Context-Free Languages of Countable Words*

Zoltán Ésik and Szabolcs Iván

Dept. of Computer Science, University of Szeged, Hungary

Abstract. We define context-free grammars with Büchi acceptance condition generating languages of countable words. We establish several closure properties and decidability results for the class of Büchi context-free languages generated by these grammars. We also define context-free grammars with Müller acceptance condition and show that there is a language generated by a grammar with Müller acceptance condition which is not a Büchi context-free language.

1 Introduction

A word over an alphabet Σ is an isomorphism type of a labeled linear order. In this paper, in addition to finite words and ω-words, we also consider words whose underlying linear order is any countable linear order, including scattered and dense linear orders, cf. [21].

Finite automata on ω-words were introduced by Büchi [9]. He used automata to prove the decidability of the monadic second-order theory of the ordinal ω. Automata on ω-words have since been extended to automata on ordinal words beyond ω, cf. [10,11,1,25,26], to words whose underlying linear order is not necessarily well-ordered, cf. [3,8], and to automata on finite and infinite trees, cf. [14,22,20]. Many decidability results have been obtained using the automata theoretic approach, both for ordinals and other linear orders, and for first-order and monadic second-order theories in general.

Countable words were first investigated in [13], where they were called "arrangements". It was shown that any arrangement can be represented as the frontier word (i.e., the sequence of leaf labels) of a possibly infinite labeled binary tree. Moreover, it was shown that words definable by finite recursion schemes are exactly those words represented by the frontiers of regular trees. These words were called regular in [6]. Courcelle [13] raised several problems that were later solved in the papers [17,23,5]. In [23], it was shown that it is decidable for two regular trees whether they represent the same regular word. In [17], an infinite collection of regular operations has been introduced and it has been shown that each regular word can be represented by a regular expression. Complete axiomatizations have been obtained in [4] and [5] for the subcollections of the regular operations that allow for the representation of the regular ordinal words and the regular scattered words, respectively. Complete axiomatization of the full

* Research supported by grant no. K 75249 from the National Foundation of Hungary for Scientific Research.

M. Leucker and C. Morgan (Eds.): ICTAC 2009, LNCS 5684, pp. 185–199, 2009.

collection of the regular operations has been obtained in [6], where it is also proved that there is a polynomial time algorithm to decide whether two regular expressions represent the same regular word. In [8,3], the authors proposed regular expressions to represent languages (i.e., sets) of scattered countable words and languages of possibly dense words with no upper bound on the size of the words. They have established Kleene theorems stating that a language of infinite words is recognizable by a finite automaton iff it can be represented by a regular expression.

In addition to automata and expressions (or terms), a third common way of representing languages of finite words is by generative grammars. Context-free grammars have been used to generate languages of ω-words in [12] and in [18]. However, we are not aware of any work on context-free grammars as a device generating languages of countable words possibly longer than ω, except for the recent [15] that deals only with linear grammars. In this paper we consider languages of countable words generated by context-free grammars equipped with a Büchi-type acceptance condition, called BCFG's. A BCFG is a system $G = (N, \Sigma, P, S, F)$, where (N, Σ, P, S) is an ordinary context-free grammar and $F \subseteq N$ is the set of repeated (or final) nonterminals. A derivation tree t of a grammar G is a possibly infinite tree whose vertices are labeled in the set $N \cup \Sigma \cup \{\epsilon\}$, so that each vertex is labeled by a nonterminal in N, a letter in the terminal alphabet Σ, or by the empty word ϵ. The labeling is locally consistent with the rules contained in P in the usual way. Moreover, it is required that each derivation tree satisfies the "Büchi condition F", i.e., on each infinite path of t at least one repeated nonterminal has to occur infinitely many times. The frontier of a derivation tree t determines a countable word w over the alphabet $N \cup \Sigma$. When w is a word over the terminal alphabet Σ and the root of t is labeled by the start symbol S, we say that w is contained in the Büchi context-free language generated by G. The language class BCFL consists of all such Büchi context-free languages.

It is well-known (see e.g., [16]) that ordinary context-free languages of finite words are precisely the frontier languages of sets of finite trees recognizable by finite tree automata. Tree automata over infinite trees have been introduced in [20]. Just as automata over ω-words, a tree automaton may be equipped with different acceptance conditions such as the Büchi and Müller acceptance conditions, or the Rabin, Streett and parity conditions, cf. [19,24]. In the setting of ω-words, these conditions are equally powerful (at least for nondeterministic automata). Nevertheless, some yield more succinct representation than others, or have different algorithmic properties. On the other hand, in the setting of infinite trees, the Büchi acceptance condition is strictly less powerful than the Müller acceptance condition which is equivalent to the Rabin, Streett, and parity conditions, cf. [19,24]. While in the present paper we are mainly concerned with the Büchi condition for generating context-free languages of countable words, we still show that the Müller condition is strictly more powerful also in the setting of countable words. This result is not immediate from the tree case.

2 Linear Orders and Words

In this section we recall some concepts for linear orders and words. A good reference on linear orders is [21].

A partial order, or partial ordering is a set P equipped with a (partial) order relation usually denoted $\leq$. We sometimes write $x < y$ if $x \leq y$ and $x \neq y$. A linear order is a partial order $(P, \leq)$ whose order relation is total, so that $x \leq y$ or $y \leq x$ for all $x, y \in P$. A countable (finite or infinite, respectively) linear order is a linear order which is a countable (finite or infinite, respectively) set. When $(P, \leq)$ and $(Q, \leq)$ are linear orders, an isomorphism (embedding, respectively) $(P, \leq) \to (Q, \leq)$ is a bijection (injection, respectively) $h : P \to Q$ such that $x \leq y$ implies $h(x) \leq h(y)$ for all $x, y \in P$. When two linear orders are isomorphic, we also say that they have the same order type (or isomorphism type).

Below when there is no danger of confusion, we will denote a linear order just by $P, Q, \ldots$. Suppose that P is a linear order. Then any subset X of P determines a sub-order of P whose order relation is the restriction of the order relation of P to X. Note that the inclusion function $X \hookrightarrow P$ is an embedding of X into P. When in addition X is such that for all $x, y \in X$ and $z \in P$, $x < z < y$ implies that $z \in X$, then we call X an interval. In particular, for any $x, y \in P$, the set $[x, y] = \{z : x \leq z \leq y\}$ is an interval.

We recall that a linear order $(P, \leq)$ is a *well-order* if each nonempty subset of P has a least element, and is *dense* if it has at least two elements and for any $x < y$ in P there is some z with $x < z < y$.[1] A *quasi-dense* linear order is a linear order $(P, \leq)$ containing a dense linear sub-order, so that P has a subset P' such that $(P', \leq)$ is a dense order. Finally, a *scattered* linear order is a linear order which is not quasi-dense.

It is clear that every finite linear order is a well-order, every well-order is a scattered order, and every dense order is quasi-dense. It is well-known that up to isomorphism there are 4 countable dense linear orders, the rationals $\mathbb{Q}$ with the usual order, $\mathbb{Q}$ endowed with a least or a greatest element, and $\mathbb{Q}$ endowed with both a least and a greatest element.

An ordinal is an order type of a well-order. The finite ordinals n are the isomorphism types of the finite linear orders. As usual, we denote by ω the least infinite ordinal, which is the order type of the finite ordinals, and of the positive integers $\mathbb{N}$ equipped with the usual order. The order type of $\mathbb{Q}$ is denoted η.

When τ and τ' are order types, we say that $\tau \leq \tau'$ if there is an embedding of a linear order of type τ into a linear order of type τ'. The relation $\leq$ defined above is a linear order of the ordinals.

We define several operations on linear orders. First, the reverse $(P, \leq')$ of a linear order $(P, \leq)$ is defined by $x \leq' y$ iff $y \leq x$, for all $x, y \in P$. We will sometimes denote the reverse order $(P, \leq')$ by P^r. It is clear that the reverse of a scattered (dense, respectively) linear order is scattered (dense, respectively).

Suppose that P and Q are linear orders. Then the sum $P + Q$ is the linear order on the disjoint union of P and Q such that P and Q are intervals of $P + Q$

[1] In [21], a singleton linear order is also called dense.

and $x \leq y$ holds for all $x \in P$ and $y \in Q$. There is a more general notion. Suppose that I is a linear order and for each $i \in I$, P_i is a linear order. Then the generalized sum $P = \sum_{i \in I} P_i$ is obtained by replacing each point i of I with the linear order P_i. Formally, the generalized sum P is the linear order on the disjoint union $\bigcup_{i \in I} P_i$ equipped with the order relation such that each P_i is an interval and for all $i, j \in I$ with $i < j$, if $x \in P_i$ and $y \in P_j$ then $x < y$. The generalized sum gives rise to a product operation. Let P and Q be linear orders, and for each $y \in Q$, let P_y be an isomorphic copy of P. Then $P \times Q$ is defined as the linear order $\sum_{y \in Q} P_y$. Note that this linear order is isomorphic to the linear order on the cartesian product of P and Q equipped with the order relation $(x, y) \leq (x', y')$ iff $y < y'$ or ($y = y'$ and $x \leq x'$).

Lemma 1. *[21] Any scattered generalized sum of scattered linear orders is scattered. Similarly, any well-ordered generalized sum of well-orders is a well-order. Every quasi-dense linear order is a dense generalized sum of (nonempty) scattered linear orders.*

Thus, when I is a scattered linear order and for each $i \in I$, P_i is a scattered linear order, then so is $\sum_{i \in I} P_i$, and similarly for well-orders. And if P is a quasi-dense linear order, then there is a dense linear order D and (nonempty) scattered linear orders P_x, $x \in D$ such that P is isomorphic to $\sum_{x \in D} P_x$.

The above operations preserve isomorphism, so that they give rise to corresponding operations $\tau + \tau'$ and $\tau \times \tau'$ on order types. In particular, the sum and product of two ordinals is well-defined (and is an ordinal). The reverse of an order type τ will be denoted $-\tau$. The ordinals are also equipped with the exponentiation operation, cf. [21].

An alphabet Σ is a finite nonempty set. A word over an alphabet Σ is a labeled linear order, i.e., a system $u = (P, \leq, \lambda)$, where $(P, \leq)$ is a linear order, sometimes denoted $\mathrm{dom}(u)$, and λ is a labeling function $P \to \Sigma$. The underlying linear order $\mathrm{dom}(\epsilon)$ of the empty word ϵ is the empty linear order. We say that a word is finite (infinite or countable, respectively), if its underlying linear order is finite (infinite or countable, respectively). An isomorphism of words is an isomorphism of the underlying linear orders that preserves the labeling. Embeddings of words are defined in the same way. We usually identify isomorphic words. We will say that a word u is a subword of a word v if there is an embedding $u \hookrightarrow v$. When in addition the image of the underlying linear order of u is an interval of the underlying linear order of v we call u a factor of v.

The order type of a word is the order type of its underlying linear order. Thus, the order type of a finite word is a finite linear order. A word whose order type is ω is called an ω-word.

Let $\Sigma = \{a, b\}$. Some examples of words over Σ are the finite word aab which is the (isomorphism class of the) 3-element labeled linear order $0 < 1 < 2$ whose points are labeled a, a and b, in this order, and the infinite words a^ω and $a^{-\omega}$, whose order types are ω and $-\omega$, respectively, with each point labeled a. For another example, consider the linear order $\mathbb{Q}$ of the rationals and label each point a. The resulting word of order type η is denoted a^η. More generally, let Σ be the alphabet $\{a_1, \ldots, a_n\}$ of size n. Then up to isomorphism there is a unique labeling of the rationals such that between any two points there are n points labeled $a_1, \ldots, a_n$, respectively. This word is denoted $(a_1, \ldots, a_n)^\eta$, cf. [17].

The reverse of a word $u = (P, \leq, \lambda)$ is $u^r = (P, \leq', \lambda)$, where $(P, \leq')$ is the reverse of $(P, \leq)$. Suppose that $u = (P, \leq, \lambda)$ and $v = (Q, \leq, \lambda')$ are words over Σ. Then their concatenation (or product) uv is the word over Σ whose underlying linear order is $P + Q$ and whose labeling function agrees with λ on points in P, and with λ' on points in Q. More generally, when I is a linear order and u_i is a word over Σ with underlying linear order $P_i = \mathrm{dom}(u_i)$, for each $i \in I$, then the generalized concatenation $\prod_{i \in I} u_i$ is the word whose underlying linear order is $\sum_{i \in I} P_i$ and whose labeling function agrees with the labeling function of P_i on the elements of each P_i. In particular, when $u_0, u_1, \ldots, u_n, \ldots$ are words over Σ, and I is the linear order ω or its reverse, then $\prod_{i \in I} u_i$ is the word $u_0 u_1 \ldots u_n \ldots$ or $\ldots u_n \ldots u_1 u_0$, respectively. When $u_i = u$ for each i, these words are denoted u^ω and $u^{-\omega}$, respectively.

In the sequel, we will make use of the substitution operation on words. Suppose that u is a word over Σ and for each letter $a \in \Sigma$, u_a is a word over Δ. Then the word $u[a \leftarrow u_a]_{a \in \Sigma}$ obtained by substituting u_a for each occurrence of a letter a in u (or replacing each occurrence of a letter a with u_a) is formally defined as follows. Let $u = (P, \leq, \lambda)$ and $u_a = (P_a, \leq_a, \lambda_a)$ for each $a \in \Sigma$. Then for each $i \in P$ let $u_i = (P_i, \leq_i, \lambda_i)$ be an isomorphic copy of $P_{\lambda(i)}$. We define

$$u[a \leftarrow u_a]_{a \in \Sigma} = \prod_{i \in P} u_i.$$

Note that when $u = a^\omega$, then $u[a \leftarrow v]$ is v^ω, and similarly for $v^{-\omega}$. For any words $u_1, \ldots, u_n$ over an alphabet Σ, we define

$$(u_1, \ldots, u_n)^\eta = (a_1, \ldots, a_n)^\eta [a_1 \leftarrow u_1, \ldots, a_n \leftarrow u_n].$$

We call a word over an alphabet Σ well-ordered, scattered, dense, or quasi-dense if its underlying linear order has the appropriate property. For example, the words a^ω, $a^\omega b^\omega a$, $(a^\omega)^\omega$ over the alphabet $\{a, b\}$ are well-ordered, the words $a^\omega a^{-\omega}$, $a^{-\omega} a^\omega$ are scattered, the words a^η, $a^\eta b a^\eta$, $(a, b)^\eta$ are dense, and the words $(ab)^\eta$, $(a^\omega)^\eta$, $(a^\eta b)^\omega$ are quasi-dense. From Lemma 1 we immediately have:

Lemma 2. *Any scattered generalized product of scattered words is scattered. Any well-ordered generalized product of well-ordered words is well-ordered. Moreover, every quasi-dense word is a dense product of (nonempty) scattered words.*

As already mentioned, we will usually identify isomorphic words, so that a word is an isomorphism type (or isomorphism class) of a labeled linear order. When Σ is an alphabet, we let Σ^*, Σ^ω and Σ^∞ respectively denote the set of all finite words, ω-words, and countable words over Σ. Σ^+ is the set of all finite nonempty words. The length of a finite word w will be denoted $|w|$.

A language over Σ is any subset L of Σ^∞. When $L \subseteq \Sigma^*$ or $L \subseteq \Sigma^\omega$, we sometimes call L a language of finite words or ω-words, or an ω-language.

Languages are equipped with several operations, including the usual set theoretic operations. We now define the generic operation of language substitution.

Suppose that $u \in \Sigma^\infty$ and for each $a \in \Sigma$, $L_a \subseteq \Delta^\infty$. Then the words in the language $u[a \leftarrow L_a]_{a\in\Sigma} \subseteq \Delta^\infty$ are obtained from u by substituting in all possible ways a word in L_a for each occurrence of each letter $a \in \Sigma$. Different occurrences of the same letter a may be replaced by different words in L_a.

Formally, suppose that $u = (P, \leq, \lambda)$. For each $x \in P$ with $\lambda(x) = a$, let us choose a word $u_x = (P_x, \leq_x, \lambda_x)$ which is isomorphic to some word in L_a. Then the language $u[a \leftarrow L_a]_{a\in\Sigma}$ consists of all words $\prod_{x\in P} u_x$.

Suppose now that $L \subseteq \Sigma^\infty$ and for each $a \in \Sigma$, $L_a \subseteq \Delta^\infty$. Then

$$L[a \leftarrow L_a]_{a\in\Sigma} = \bigcup_{u\in L} u[a \leftarrow L_a]_{a\in\Sigma}.$$

We call $L[a \leftarrow L_a]_{a\in\Sigma}$ the language obtained from L by substituting the language L_a for each $a \in \Sigma$.

As mentioned above, set theoretic operations on languages in Σ^∞ have their standard meaning. Below we define some other operations.

Let $L, L_1, L_2, \ldots, L_m \subseteq \Sigma^\infty$. Then we define:

1. $L_1L_2 = ab[a \leftarrow L_1, b \leftarrow L_2] = \{uv : u \in L_1, v \in L_2\}$.
2. $L^* = \{a\}^*[a \leftarrow L] = \{u_1 \ldots u_n : n < \omega,\ u_i \in L\}$.
3. $L^\omega = \{a^\omega\}[a \leftarrow L] = \{u_0u_1 \ldots u_n \ldots : u_i \in L\}$.
4. $L^{-\omega} = \{a^{-\omega}\}[a \leftarrow L] = \{\ldots u_n \ldots u_1u_0 : u_i \in L\}$.
5. $(L_1, \ldots, L_m)^\eta = \eta(a_1, \ldots, a_m)[a_1 \leftarrow L_1, \ldots, a_m \leftarrow L_m]$.
6. $L^\infty = \{a\}^\infty[a \leftarrow L]$.

The above operations are respectively called concatenation, star, $^\omega$-power, $^{-\omega}$-power, $^\eta$-power, and $^\infty$-power.

Some more operations. The reverse L^r of a language $L \subseteq \Sigma^\infty$ is defined as $L^r = \{u^r : u \in L\}$. The prefix language $\mathrm{Pre}(L)$ is given by $\mathrm{Pre}(L) = \{u : \exists v\ uv \in L\}$ and the suffix language $\mathrm{Suf}(L)$ is defined symmetrically. The infix (or factor) language $\mathrm{In}(L)$ is $\{u : \exists v, w\ vuw \in L\}$, and the language $\mathrm{Sub}(L)$ of subwords of L is the collection of all words u such that there is an embedding $u \hookrightarrow v$ for some $v \in L$.

3 Büchi Context-Free Languages

Recall that an ordinary context-free grammar (CFG) is a system $G=(N, \Sigma, P, S)$ where N and Σ are the disjoint alphabets of nonterminals and terminal symbols (or letters), P is a finite set of productions of the form $A \to p$ where $A \in N$ and $p \in (N \cup \Sigma)^*$, and $S \in N$ is the start symbol. Each context-free grammar $G = (N, \Sigma, P, S)$ generates a context-free language $L(G) \subseteq \Sigma^*$ which can be defined either by using the derivation relation $\Rightarrow^*$ or by using the concept of derivation trees.

We recall that for finite words $p, q \in (N \cup \Sigma)^*$ it holds that $p \Rightarrow q$ if p and q can be written as $p = p_1Ap_2$, $q = p_1rp_2$ such that $A \to r$ is in P. The relations $\Rightarrow^+$ and $\Rightarrow^*$ are respectively the transitive closure and the reflexive-transitive closure of the direct derivation relation $\Rightarrow$. The context-free language generated by G is $L(G) = \{u \in \Sigma^* : S \Rightarrow^* u\}$. Two context-free grammars G and G' having the same terminal alphabet are called equivalent if $L(G) = L(G')$. We let CFL denote the class of all context-free languages.

A derivation tree is a partial mapping $t : \mathbb{N}^* \to N \cup \Sigma \cup \{\epsilon\}$ whose domain $\mathrm{dom}(t)$ is finite, nonempty and prefix closed (i.e., $uv \in \mathrm{dom}(t) \Rightarrow u \in \mathrm{dom}(t)$). The elements of $\mathrm{dom}(t)$ are the vertices of t, and for any vertex v, $t(v)$ is the label of v. The empty word ϵ is the root of t, and $t(\epsilon)$ is the root symbol. The vertices in $\mathrm{dom}(t)$ are equipped with both the lexicographic order and the prefix order. Let $x, y \in \mathrm{dom}(t)$. We say that $x \leq y$ in the prefix order if $y = xz$ for some $z \in \mathbb{N}^*$. Moreover, we say that $x < y$ in the lexicographic order if $x = uiz$ and $y = ujz'$ for some $u, z, z' \in \mathbb{N}^*$ and $i, j \in \mathbb{N}$ with $i < j$. The leaves of t are the maximal elements of $\mathrm{dom}(t)$ with respect to the prefix order. When $x, y \in \mathrm{dom}(t)$ and $y = xi$ for some $i \in \mathbb{N}$, then we say that y is the ith successor of x and x is the predecessor of y. The function t is required to satisfy the local consistency condition that whenever $t(u) = A$ with $A \in N$ and u is not a leaf, then either $A \to \epsilon \in P$ and $t(u1) = \epsilon$ and $t(ui)$ is not defined for any $i \in \mathbb{N}$ with $i > 1$, or there is a production $A \to p$ such that $|p| = n$ with $n > 0$ and $t(ui)$ is defined for some $i \in \mathbb{N}$ iff $i \leq n$, moreover, $t(ui)$ is the ith letter of p for each $i \leq n$. The frontier of t is the linearly ordered set of leaves whose order is the lexicographic order. The frontier determines a word in $(N \cup \Sigma)^*$ whose underlying linear order is obtained from the frontier of t by removing all those vertices whose label is ϵ. The labeling function is the restriction of the function t to the remaining vertices. This word is sometimes called the frontier word of t. It is well-known that a word u in Σ^* belongs to $L(G)$ iff there is a derivation tree whose root is labeled S and whose frontier word is u.

We now define context-free grammars generating countable words.

Definition 1. *A* context-free grammar with Büchi acceptance condition, *or BCFG is a system $G = (N, \Sigma, P, S, F)$ where N, Σ, P, S are the same as above, and $F \subseteq N$ is the set of* repeated nonterminals*.*

Note that each BCFG has an underlying CFG. Suppose $G = (N, \Sigma, P, S, F)$ is a BCFG. A derivation tree t is defined as above except that $\mathrm{dom}(t)$ may now be infinite. However, we require that at least one repeated nonterminal occurs infinitely often along each infinite path. When the root symbol of t is A and the frontier word of t is p, we also write $A \Rightarrow^\infty p$. (Here, it is allowed that A is a terminal in which case $A = p$.) The language (of countable words) generated by G is $L^\infty(G) = \{u \in \Sigma^\infty : S \Rightarrow^\infty u\}$. When G and G' are BCFG's with the same terminal alphabet Σ generating the same language, then we say that G and G' are equivalent.

Definition 2. *We call a set $L \subseteq \Sigma^\infty$ a Büchi context-free language, or a BCFL, if it can be generated by some BCFG, i.e., when $L = L^\infty(G)$ for some BCFG $G = (N, \Sigma, P, S, F)$.*

Suppose that $G = (N, \Sigma, P, S, F)$ is a BCFG with underlying CFG $G' = (N, \Sigma, P, S)$. Then we define $L^*(G)$ as the CFL $L(G')$. Note that in general it does not hold that $L^*(G) = L^\infty(G) \cap \Sigma^*$. Later we will see that for every BCFG $G = (N, \Sigma, P, S, F)$ it holds that $L^\infty(G) \cap \Sigma^*$ is a CFL. It is clear that CFL $\subseteq$ BCFL, for if $G = (N, \Sigma, P, S, F)$ is a BCFG with $F = \emptyset$, then $L^\infty(G) = L^*(G)$.

Example 1. Consider the sequence $(w_n)_{n<\omega}$ of words over $\{a\}$ defined inductively by $w_0 = a$, and for each $n < \omega$, $w_{n+1} = w_n^\omega$. Note that the order type of w_n is ω^n. For each n, the BCFG $G_n = (N, \{a\}, P, S_n, N)$ with

$$N = \{S_0, \ldots, S_n\} \quad \text{and} \quad P = \{S_0 \to a\} \cup \{S_i \to S_{i-1}S_i : 1 \leq i \leq n\}$$

generates the singleton language $\{w_n\}$, cf. [7]. Using this, it follows that the BCFG $G'_n = (N \cup \{S\}, \{a\}, P \cup \{S \to S_i : 0 \leq i \leq n\}, S, N)$ generates the set $\{w_i : 0 \leq i \leq n\}$.

Example 2. Let Σ be an alphabet and let $a_1, \ldots, a_n \in \Sigma$ be letters in Σ. The singleton language containing the word $(a_1, \ldots, a_n)^\eta$ is a BCFL generated by $G = (\{S\}, \Sigma, \{S \to Sa_1Sa_2 \ldots Sa_nS\}, S, \{S\})$.

Example 3. Consider the language L over the 1-letter alphabet $\{a\}$ consisting of all words in $\{a\}^\infty$ whose domain is well-ordered of order type $< \omega^n$. Then L is generated by the BCFG $G = (N, \{a\}, P, S_n, N - \{S_n\})$ with $N = \{S_n, \ldots, S_0\}$ and $P = \{S_i \to \epsilon : 0 \leq i \leq n\} \cup \{S_0 \to a\} \cup \{S_i \to S_{i-1}S_i : 1 \leq i \leq n\}$.

Let L' be the subset of L consisting of those words whose domain is a limit ordinal. Then L' is the set of all finite concatenations of the words w_i, $1 \leq i < n$ of Example 1. L' is generated by the BCFG $G = (N, \{a\}, P, S, N - \{S\})$ with $N = \{S, S_0, \ldots, S_{n-1}\}$ and

$$P = \{S \to S_iS : 1 \leq i < n\} \cup \{S \to \epsilon\} \cup \{S_0 \to a\} \cup \{S_i \to S_{i-1}S_i : 1 \leq i < n\}.$$

Example 4. The language $\{a^\omega b^{-\omega}\}^* \cup \{a^\omega b^{-\omega}\}^\omega$ is a BCFL generated by $G = (N, \{a, b\}, P, S, N)$ with $N = \{S, X\}$ and $P = \{S \to XS, S \to \epsilon, X \to aXb\}$.

Example 5. Using the fact (see e.g., Theorem 2.5 in [21]) that any countable linear order can be embedded into $\mathbb{Q}$, we get that Σ^∞ is a BCFL for any alphabet Σ, generated by the BCFG $G = (\{S\}, \Sigma, \{S \to \epsilon, S \to SS\} \cup \{S \to SaS : a \in \Sigma\}, S, \{S\})$.

4 Normal Forms

The results of this section show that each BCFG can be transformed in polynomial time into an equivalent BCFG which is "weakly ϵ-free" and does not contain useless nonterminals nor any chain productions. Moreover, each BCFG can be transformed into an equivalent "ϵ-free" BCFG having no useless nonterminals.

Definition 3. *Let $G = (N, \Sigma, P, S, F)$ be a BCFG. We say that a nonterminal A is useful if there exist words $p, q \in (N \cup \Sigma)^*$ and $u \in \Sigma^\infty$ such that $S \Rightarrow^* pAq$ and $A \Rightarrow^\infty u$. We say that G contains no useless nonterminals if either $N = \{S\}$, $P = \emptyset$ and $F = \emptyset$, or each nonterminal is useful.*

Note that when $G = (N, \Sigma, P, S, F)$ contains no useless nonterminals, then $L^\infty(G)$ is empty iff $N = \{S\}$, $P = \emptyset$ and $F = \emptyset$. Moreover, if $L^\infty(G)$ is not empty, then for each $A \in N$ there are words $u, v \in \Sigma^\infty$ with $S \Rightarrow^\infty uAv$.

Definition 4. *Let $G = (N, \Sigma, P, S, F)$ be a BCFG. We call G weakly ϵ-free if either $L^\infty(G) = \emptyset$, or for each nonterminal A there is a nonempty word $u \in \Sigma^\infty$ with $A \Rightarrow^\infty u$, or $S \to \epsilon$ is the only production.*

As usual, a chain production is of the form $A \to B$, where A, B are nonterminals.

Proposition 1. *For each BCFG G one can construct in polynomial time an equivalent weakly ϵ-free BCFG G' without any chain productions which contains no useless nonterminals.*

Definition 5. *We say that the BCFG $G = (N, \Sigma, P, S, F)$ is ϵ-free if the following conditions hold: 1. G is weakly ϵ-free. 2. Except possibly for the production $S \to \epsilon$, the right side of any other production is a nonempty word. Moreover, if $S \to \epsilon$ is a production, then S does not occur on the right side of any other production. 3. For each derivation tree t whose frontier determines a nonempty word in Σ^∞ there is a derivation tree t' with the same root symbol and frontier word which is well-founded in the following strict sense: For each vertex $x \in \mathrm{dom}(t')$, the subtree $t'|_x$ of t' rooted at x has at least one leaf labeled in Σ.*

Proposition 2. *For each BCFG G one can construct in polynomial time an equivalent ϵ-free grammar without useless nonterminals.*

Proposition 3. *Suppose that $G = (N, \Sigma, P, S, F)$ is an ϵ-free BCFG. Then $L^\infty(G) \cap \Sigma^* = L^*(G)$.*

Corollary 1. *A language $L \subseteq \Sigma^*$ is in BCFL iff L is in CFL.*

Remark 1. Suppose that $G = (N, \Sigma, P, S, F)$ is a BCFG with $F = N$. By an argument similar to the proof of the well-known pumping lemma for ordinary context-free languages we show that if $L^\infty(G) \cap \Sigma^*$ is infinite, then $L^\infty(G)$ contains an infinite word. Indeed, without loss of generality we may assume that G is ϵ-free without chain productions and useless nonterminals. Since $L^\infty(G) \cap \Sigma^*$ is infinite, there is a word $w \in L^\infty(G) \cap \Sigma^+$ with a finite strictly well-founded derivation tree rooted S such that at least one nonterminal is repeated along some path. This implies that w can be written as $xyuvz$ such that $yv \neq \epsilon$ and for some nonterminal A we have $S \Rightarrow^* xAz$, $A \Rightarrow^* yAv$ and $A \Rightarrow^* u$. Since $F = N$ we have $A \in F$. Thus, $S \Rightarrow^\infty xy^\omega v^{-\omega} z$, showing that $L^\infty(G)$ contains the infinite word $xy^\omega v^{-\omega} z$.

5 Closure Properties

In this section we establish the fact that BCFL's are effectively closed under substitution and use this result to derive the closure of BCFL's under the operations of union, concatenation, $^\omega$-power, $^{-\omega}$-power, $^\eta$-power and $^\infty$-power. Recall the definition of language substitution from Section 2.

Theorem 1. *If the languages L, L_a, $a \in \Sigma$ are BCFL's then so is $L' = L[a \leftarrow L_a]_{a\in\Sigma}$. Moreover, given BCFG's generating the languages L, L_a, $a \in \Sigma$, one can effectively construct a BCFG generating L'.*

Corollary 2. *The class BCFL is effectively closed under binary set union, concatenation, $^\omega$-power, $^{-\omega}$-power, $^\eta$-power and $^\infty$-power.*

Thus, for example, given a BCFG generating L, one can effectively construct a BCFG generating L^η. Moreover, for any ordinary context-free language $L \subseteq \Sigma^*$, L^ω, $L^{-\omega}$, L^η, L^∞ are BCFL's. We mention the following results.

Proposition 4. *If L is a Büchi context-free language, then L^r, $\mathrm{Pre}(L)$, $\mathrm{Suf}(L)$, $\mathrm{In}(L)$ and $\mathrm{Sub}(L)$ are all effectively Büchi context-free languages.*

Proposition 5. *For every alphabet Σ, the set of all dense words in Σ^∞ and the set of all quasi-dense words in Σ^∞ are BCFL's.*

Remark 2. Since a language of finite words $L \subseteq \Sigma^*$ is a BCFL iff it is a CFL, and since CFL's are not closed under intersection, it follows that BCFL's are not closed under complementation and intersection either.

6 Some Decidable Properties

In this section we show that it is decidable in polynomial time for a Büchi context free language given by a BCFG whether it is empty, consists of finite words, consists of infinite words, consists of ω-words, consists of well-ordered words, consists of scattered words, or it consists of dense words. We also establish a limitedness property of BCFL's.

Let $G = (N, \Sigma, P, S, F)$ be a BCFG. We define a directed graph Γ_G whose set of vertices is N. There is an edge $A \to B$ exactly when B occurs on the right side of a production whose left side is A. We partition N into strongly connected components. As usual, the strongly connected components can be partially ordered by $\mathcal{S} \leq \mathcal{S}'$ iff there is a sequence of nonterminals $A_0, \ldots, A_m$ such that $A_0 \in \mathcal{S}'$, $A_m \in \mathcal{S}$ and for each $i < m$ there is an edge from A_i to A_{i+1}.

The first fact is clear, since for every BCFG one can construct in polynomial time an equivalent BCFG without useless nonterminals.

Theorem 2. *It is decidable in polynomial time whether a BCFG generates an empty language.*

Theorem 3. *Let $G = (N, \Sigma, P, S, F)$ be a weakly ϵ-free BCFG having no useless nonterminal. Then $L^\infty(G)$ contains an infinite word iff there is a strongly connected component $\mathcal{S}$ of Γ_G which contains a nonterminal in F, and there is a production $A \to p$ with $A \in \mathcal{S}$ such that $|p| \geq 2$ and at least one nonterminal in $\mathcal{S}$ occurs in p.*

Corollary 3. *It is decidable in polynomial time whether the language $L^\infty(G)$ generated by a given BCFG G consists of finite words.*

Theorem 4. *It is decidable in polynomial time whether the language $L^\infty(G)$ generated by a given BCFG $G = (N, \Sigma, P, S, F)$ contains only infinite words.*

Below, we will make use of the notion of the *rank* of a scattered countable word. Let Σ be an alphabet. We define the sequence $(V_\alpha^\Sigma)_\alpha$ of subsets of Σ^∞, where α ranges over all countable ordinals. Let $V_0^\Sigma = \Sigma^*$. Then for any countable ordinal $\alpha > 0$, let V_α^Σ be the least set of words closed under finite concatenation which contains $\bigcup_{\beta<\alpha} V_\beta^\Sigma$ together with all words of the form $u_0u_1 \dots u_i \dots$ and $\dots u_i \dots u_1u_0$, where each u_i, $i < \omega$ is in $V_{\beta_i}^\Sigma$ for some β_i with $\beta_i < \alpha$. The following fact is immediate from Hausdorff's theorem [21].

Proposition 6. *A word in Σ^∞ is scattered iff it belongs to V_α^Σ for some countable ordinal α.*

Definition 6. *The* rank *of a scattered word w in Σ^∞ is the least ordinal α such that w is in V_α^Σ. If this ordinal is finite we say that w is of finite rank.*

Example 6. Consider the following languages over the singleton alphabet. Let $L_0 = \{a\}$ and $L_{n+1} = \{w^\omega, w^{-\omega} : w \in L_n\}$, for all $n < \omega$. Then for each n and for each word $w \in L_n$, we have that w is scattered of rank n. In particular, let $w_0 = a$ and $w_{n+1} = w_n^\omega$, for all $n < \omega$. Then each w_n is scattered of rank n.

Example 7. For any alphabet Σ and $n < \omega$, the set L_n of all scattered words in Σ^∞ of rank at most n is a BCFL: $L_0 = \Sigma^*$ and $L_{n+1} = (L_n^\omega \cup L_n^{-\omega})^*$.

Theorem 5. *Let $G = (N, \Sigma, P, S, F)$ be a weakly ϵ-free BCFG with no useless nonterminals. Then $L^\infty(G)$ consists of scattered words iff for each strongly connected component $\mathcal{S}$ of Γ_G with $\mathcal{S} \cap F \neq \emptyset$ and for each production $A \to p$ with $A \in \mathcal{S}$, the word p contains at most one occurrence of a nonterminal in $\mathcal{S}$.*

Corollary 4. *It is decidable in polynomial time whether the language $L^\infty(G)$ generated by a given BCFG G contains only scattered words.*

Corollary 5. *Suppose that $G = (N, \Sigma, P, S, F)$ is a BCFG such that $L^\infty(G)$ contains only scattered words. Then the rank of each word in $L^\infty(G)$ is at most the number of nonterminals in N.*

Corollary 6. *Let $w_0 = a$ and $w_{n+1} = (w_n)^\omega$ for all $n < \omega$. There exists no BCFL consisting only of scattered words containing all words w_n, for all $n < \omega$. In particular, for any alphabet Σ, the set of all scattered words in Σ^∞ is not a BCFL. Similarly, the set of all well-ordered words in Σ^∞ is not a BCFL.*

The language of all quasi-dense words in Σ^∞ is a BCFL, while its complement, the language of all scattered words in Σ^∞ is not. Thus we have:

Corollary 7. *For every alphabet Σ, including the singleton alphabet, the set of all BCFL's in Σ^∞ is not closed under complementation.*

Definition 7. *Suppose that $L \subseteq \Sigma^\infty$ is a language consisting of scattered words of finite rank bounded by some $n < \omega$. Then we define the rank of L as the maximum rank of a word in L.*

Theorem 6. *There is a polynomial time algorithm to compute the rank of a BCFL of scattered words generated by a BCFG.*

Theorem 7. *Let $G = (N, \Sigma, P, S, F)$ be a weakly ϵ-free BCFG with no useless nonterminals. Then $L^\infty(G)$ contains only well-ordered words iff for each strongly connected component $\mathcal{S}$ of Γ_G containing a nonterminal in F and for each production $A \to p$ with $A \in \mathcal{S}$, if p contains a nonterminal in $\mathcal{S}$ then it contains a single occurrence of such a nonterminal, and moreover, this nonterminal is the rightmost letter of p.*

Corollary 8. *It is decidable in polynomial time whether the language $L^\infty(G)$ generated by a given BCFG G contains only well-ordered words.*

Theorem 8. *Suppose that $G = (N, \Sigma, P, S.F)$ is a weakly ϵ-free BCFG without useless nonterminals and chain productions. Then $L^\infty(G)$ consists of finite and ω-words iff the following holds: Whenever $\mathcal{S}$ is a strongly connected component of Γ_G containing a nonterminal in F such that for at least one production whose left side is in $\mathcal{S}$, the right side of the production contains a nonterminal in $\mathcal{S}$, and whenever $A \in \mathcal{S}$, then there is no finite derivation $S \Rightarrow^* pAp'$ for any words $p, p' \in (N \cup \Sigma)^*$ such that $p' \neq \epsilon$.*

Corollary 9. *It can be decided in polynomial time whether the language generated by a BCFG contains only finite or ω-words, or only ω-words.*

Theorem 9. *It is decidable in polynomial time for a BCFG $G = (N, \Sigma, P, S, F)$ whether each word in $L^\infty(G)$ is dense.*

7 A Comparison

In this section, we compare the class of regular ω-languages [19] and the class of context-free ω-languages as defined by Cohen and Gold [12] with the class of those ω-languages that are BCFL's.

Recall that a Büchi automaton is a system $\mathbf{A} = (Q, \Sigma, \delta, q_0, F)$ which consists of an alphabet Q of states, an alphabet Σ of letters, a transition relation $\delta \subseteq Q \times \Sigma \times Q$, an initial state $q_0 \in Q$ and a set F of repeated states. A run of the automaton $\mathbf{A}$ on a word $w = a_0a_1 \ldots \in \Sigma^\omega$ is a sequence of states $q_0, q_1, \ldots$ where q_0 is the initial state and $(q_i, a_i, q_{i+1}) \in \delta$ holds for all i. Moreover, it is required that at least one state in F occurs infinitely often in the run. The automaton $\mathbf{A}$ accepts the language $L(\mathbf{A}) \subseteq \Sigma^\omega$ consisting of those words having at least one run. An ω-language is regular if some Büchi automaton accepts it.

Proposition 7. *Every regular language $L \subseteq \Sigma^\omega$ is a BCFL.*

Theorem 10. *An ω-language is a BCFL if and only if it is context-free in the sense of Cohen and Gold [12].*

Remark 3. The papers [8,3] define finite automata acting on infinite words and using this automaton model, provide a definition of recognizable languages of both countable words and all words with no upper bound on the cardinality of the word. Here we briefly compare BCFL's with the class REC of recognizable languages of countable words. On one hand, for any alphabet Σ, the set of all well-ordered words in Σ^∞ is in REC but not in BCFL. On the other hand, any nonregular context-free language in Σ^* is a BCFL which is not in REC. Thus, the two classes REC and BCFL are incomparable.

8 An Undecidable Property

The main result of this section is that for any fixed alphabet Σ, it is undecidable whether a BCFL given by a BCFG is the universal language Σ^∞.

First we note that the language $\Sigma^{+\infty} = \Sigma^\infty \Sigma \Sigma^\infty$ of all nonempty words in Σ^∞ is a BCFL. Next, the set of all words in Σ^∞ with no first letter is also a BCFL since it can be given as $(\Sigma^{+\infty})^{-\omega} \cup \{\epsilon\}$. Consider now the set of all words in Σ^∞ having a first letter. This set can be subdivided into two sets: 1. All words starting with an ω-word which is a BCFL given by $\Sigma^\omega \Sigma^\infty$. 2. All words starting with a nonempty finite word followed by a word that does not have a first letter. This is again a BCFL given by the expression $\Sigma^+((\Sigma^{+\infty})^{-\omega} \cup \{\epsilon\})$.

Suppose now that $G = (N, \Sigma, P, S)$ is an ordinary CFG with no ϵ-productions generating the language of finite words $L = L(G) \subseteq \Sigma^+$. Then consider the following language $L' \subseteq \Sigma^\infty$. L' consists of all words in Σ^∞ not having a first letter together with all words that start with an ω-word as well as those words starting with a finite word in L followed by a word not having a first letter. An expression for this language is $((\Sigma^{+\infty})^{-\omega} \cup \{\epsilon\}) \cup \Sigma^\omega \Sigma^\infty \cup L((\Sigma^{+\infty})^{-\omega} \cup \{\epsilon\})$, showing that L' is a BCFL.

Lemma 3. *$L' = \Sigma^\infty$ iff $L = \Sigma^+$.*

Since it is undecidable for an ordinary context-free grammar without ϵ-productions over a fixed alphabet of size at least two whether it generates the language of all finite nonempty words, and since BCFL's are effectively closed under the operations that appear in the above expressions, we immediately have that the universality problem is undecidable for BCFL's.

Proposition 8. *Let Σ be an alphabet of size at least two. Then it is undecidable for a BCFG $G = (N, \Sigma, P, S, F)$ whether $L^\infty(G) = \Sigma^\infty$.*

Theorem 11. *It is undecidable for a BCFG G over the unary alphabet $\{a\}$ whether $L^\infty(G) = \{a\}^\infty$.*

9 Müller Context-Free Languages

In this section we define context-free grammars with Müller acceptance condition and show that their generative power strictly exceeds the generating power of context-free grammars with Büchi acceptance condition.

Definition 8. *A context-free grammar with Müller acceptance condition, or MCFG is a system* $G = (N, \Sigma, P, S, \mathcal{F})$ *where* (N, Σ, P, S) *is an (ordinary) CFG and* $\mathcal{F}$ *is a set of subsets of* N*.*

When G is such an MCFG, a derivation tree t over G is defined as for BCFG's except that we require that for every infinite path π of t, the set of nonterminals occurring infinitely often as a vertex label along π belongs to $\mathcal{F}$. We write $X \Rightarrow^\infty p$ when there is a derivation tree with root symbol X and frontier word p.

Definition 9. *Let* $G = (N, \Sigma, P, S, \mathcal{F})$ *be an MCFG. The language* $L^\infty(G)$ *generated by* G *is the collection of all words* $u \in \Sigma^\infty$ *that are frontier words of some derivation tree whose root symbol is* S*. A language* $L \subseteq \Sigma^\infty$ *is called a Müller context-free language, or an MCFL, if* L *is generated by some MCFG.*

Theorem 12. *BCFL is strictly included in MCFL.*

In fact, an MCFL that is not a BCFL is provided by Corollary 6.

10 Conclusion and Further Research Topics

We have defined two types of context-free grammars generating languages of countable words, BCFG's and MCFG's, corresponding to the Büchi- and Müller-type acceptance conditions of automata on ω-words and automata on infinite trees. We showed that BCFG's can be transformed into equivalent BCFG's that are (weakly) ϵ-free and do not have chain productions or useless nonterminals. We established several closure properties of the class BCFL of languages that can be generated by BCFG's. We proved that many properties, including several order theoretic properties of BCFL's are decidable in polynomial time, whereas the universality problem is undecidable even for the single letter alphabet. We showed that the BCFL's of finite words are exactly the usual CFL's, and that the ω-languages that are BCFL's are exactly the context-free ω-languages of Cohen and Gold [12]. We showed that every BCFL of scattered words consists of words of finite bounded rank. Finally we showed that there is a language that can be generated by an MCFG which is not a BCFL.

It follows from our proof of Theorem 4 that it is decidable in polynomial time whether a finite word belongs to the language generated by a BCFG. The same question for regular words seems very interesting, where a regular word may be defined as a word generated by a BCFG which contains exactly one production for each nonterminal.

The present paper focuses on BCFG's and BCFL's. It would be interesting to see how much differently MCFG's behave. We have seen that they have a strictly larger generative power, and they also have different algorithmic properties. It would also be interesting to develop a suitable pushdown automaton model.

References

1. Bedon, N.: Finite automata and ordinals. Theor. Comp. Sci. 156, 119–144 (1996)
2. Bedon, N., Bès, A., Carton, O., Rispal, C.: Logic and rational languages of words indexed by linear orderings. In: Hirsch, E.A., Razborov, A.A., Semenov, A., Slissenko, A. (eds.) CSR 2008. LNCS, vol. 5010, pp. 76–85. Springer, Heidelberg (2008)
3. Bès, A., Carton, O.: A Kleene theorem for languages of words indexed by linear orderings. In: De Felice, C., Restivo, A. (eds.) DLT 2005. LNCS, vol. 3572, pp. 158–167. Springer, Heidelberg (2005)
4. Bloom, S.L., Choffrut, Ch.: Long words: the theory of concatenation and ω-power. Theor. Comp. Sci. 259, 533–548 (2001)
5. Bloom, S.L., Ésik, Z.: Axiomating omega and omega-op powers of words. Theor. Inform. Appl. 38, 3–17 (2004)
6. Bloom, S.L., Ésik, Z.: The equational theory of regular words. Inform. and Comput. 197, 55–89 (2005)
7. Bloom, S.L., Ésik, Z.: Regular and algebraic words and ordinals. In: Mossakowski, T., Montanari, U., Haveraaen, M. (eds.) CALCO 2007. LNCS, vol. 4624, pp. 1–15. Springer, Heidelberg (2007)
8. Bruyère, V., Carton, O.: Automata on linear orderings. J. Comput. System Sci. 73, 1–24 (2007)
9. Büchi, J.R.: Weak second-order arithmetic and finite automata. Z. Math. Logik und Grundlagen Math. 6, 66–92 (1960)
10. Büchi, J.R.: The monadic second order theory of ω_1. In: Decidable theories, II. Lecture Notes in Math., vol. 328, pp. 1–127. Springer, Heidelberg (1973)
11. Choueka, Y.: Finite automata, definable sets, and regular expressions over ω^n-tapes. J. Comput. System Sci. 17(1), 81–97 (1978)
12. Cohen, R.S., Gold, A.Y.: Theory of ω-languages, parts one and two. Journal of Computer and System Science 15, 169–208 (1977)
13. Courcelle, B.: Frontiers of infinite trees. RAIRO Theor. Inf. 12, 319–337 (1978)
14. Doner, J.: Tree acceptors and some of their applications. J. Comput. System Sci. 4, 406–451 (1970)
15. Ésik, Z., Ito, M., Kuich, W.: Linear languages of finite and infinite words (to appear)
16. Gécseg, F., Steinby, M.: Tree automata. Akadémiai Kiadó, Budapest (1984)
17. Heilbrunner, S.: An algorithm for the solution of fixed-point equations for infinite words. RAIRO Theor. Inf. 14, 131–141 (1980)
18. Nivat, M.: Sur les ensembles de mots infinis engendrés par une grammaire algébrique (French). RAIRO Inform. Théor. 12(3), 259–278 (1978)
19. Perrin, D., Pin, J.-E.: Infinite Words. Elsevier, Amsterdam (2004)
20. Rabin, M.O.: Decidability of second-order theories and automata on infinite trees. Trans. Amer. Math. Soc. 141, 1–35 (1969)
21. Rosenstein, J.G.: Linear Orderings. Academic Press, London (1982)
22. Thatcher, J.W., Wright, J.B.: Generalized finite automata theory with an application to a decision problem of second-order logic. Math. Systems Theory 2, 57–81 (1968)
23. Thomas, W.: On frontiers of regular sets. RAIRO Theor. Inf. 20, 371–381 (1986)
24. Thomas, W.: Automata on infinite objects. In: Handbook of Theoretical Computer Science, vol. B, pp. 135–191. Elsevier, Amsterdam (1990)
25. Wojciechowski, J.: Classes of transfinite sequences accepted by finite automata. Fundamenta Informaticæ 7, 191–223 (1984)
26. Wojciechowski, J.: Finite automata on transfinite sequences and regular expressions. Fundamenta Informaticæ 8, 379–396 (1985)