a derivation

$$S' \xrightarrow{*} gay_1 \cdots y_n\bar{a}d$$

where $Y_i \xrightarrow{*} y_i$ for $i = 1, \ldots, n$. Since the grammar is codeterministic, it follows that

$$S' \xrightarrow{*} gaY_1 \cdots Y_n\bar{a}d$$

and since the grammar is balanced, this derivation decomposes into

$$S' \xrightarrow{*} gYd \longrightarrow gaY_1 \cdots Y_n\bar{a}d$$

for some production $Y \to aY_1 \cdots Y_n\bar{a}$. Observe that (g, d) is a context for Y. It follows easily that $X \sim Y$. □

Proof of Theorem 6.3. Let G be a balanced codeterministic grammar, and define a quotient grammar $G/\sim$ by identifying equivalent variables in G. More precisely, the variables in the quotient grammar are the equivalence classes of variables in G. Denote the equivalence class of X by $[X]$. The productions of $G/\sim$ are all productions $[X] \to a[X_1] \cdots [X_n]\bar{a}$, where $X \to aX_1 \cdots X_n\bar{a}$ is a production in G. Observe that the sets of productions of $G/\sim$ are still regular.

Note that if $X \sim Y$ in G and X is an axiom, then Y also is an axiom, because X is an axiom iff $(\varepsilon, \varepsilon)$ is a context for X. Thus the axioms in $G/\sim$ are equivalence classes of axioms in G.

Set $L = L(G, H)$ and $L' = L(G/\sim, H/\sim)$. It is easily seen that $L \subset L'$. Indeed, whenever $X \xrightarrow{*} u$ in G, then $[X] \xrightarrow{*} u$ in $G/\sim$. Conversely, suppose $[X] \xrightarrow{k} u$ in $G/\sim$. We show that there exists Y in $[X]$ such that $Y \xrightarrow{*} u$. This clearly holds if $k = 1$. If $k > 1$, then $[X] \longrightarrow a[X_1] \cdots [X_n]\bar{a} \xrightarrow{k-1} ax_1 \cdots x_nbara$ with $[X_i] \xrightarrow{*} x_i$. By induction, there exist variables Y_i in $[X_i]$ such that $Y_i \xrightarrow{*} x_i$ in G. Moreover, by the previous lemma, there exists a production

$$Y \to aY_1 \cdots Y_n\bar{a}$$

in G for some Y in $[X]$. Thus $Y \xrightarrow{*} u$. This proves the claim. It follows that if $u \in L'$, then $u \in L$. □

Before stating the next result, it is convenient to recall the syntactic congruence of a language. Given a language L, the context of a terminal word u is the set $C_L(u) = \{(g, d) \mid gud \in L\}$. Observe that this is independent of the device generating L. The syntactic congruence $\equiv_L$ is defined by $x \equiv_L y$ iff $C_L(x) = C_L(y)$. This congruence will be considered later.

Theorem 6.5. *Two equivalent reduced grammars are the same up to renaming of the variables.*

Proof. Let G be a reduced grammar generating the language L. If X is a variable of X and $X \xrightarrow{*} u$, then $C_G(X) = C_L(u)$. Indeed, if $gud \in L$, there is a derivation $S \xrightarrow{*} gud$ for some axiom. This can be factorized into $S \xrightarrow{*} gYd \xrightarrow{*} gud$ for some variable Y because G is balanced, and $Y = X$ because G is codeterministic. Thus (g, d) is a context for X. The converse inclusion is clear.

Consider another reduced grammar G' also generating the language L. Let X be a variable in G, let $u \in L(G, X)$ and let (g, d) be a context for X. Then $gud \in L$. Thus, there exists a derivation $S' \xrightarrow{*} gud$ in G'. Since u is a Dyck prime and G' is balanced, there is a variable X' in G' such that $u \in L(G', X')$. Moreover, (g, d) is also a context X' (in G'). By the previous remark, $C_G(X) = C_L(u) = C_{G'}(X')$. Consider another word v in $L(G, X)$. Then there is a variable Y' such that $v \in L(G', Y')$. However $C_{G'}(X') = C_{G'}(Y')$ and, since G' is reduced, $X' = Y'$. Thus, to each variable X in G there corresponds a unique variable X' in G' that has same contexts. It follows easily that $L(G, X) = L(G', X')$.

It remains to show that the productions are the same. For this, consider a production $X \to aY_1 \cdots Y_n\bar{a}$ in G. Then there are words $u_1, \ldots, u_n$ such that $X \xrightarrow{*} au_1 \cdots u_n\bar{a}$, $Y_i \xrightarrow{*} u_i$ in G. In the grammar G', there is a variable X' such that $X' \xrightarrow{*} au_1 \cdots u_n\bar{a}$. Since G' is balanced and codeterministic there are variables Y'_i such that $X' \to aY'_1 \cdots Y'_n\bar{a}$ and $Y'_i \xrightarrow{*} u_i$ in G'. This finishes the proof. □

Observe that a reduced grammar is minimal in the sense that it has a minimal number of variables.

7 Complete Balanced Grammars

In this section, we consider complementation. Any balanced language is a subset of the language D of Dyck primes. Thus, complementation of a balanced language makes only sense only with respect to the set D.

Proposition 7.1. *The complement of a balanced language with respect to the set of Dyck primes is balanced.*

It is straightforward that balanced languages are closed under union. They are therefore also closed under intersection.

Proof. Let L be a balanced language and let G be a balanced codeterministic grammar generating it, so that $L = L(G, W)$ for some subset W of the set of variables V. Set also $M = L(G, V)$. Then M is precisely the set of Dyck factors of words in L. Hence, $D \setminus M$ is the possibly empty set of Dyck primes that are not Dyck factors of words in L. We show that $D \setminus M$ is balanced.

Consider first the subset N of $D \setminus M$ composed of words x such that any proper Dyck factor y of x is in M. Thus

$$N = (D \setminus M) \setminus (A \cup \bar{A})^+(D \setminus M)(A \cup \bar{A})^+$$

A word is in $D \setminus M$ if and only if it has a Dyck factor in N.

A word $x \in N$ has the form $x = ay_1 \cdots y_n\bar{a}$, where $y_1, \ldots, y_n \in M$. Thus, there is a derivation

$$aX_1 \cdots X_n\bar{a} \xrightarrow{*} x$$

and the word $aX_1 \cdots X_n\bar{a}$ is not a handle in G. Conversely, if $aX_1 \cdots X_n\bar{a}$ is not a handle, then any word it generates is not in M because the grammar is codeterministic. Set $U_a = \bigcup_{X \in V} R_{X,a}$, consider the grammar G' obtained by adding a variable Φ and the productions

$$\Phi \to \bigcup_{a \in A} a(V^* \setminus U_a)\bar{a}$$

Then $N = L(G', \Phi)$.

Consider the grammar G'' obtained form G' by adding the productions

$$\Phi \to \bigcup_{a \in A} a(V + \Phi)^*\Phi(V + \Phi)^*\bar{a}$$

Since a word is in $D \setminus M$ if and only if it has a Dyck factor in N, one has $D \setminus M = L(G'', \Phi)$.

Observe finally that, in view of codeterminism,

$$D \setminus L = D \setminus M \cup \bigcup_{X \in V \setminus W} L(G, X)$$

This finishes the proof. □

A balanced grammar G with set of variables V is *complete* if

$$D = \bigcup_{X \in V} L(G, X)$$

Proposition 7.2. *For each balanced codeterministic grammar G, there exists a balanced complete codeterministic grammar G' with at most one additional variable Φ such that $L(G, X) = L(G', X)$ for all variables $X \neq \Phi$.*

Proof. This is an immediate consequence of the proof of the previous proposition, since the grammar G'' constructed in that proof is indeed complete. □

As a consequence, if G is a minimal grammar for a language L and G' is minimal for $D \setminus L$ than G and G' have the same number of variables, up to at most one.

8 A Characterization

We have recalled (Theorem 5.10) that it is decidable whether a context-free language L is well-formed, that is whether L is a subset of a set of Dyck primes. We also have seen (Theorem 5.11) that it is undecidable whether L is balanced,

that is whether there exists a (regular) balanced grammar generating L. In the case of a single pair of parentheses, a remarkable result of Knuth [6] shows on the contrary that, given a finite context-free grammar generating L, it is decidable whether there exists an equivalent finite balanced grammar generating the context-free language L. Moreover, Knuth gives an algorithm for constructing a finite balanced grammar from a given finite context-free grammar, if such a grammar exists.

The purpose of this section is investigate this relationship. More precisely, we shall prove a property that is equivalent for a language to be balanced. This property is of course undecidable. However, it trivially holds for languages generated by finite balanced grammars. In this way, we have a characterization that in some sense explains why Knuth's algorithm works, and why it cannot work in the general case.

Recall that the syntactic congruence $\equiv_L$ of a language L is defined by $x \equiv_L y$ iff $C_L(x) = C_L(y)$. Here $C_L(u) = \{(g, d) \mid gud \in L\}$ is the set of *contexts* of u in L. The equivalence class of u is denoted $[u]_L$ or $[u]$ if L is understood. Any language is a union of congruence classes for its syntactic congruence. It is well known that a language is regular if and only if its syntactic congruence has a finite number of equivalence classes.

A language L will be called *M-finite*, where M is a language if the number of equivalence classes of $\equiv_L$ intersecting M is finite. We will be concerned with languages that are D-finite or D^*-finite. Since D is a subset of D^*, any D^*-finite language is also D-finite. We will see that in some special cases, the converse also holds.

Observe that for a given (balanced) language L, the set of Dyck primes needs not to be a union of equivalence classes of $\equiv_L$. Consider indeed the language

$$L = \{aab\bar{b}\bar{a}ab\bar{b}\bar{a}\bar{a}, aab\bar{b}a\bar{a}b\bar{b}\bar{a}\bar{a}\}$$

The pair $(aab\bar{b}, b\bar{b}\bar{a}\bar{a})$ is the only context of both words $a\bar{a}$ and $\bar{a}a$. So they are equivalent for $\equiv_L$. However, $a\bar{a}$ is a Dyck word and $\bar{a}a$ is not.

Theorem 8.1. *A language L over $A \cup \bar{A}$ is balanced if and only if it is well-formed and D^*-finite.*

Proof. Assume first that L is well-formed and D^*-finite. We construct a balanced grammar generating L. Since D is a subset of D^*, the language L is also D-finite. Let V be a finite set of variables in bijection with the equivalence classes intersecting D. For $u \in D$, denote by $X_{[u]}$ the variable associated to the equivalence class $[u]$. Conversely, let $[X]$ be the equivalence class of $\equiv_L$ associated to X. For $X \in V$ there is a word $u \in D$ such that $X = X_{[u]}$ and $[X] = [u]$.

Each word w in D^* has a unique factorization $w = u_1 \cdots u_n$ with $u_i \in D$. We define a word $\phi(w)$ over V associated to w by $\phi(w) = X_{[u_1]} \cdots X_{[u_n]}$. The mapping ϕ is an isomorphism from D^* onto V^*. We consider the grammar defined by the productions $X \to aR_{X,a}\bar{a}$, where

$$R_{X,a} = \{\phi(w) \mid aw\bar{a} \in D \cap [X]\}$$

and with axioms $\{X_{[u]} \mid u \in L\}$. This grammar generates L. Indeed, it is easily checked that variable X generates $[X] \cap D$. Thus $X_{[u]}$ generates the class $[u] \cap D$, for $u \in D$. Thus if the sets $R_{X,a}$ are regular, the grammar is balanced.

Consider a fixed $X \in V$ and a letter $a \in A$. Denote by $\approx$ the syntactic congruence of $R_{X,a}$. Thus for $p, q \in V^*$, one has $p \approx q$ iff $rps \in R_{X,a} \Leftrightarrow rqs \in R_{X,a}$.

Let p, q be words in V^* and let y, z be words in D^* such that $\phi(y) = p$, $\phi(z) = q$. Assume $y \equiv_L z$. Let $r, s \in V^*$ be such that $rps \in R_{X,a}$. Choose g, d such that $\phi(g) = r$, $\phi(d) = s$. Then $agyd\bar{a} \in [X]$. Consequently $agzd\bar{a} \in [X]$, showing that $rqs \in R_{X,a}$, and therefore $p \approx q$. This shows that to each equivalence class of $\equiv_L$ intersecting D^* corresponds one equivalence class of $R_{X,a}$. Since there are finitely many of the former, there are finitely may of the second, and $R_{X,a}$ is regular.

Conversely, assume now that L is balanced. Then it is of course well-formed. Consider a codeterministic balanced grammar G generating L. Let $u \in D^*$ be a Dyck word that is a factor of some word in L, and set $u = v_1 \cdots v_n$, with $v_1, \ldots, v_n \in D$. There exists a unique word $X_1 \cdots X_n \in V^*$ such that $S \xrightarrow{*} gX_1 \cdots X_n d$ for some words g, d and some axiom S, and $X_i \xrightarrow{*} v_i$. We denote this word $X_1 \cdots X_n$ by $X(u)$. Define an equivalence relation on words in D^* by $u \sim v$ if and only if $X(u) \equiv_{R_{X,a}} X(v)$ for all $X \in V$ and $a \in A$. Here $\equiv_{R_{X,a}}$ is the syntactic congruence of the language $R_{X,a}$. Since the sets $R_{X,a}$ are regular, there are only finitely many equivalence class for $\sim$. We show that $u \sim v$ implies $u \equiv_L v$. This shows that the set of Dyck words that are factors of words in L are contained in a finite number of classes for $\equiv_L$. The other Dyck words all have empty set of contexts for L, and therefore are in the same class. This proves the proposition.

Assume $gud \in L$. Then there exists a unique derivation of the form

$$S \xrightarrow{*} g_1 X d_1, \qquad X \to aZ_1 \cdots Z_p X(u) Y_1 \cdots Y_q \bar{a}$$

such that $Z_1 \cdots Z_p \xrightarrow{*} g_2$, $Y_1 \cdots Y_q \xrightarrow{*} d_2$, and $g = g_1 a g_2$, $d = d_2 \bar{a} d_1$. Observe that $(Z_1 \cdots Z_p, Y_1 \cdots Y_q)$ is a context for the word $X(u)$ in the language $R_{X,a}$. Since $u \sim v$, it is also a context for $X(v)$. Thus $X \to aZ_1 \cdots Z_p X(v) Y_1 \cdots Y_q \bar{a}$ whence $S \xrightarrow{*} gvd$, showing that $gvd \in L$. □

Observe that it is undecidable, whether a well-formed (even context-free) language L, is D^*-finite. Indeed, by the theorem, this is equivalent for L to be balanced, and this latter property is undecidable (Theorem 5.11).

9 Bounded Width

In the sequel, we describe a condition, the bounded width property, that implies the existence of a balanced grammar.

Let L be a well-formed language over $A \cup \bar{A}$. We denote by $F(L)$ the set of factors of words in L. Given $N \geq 0$, we denote by $D^{(N)} = \{\varepsilon\} \cup D \cup \cdots \cup D^N$ the

set of product of at most N Dyck primes. The language L has *bounded width* if there exist $N \geq 0$ such that

$$F(L) \cap D^* \subset D^{(N)}$$

This means that every Dyck word that is a factor of a word in L is a product of at most N Dyck primes. The smallest N with this property is the width of L.

Example 9.1. The language $L = \{ab^n\bar{b}^n\bar{a} \mid n > 0\}$ has width 1.

Example 9.2. The language $L = \{a(b\bar{b})^n(c\bar{c})^n\bar{a} \mid n > 0\}$ has unbounded width.

We recall without proof a result from [1] (Theorem 6.1).

Proposition 9.3. *Given a well-formed context-free language L, it is decidable whether L has bounded width.*

Bounded width has many implications. As already mentioned, if a well-formed language L is D^*-finite, then it is D-finite. Bounded width implies the converse.

Proposition 9.4. *Let L be a well-formed language with bounded width. If L is D-finite, then it is D^*-finite.*

Proof. Let q be the number of equivalence classes of L intersecting D. Let N be the width of L. Let $u = u_1 \cdots u_n \in D^*$, with $u_1, \ldots, u_n \in D$. By a general result on congruences,

$$[u_1] \cdots [u_n] \subset [u]$$

If $n > N$, then u is the equivalence class of words that are not factors of L. Otherwise, $[u]$ contains at least one of the $q + q^2 + \cdots q^N$ products of equivalence classes. Thus the number of equivalence classes of L intersecting D^* is bounded by this number. □

The proposition is false if the width is unbounded.

Example 9.5. Consider the language $L = \{a(b\bar{b})^n(c\bar{c})^n\bar{a} \mid n > 0\}$ of the preceding example. There are just for classes of the syntactic congruence of L intersecting D. Their intersections with D are L, $\{b\bar{b}\}$, $\{c\bar{c}\}$, and the set $D \setminus F(L)$ of Dyck primes which are not factors of words of L. On the contrary, there are infinitely many equivalence classes intersecting D^*. For instance, each of the $(b\bar{b})^n$ is in a separate class, with $(a, (c\bar{c})^n\bar{a})$ as a context.

Another property resulting from bounded width is the following.

Proposition 9.6. *Le G be a balanced grammar generating a language L with bounded width. Then G is finite.*

Proof. Let $G = (V, A \cup \bar{A}, \mathcal{P})$ be a balanced grammar with productions

$$X \to \bigcup_{a \in A} aR_{X,a}\bar{a}$$

Assume that a language $R_{X,a}$ is infinite. Then, for arbitrarily great n, there are derivations $X \xrightarrow{*} az_1 \cdots z_n\bar{a}$, and since these words are factors of L, the language L has unbounded width. Thus all $R_{X,a}$ are finite. □

We shall prove the following proposition.

Proposition 9.7. *A well-formed context-free language with bounded width is D-finite.*

In view of Theorem 8.1 and Proposition 9.4, we get

Corollary 9.8. *A well-formed context-free language with bounded width is balanced.*

In fact, we have

Theorem 9.9. *Let L be a well-formed context-free language. Then L has bounded width if and only if L is generated by a finite balanced grammar. Moreover, the construction of the grammar is effective.*

The rest of the paper is concerned with the proof of Proposition 9.7.

We need some notation. The *Dyck reduction* is the semi-Thue reduction defined by the rules $a\bar{a} \to \varepsilon$ for $a \in A$. A word is *reduced* or *irreducible* if it cannot be further reduced, that means if it has no factor of the form $a\bar{a}$. Every word w reduces to a unique irreducible word denoted $\rho(w)$. We also write $w \equiv w'$ when $\rho(w) = \rho(w')$. If w is a factor of some Dyck prime, then $\rho(w)$ has no factor of the form $a\bar{b}$, for $a, b \in A$. Thus $\rho(w) \in \bar{A}^*A^*$.

In the sequel, G denotes a reduced finite context-free grammar over $T = A \cup \bar{A}$, generating a language L. For each variable X, we set

$$\mathrm{Irr}(X) = \{\rho(w) \mid X \xrightarrow{*} w, w \in T^*\}$$

This is the set of reduced words of all words generated by X. If L is well-formed, then $\mathrm{Irr}(S) = \{\varepsilon\}$ for every axiom S. Moreover, $\mathrm{Irr}(X)$ is finite for each variable X. Indeed, consider any derivation $S \xrightarrow{*} gXd$ with $g, d \in T^*$. Any $u \in \mathrm{Irr}(X)$ is of the form $u = \bar{x}y$, for $x, y \in A^*$. Since $\rho(gud) = \rho(\rho(g)u\rho(d)) = \varepsilon$, the word x is a suffix of $\rho(g)$, and $\bar{y}$ is a prefix of $\rho(d)$. Thus $|u| \leq |\rho(g)| + |\rho(d)|$, showing that the length of the words in $\mathrm{Irr}(X)$ is bounded.

A grammar is *qualified* if $\mathrm{Irr}(X)$ is a singleton for every variable X. It is easy to qualify a grammar. For this, every variable X is replaced by variables X_u, one for each $u \in \mathrm{Irr}(X)$. In each production $Y \to m$, each variable X in the handle is replaced by all possible X_u. For each new handle m' obtained in this way, substitute u for X_u for all variables, and then compute the reduced word r of the resulting word. The word r is in $\mathrm{Irr}(Y)$. Add the production $Y_r \to m'$. When this is done for all possible choices, the resulting grammar is qualified.

We recall the following two lemmas from [1].

Lemma 9.10. *If $X \xrightarrow{+} gXd$ for some words in $g, d \in (A \cup \bar{A})^*$, then there exist words $x, y, p, q \in A^*$ such that*

$$\rho(g) = \bar{x}px, \quad \rho(d) = \bar{y}\bar{q}y$$

and moreover p and q are conjugate words.

A pair (g, d) such that $X \xrightarrow{+} gXd$ is a *lifting* pair if the word p in Lemma 9.10 is nonempty, it is a *flat* pair if $p = \varepsilon$.

Lemma 9.11. *The language L has bounded width iff G has no flat pair.*

We are now ready for the proof of Proposition 9.7. Consider a finite context-free grammar G, with axiom S, generating the well-formed language L with bounded width. Consider a word Dyck prime u that is a factor of a word in L. We define, for each word u, a set of tuples called *borders* of u. We shall see that if two Dyck primes u, u' have the same set of borders, then they are equivalent in the syntactic equivalence of L. The main argument to show that L is D-finite will be to prove that the set of all borders is finite. This relies on the fact that L has bounded width.

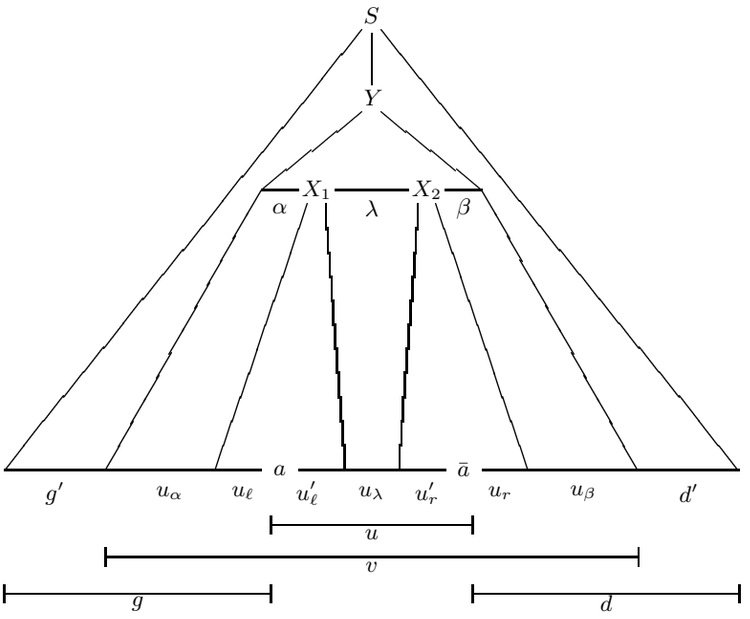

Fig. 1. The derivation tree.

Let (g, d) be any context for u. Consider a derivation $S \xrightarrow{*} gud$. In the derivation tree associated to this derivation (Figure 1), we consider the smallest sub-

tree that generates a word v that has as factor the Dyck prime u. Let Y be the root of this tree. Then $S \xrightarrow{*} g'Yd'$, $Y \xrightarrow{*} v$, and u is a factor of v. The minimality condition on the subtree implies that the derivation factorizes into $Y \longrightarrow \alpha X_1 \lambda X_2 \beta \xrightarrow{*} v$ where $\alpha \xrightarrow{*} u_\alpha$, $X_1 \xrightarrow{*} u_\ell a u'_\ell$, $\lambda \xrightarrow{*} u_\lambda$, $X_2 \xrightarrow{*} u'_r \bar{a} u_r$, $\beta \xrightarrow{*} u_\beta$ and

$$v = u_\alpha u_\ell a u'_\ell u_\lambda u'_r \bar{a} u_r u_\beta$$

with $v = u_\alpha u_\ell u u_r u_\beta$ and $u = a u'_\ell u_\lambda u'_r \bar{a}$. Observe that $g = g' u_\alpha u_\ell$ and $d = u_r u_\beta d'$. Notice that there might be the special case $X_1 = a$ and similarly $X_2 = \bar{a}$. Also, u_λ may be the empty word.

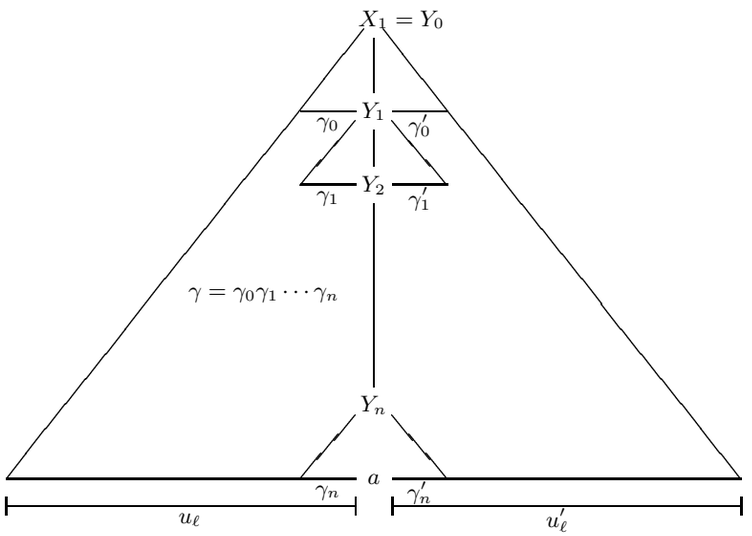

Fig. 2. The path from X_1 to the distinguished letter a.

Consider now the variables $Y_0 = X_1, Y_1, \ldots, Y_n$ on the path from X_1 to the initial letter a of u (Figure 2). Denote the productions used on this path $Y_i \to \gamma_i Y_{i+1} \gamma'_i$ for $i = 0, \ldots, n-1$, and $Y_n \to \gamma_n a \gamma'_n$. It follows that $\gamma_0\gamma_1 \cdots \gamma_n \xrightarrow{*} u_\ell$. Similarly, there are words $\delta_0, \ldots, \delta_m$ such that $\delta_m \cdots \delta_0 \xrightarrow{*} u_r$. A *border* of u is the tuple $(Y, \alpha, \gamma, \delta, \beta)$, with $\gamma = \gamma_0\gamma_1 \cdots \gamma_n$ and $\delta = \delta_m \cdots \delta_0$. If $(Y, \alpha, \gamma, \delta, \beta)$ is a border of u, then by construction, there are words g', d', u_α, u_ℓ, u_r, u_β with $S \xrightarrow{*} g'Yd'$, $\alpha \xrightarrow{*} u_\alpha$, $\gamma \xrightarrow{*} u_\ell$, $\delta \xrightarrow{*} u_r$, $\beta \xrightarrow{*} u_\beta$ such that $(g' u_\alpha u_\ell, u_r u_\beta d')$ is a context for u in L. It follows that if u' has the same borders that u has, then u' has the same contexts as u.

In order to complete the proof, we show that if L has bounded width, then the lengths of the components γ and δ in any border are uniformly bounded. This shows that the set of all borders of all Dyck primes is finite.

We carry out the proof for γ. As described above, $\gamma = \gamma_0\gamma_1\cdots\gamma_n$, where n is the length of the path from variable X_1 to the initial letter a. If this length is not bounded, then there is a variable, say X that appears arbitrarily often on this path. Consider all consecutive occurrences of this variable on the path. Assume there are $k+1$ of them. Each of the first k yields an iterative pair $X \xrightarrow{+} g_i X d_i$, and by Lemma 9.10, there exist words $x_i, y_i \in A^*$, $p_i, q_i \in A^+$ such that $\rho(g_i) = \bar{x}_i p_i x_i$, $\rho(d_i) = \bar{y}_i \bar{q}_i y_i$. Consider the derivation obtained by composing these iterating pairs:

$$X \xrightarrow{*} g_1 g_2 \cdots g_k X d_k d_{k-1} \cdots d_1, \quad X \xrightarrow{*} w$$

The resulting word $g_1 g_2 \cdots g_k w d_k d_{k-1} \cdots d_1$ is a factor of $u_\ell a u'_\ell$. Moreover, the occurrence of the letter a is an occurrence in the factor w, that is $w = w'aw''$, and the letter a cannot be reduced, in the Dyck reduction, with any letter in $w'' d_k d_{k-1} \cdots d_1$ since it reduces with the letter $\bar{a}$ in $u'_r \bar{a} u_r$. Hence this occurrence of a remains in $\rho(w)$. The word $g_1 g_2 \cdots g_k w d_k d_{k-1} \cdots d_1$ simplifies into $\bar{x}_1 p_1 x_1 \cdots \bar{x}_k p_k x_k \rho(w) \bar{y}_k \bar{q}_k y_k \cdots \bar{y}_1 \bar{q}_1 y_1$.

Observe that in the suffix $\bar{y}_k \bar{q}_k y_k \cdots \bar{y}_1 \bar{q}_1 y_1$, the number of barred letters exceeds by $|\bar{q}_k \cdots \bar{q}_1|$ the number of unbarred letters. All these letters must reduce to the empty word with letters in w''. Since $\rho(w)$ is fixed, this cannot happen. Thus k is uniformly bounded.

The set of all borders of all Dyck primes is finite. If $(Y, \alpha, \gamma, \delta, \beta)$ is a border, there are words g', d', u_α, u_ℓ, u_r, u_β with $S \xrightarrow{*} g'Yd'$, $\alpha \xrightarrow{*} u_\alpha$, $\gamma \xrightarrow{*} u_\ell$, $\delta \xrightarrow{*} u_r$, $\beta \xrightarrow{*} u_\beta$ and a word z such that $u_\ell z u_r$ is a Dyck prime. Wee have seen that the lengths of γ and δ are bounded. The existence of z is easy to check for a given pair (u_ℓ, u_r). Thus the construction is effective. □

Acknowledgment. We thank Isabelle Fagnot for helpful discussions.

References

1. J. Berstel and L. Boasson. XML-grammars. In *MFCS 2000 Mathematical Foundations of Computer Science* (M. Nielsen and B. Rovan, Eds.), Springer-Verlag, Lect. Notes Comput. Sci. **1893**, pages 182–191, 2000.
2. J.H. Conway. *Regular Algebra and Finite Machines.* Chapman and Hall, London, 1971.
3. N. Chomsky and M.P. Schützenberger. The Algebraic Theory of Context-Free Languages. In *Computer Programming and Formal Systems* (P. Braffort and D. Hirschberg, Eds.), North-Holland, Amsterdam, pages 118–161, 1963.
4. S. Ginsburg and M.A. Harrison. Bracketed Context-Free Languages. *J. Comput. Syst. Sci.*, 1:1–23, 1967.
5. Michael A. Harrison. *Introduction to Formal Language Theory.* Addison-Wesley, Reading, Mass., 1978.
6. D.E. Knuth. A Characterization of Parenthesis Languages. *Inform. Control*, 11:269–289, 1967.
7. A.J. Korenjak and J.E. Hopcroft. Simple Deterministic Grammars. In *7th Switching and Automata Theory*, pages 36–46, 1966.

8. R. McNaughton. Parenthesis Grammars. *J. Assoc. Mach. Comput.*, 14:490–500, 1967.
9. W3C Recommendation REC-xml-19980210. *Extensible Markup Language (XML) 1.0*, 10 February 1998. `http://www.w3.org/TR/REC-XML`.
10. W3C Working Draft. *XML Schema Part 0,1 and 2*, 22 September 2000. `http://www.w3.org/TR/xmlschema-0,1,2`.