ON THE ACTIVE AND FULL USE OF MEMORY IN RIGHT-BOUNDARY GRAMMARS AND PUSH-DOWN AUTOMATA

by
A. Ehrenfeucht, H. J. Hoogeboom and G. Rozenberg

CU-CS-311-86 September, 1985

University of Colorado, Department of Computer Science, Boulder, Colorado.

ON THE ACTIVE AND FULL USE OF MEMORY IN RIGHT-BOUNDARY GRAMMARS AND PUSH-DOWN AUTOMATA

by

A. Ehrenfeucht*, H.J. Hoogeboom** and G. Rozenberg***

All correspondence to the third author.

*University of Colorado, Department of Computer Science, Boulder, Colorado

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

***Institute of Applied Mathematics and Computer Science, University of Leiden, Leiden, The Netherlands and University of Colorado, Department of Computer Science, Boulder, Colorado.

ABSTRACT

A *coordinated pair system* (*cp system* for short) consists of a pair of grammars the first of which is right-linear (*rl*) and the second right-boundary (*rb*). A *right-boundary grammar* is like a right-linear grammar except that one does not distinguish between terminal and nonterminal symbols - still the rewriting is applied to the last symbol of a string only (and erasing productions are allowed). A rewriting in a cp system consists of a pair of rewritings: one in the first and one in the second grammar - such a rewriting is possible if the pair of productions involved is in the finite set of *rewrites* given with the system. It is easily seen that cp systems correspond very closely to (are another formulation of) push-down automata: the right-linear component models the input and the finite state control while the rb component models the push-down store.

A rb grammar G transforms (rewrites) strings which are stored in a one-way (potentially infinite) tape. If one observes during a derivation δ the use of a fixed n-th cell of the tape and one notes the symbol stored there, each time that (the contents of) the cell is rewritten, then one gets the *n-active record of δ*; the set of all n-active records for all successful derivations δ forms the n-active language of G, denoted $ACT_n(G)$. It is proved that for each rb grammar G and each $n \in \mathbf{N}^+$, $ACT_n(G)$ is regular and moreover, for each $M \subseteq \mathbf{N}^+$, $\bigcup_{n \in M} ACT_n(G)$ is regular.

Another way to register the use of memory during a derivation δ is to record the contents of (a fixed) n-th cell during *all* consecutive steps of δ - in this way one gets the n-full record of δ. The set of all n-full records for all successful derivations δ forms the n-full record language of G, denoted $FR_n(G)$. It is proved that, as in the case of active records, $FR_n(G)$ regular for each n and, *unlike* in the case of active records, $\bigcup_{n \in M} FR_n(G)$ does not have to be regular

even if $M = \mathbf{N}$ (actually one can get arbitrarily complex languages in this way).

Then we provide a representation theorem allowing one to represent a cp system by a rb grammar and using this theorem we transfer the above results on the use of memory to cp systems.

INTRODUCTION.

The literature is full of various notions of machines (automata) and grammars each one developed with a specific, practical or theoretical, motivation behind it (see, e.g., [H] and [S2]). The notion of an *ects system* provides a common framework for quite a variety of these models (see [R]). Within the ects model various notions of *machines and grammars* are considered as systems of basic units (which are rather simple rewriting systems working together in a "coordinated fashion"). It is demonstrated in [R] that *right-boundary grammars (rb grammars* for short) constitute such a basic (perhaps the most basic) unit. A right-boundary grammar is like a *right-linear grammar* except that one does not distinguish between terminal and nonterminal symbols - still the rewriting is applied to the last symbol of a string only (and erasing productions are allowed); the notion of a rb grammar is a special case of the *regular canonical system* of Buchi (see [B] and [S1]). A well-known subclass of ects systems are *coordinated pair systems (cp systems* for short). A cp system consists of two grammars the first of which is right-linear and the second is right-boundary; it turns out that cp systems correspond very closely to (are another formulation of) *push-down automata*. The theory of cp systems (or: the cp system approach to the theory of push-down automata) is presented in [EHR1] and [EHR2].

This paper continues the research on the theory of cp systems and in particular it presents results describing the use of memory in right-boundary grammars (and cp systems). The basic idea investigated in the paper is as follows.

A right-boundary grammar G represents (transformations of) a data structure which is a linear one-way (potentially) infinite array of (memory) cells the processing of which takes place at the (right) end of the array. Hence during each derivation in G one can record the history of the use (the "scheduling") of each single cell. In other words each time (the contents of) a given cell is rewritten a note is made of the letter being stored there at that time (the *active* letter at this moment) and the sequence of all such "notes of activity" during a given derivation δ forms the *active record* of this cell during the derivation δ. The set of all active records of the n-th memory cell in all successful derivations forms the *n-active language of G*, denoted $ACT_n(G)$ (a derivation is *successful* if it leads from the axiom of G to the empty word).

Another, very natural, approach to recording the memory use is to apply a standard 'snapshot' approach. Here observing the n-th memory cell means to record the contents of this cell during all consecutive steps of the derivation. In this way we obtain, for a given derivation δ and for a given $n \in \mathbf{N}^+$, the *n-full record* of δ (where if a line of δ is shorter than n, then we insert the $\$$ symbol symbolizing the idle state of the given cell). The set of all n-full records of the n-th memory cell in all successful derivations forms the *n-full record language of G*, denoted $FR_n(G)$.

We prove that for each rb grammar G and each n, both $ACT_n(G)$ and $FR_n(G)$ are regular (Corollary 1.3 and Theorem 4.4). Actually the regularity of active records is quite "deep"; it turns out that for an arbitrary subset M of positive integers $\bigcup_{m \in M} ACT_m(G)$ is regular (Theorem 3.4) - this is strong regularity indeed!! The situation is drastically different for full records: infinite unions $\bigcup_{m \in M} FR_m(G)$ do not have to be regular (even if M is taken to be the set of *all* positive integers).

In order to transfer these results to cp systems we prove a rb representation theorem for cp systems (Theorem 6.1): rather than to consider a cp system one can consider a rb grammar. This representation theorem allows us to transfer the above results on active and full records for rb grammars to the framework of cp systems.

0. PRELIMINARIES

We assume the reader to be familiar with basic formal language theory (see, e.g., [H] or [S2]).

For a set Z, $\#Z$ denotes its cardinality. If V is a finite set of integers we use max V and min V to denote the maximal and the minimal element of V respectively.

For a word x, $|x|$ denotes its length and, if $1 \leq k \leq |x|$, then $x(k)$ denotes the k-th letter of x. If x is nonempty, then we use $last(x)$ to denote $x(|x|)$. Λ denotes the empty word.

A letter to letter homomorphism is called a *coding*.

A *context-free grammar*, abbreviated *cf grammar*, is specified in the form $G = (\Sigma, P, S, \Delta)$, where Σ is its alphabet, P its set of productions, $S \in \Sigma$ its axiom and Δ its terminal alphabet. For $x, y \in \Sigma^*$ we write $x \overset{\pi}{\underset{G}{\Longrightarrow}} y$ if x directly derives y using production π.

A *right-linear grammar*, abbreviated *rl grammar*, is a context-free grammar $G = (\Sigma, P, S, \Delta)$ which has its productions in the set $(\Sigma - \Delta) \times \Delta^* ((\Sigma - \Delta) \cup \{\Lambda\})$.

1. RIGHT-BOUNDARY GRAMMARS AND THEIR ACTIVE RECORDS

In this section we introduce basic notions concerning right-boundary grammars and then we formalize the (active) use of memory by derivations in these grammars.

Definition 1.1. A *right-boundary grammar*, abbreviated *rb grammar*, is a triple $G = (\Sigma, P, \omega)$, where

Σ is an *alphabet*,

$P \subseteq \Sigma \times \Sigma^*$ is a finite set of *productions*, and

$\omega \in \Sigma^+$ is the *axiom* of G. ∎

For a rb grammar $G = (\Sigma, P, \omega)$ we use $maxr(G)$ to denote $\max\{|w| \mid A \rightarrow w \in P\}$.

Definition 1.2. Let $G = (\Sigma, P, \omega)$ be a rb grammar.

(1) Let $x, y \in \Sigma^*$ and let $\pi = A \rightarrow w \in P$. x *directly derives* y *in* G *(using* π*)*, written $x \underset{G}{\Longrightarrow} y$ $(x \overset{\pi}{\underset{G}{\Longrightarrow}} y)$, if $x = zA$ and $y = zw$ for some $z \in \Sigma^*$.

Let $\overset{*}{\underset{G}{\Longrightarrow}}$ be the reflexive and transitive closure of $\underset{G}{\Longrightarrow}$. If $x \overset{*}{\underset{G}{\Longrightarrow}} y$, then we say that x *derives* y *in* G.

(2) A *derivation (in* G*)* is a sequence $\delta = (x_0, x_1, \ldots, x_n)$, $n \geq 0$, of words from Σ^* such that, for every $1 \leq i \leq n$, $x_{i-1} \underset{G}{\Longrightarrow} x_i$. We say that δ *derives* x_n *from* x_0 and denote it by $\delta: x_0 \overset{*}{\underset{G}{\Longrightarrow}} x_n$.

For $0 \leq i \leq n$, x_i is called the *i-th line of* δ and is denoted by $\delta(i)$. n is called the *length of* δ *and is denoted* by $lg(\delta)$.

(3) A derivation $\delta : \omega \overset{*}{\underset{G}{\Longrightarrow}} \Lambda$ is called *successful.*

(4) Let $\delta_1 = (\delta_1(0), \delta_1(1), \ldots, \delta_1(m))$ and $\delta_2 = (\delta_2(0), \delta_2(1), \ldots, \delta_2(n))$ be two

derivations in G such that $\delta_1(m) = \delta_2(0)$. The *composition of* δ_1 *and* δ_2, denoted $\delta_1 \otimes \delta_2$, is the derivation $(\delta_1(0), \delta_1(1), \ldots, \delta_1(m), \delta_2(1), \ldots, \delta_2(n))$. ∎

Lemma 1.1. Let $G = (\Sigma, P, \omega)$ be a rb grammar and let $x, y \in \Sigma^+$. If $x \underset{G}{\Longrightarrow} y$, then there exists a unique production $\pi \in P$ such that $x \overset{\pi}{\underset{G}{\Longrightarrow}} y$. ∎

Definition 1.3. Let G be a rb grammar and let $\delta = (\delta(0), \ldots, \delta(n))$, $n \geq 0$ be a derivation in G. The sequence $(\pi_1, \ldots, \pi_n)$ of productions such that $\delta(i-1) \overset{\pi_i}{\underset{G}{\Longrightarrow}} \delta(i)$ for every $1 \leq i \leq n$ is called the *control sequence of* δ and is denoted by $cont(\delta)$; if $n = 0$, then $cont(\delta)$ is the empty sequence. ∎

Remark 1.1. (1) Lemma 1.1 guarantees the uniqueness of the control sequence (for each derivation δ).

(2) Note that if $\delta_1 : u \overset{*}{\underset{G}{\Longrightarrow}} v$ and $\delta_2 : v \overset{*}{\underset{G}{\Longrightarrow}} w$ are two derivations in a rb grammar G, then

$lg(\delta_1 \otimes \delta_2) = lg(\delta_1) + lg(\delta_2)$ and

$cont(\delta_1 \otimes \delta_2) = cont(\delta_1) cont(\delta_2)$. ∎

In order to simplify our notation we will skip the inscription "G" whenever G is understood from the context. Hence, e.g., we will write $\Longrightarrow$ and $\overset{*}{\Longrightarrow}$ rather than $\underset{G}{\Longrightarrow}$ and $\overset{*}{\underset{G}{\Longrightarrow}}$ respectively.

If all the lines of a derivation δ in a rb grammar G are written under each other (adjusted letter-by-letter), then the most natural way of storing δ in a memory suggests by itself that all the first letters of the lines of δ are stored in the first memory cell, all the second letters are stored in the second memory cell, etc.

This can be depicted as follows:

Figure 1.

Now if one wants to get an idea of the use of memory within this particular derivation δ, then one can choose $n \geq 1$ and then observe the actions performed on the n-th memory cell. The significant moments are those when this particular memory cell becomes *active* (i.e. the symbol stored there is rewritten).

This natural intuition of the memory use associated with a derivation in a right-boundary grammar leads us to the following definition.

Definition 1.4. Let $\delta = (\delta(0), \ldots, \delta(k))$ be a derivation in a rb grammar G and let $n \in \mathbf{N}^+$.

(1) A line $\delta(i)$ of δ with $|\delta(i)| = n$ is called an *n-active* line of δ.

(2) The *n-active record of* δ, denoted $act_n(\delta)$, is the word $\varphi_n(\delta(0))\varphi_n(\delta(1)) \cdots \varphi_n(\delta(k-1))$, where $\varphi_n : \Sigma^* \to \Sigma \cup \{\Lambda\}$ is the mapping defined by

$$\varphi_n(u) = \begin{cases} u(n), & \text{if } |u| = n, \\ \Lambda & \text{, otherwise.} \end{cases}$$ ■

Remark 1.2. (1) Note that in determining the n-active record of δ the last word of δ is not taken into account. Therefore:

(2) If $\delta_1 : u \overset{*}{\Longrightarrow} v$, $\delta_2 : v \overset{*}{\Longrightarrow} w$ are two derivations in a rb grammar G, then $act_n(\delta_1 \otimes \delta_2) = act_n(\delta_1) act_n(\delta_2)$. ■

Example 1.1. Let $G = (\{A, B, C\}, P, A)$ be the rb grammar with $P = \{A \to BC, B \to AB, B \to \Lambda, C \to BB\}$.

Then $\delta = (BBB, BB, BAB, BAAB, BAA, BABC, BABBB, BABB, BAB)$ is a derivation in G of length 8 with

$cont(\delta) = (B \to \Lambda, B \to AB, B \to AB, B \to \Lambda, A \to BC, C \to BB, B \to \Lambda, B \to \Lambda)$,

$act_1(\delta) = \Lambda$, $act_2(\delta) = B$, $act_3(\delta) = BBA$, $act_4(\delta) = BCB$, $act_5(\delta) = B$ and $act_n(\delta) = \Lambda$ for each $n \geq 6$.

This is easily seen if we write the consecutive lines of δ under each other.

Figure 2.

■

If we record now, for an $n \geq 1$, the n-active records for *all* successful derivations in a rb grammar G, then we get a complete picture of the use of the n-th memory cell (from our "intuitive model") in G.

Definition 1.5. Let $G = (\Sigma, P, \omega)$ be a rb grammar and let $n \in \mathbf{N}^+$. The *n-active language of G*, denoted $ACT_n(G)$, is the language $\{act_n(\delta) \mid \delta : \omega \overset{*}{\underset{G}{\Longrightarrow}} \Lambda\}$. ■

We demonstrate now that for each n, $ACT_n(G)$ is regular, which intuitively means that this "schedule of active use" of each memory cell may be realized (implemented) by a finite automaton. Actually we can prove a somewhat stronger result.

Theorem 1.2. Let $G = (\Sigma, P, \omega)$ be a rb grammar and let $w_1, w_2 \in \Sigma^*$. Then $\{act_n(\delta) \mid \delta : w_1 \overset{*}{\underset{G}{\Longrightarrow}} w_2\}$ is regular for each $n \in \mathbf{N}^+$.

Proof.

Let $n \in \mathbf{N}^+$. We construct a finite automaton $\mathbf{A}_n = (Q_n, \Sigma, \Pi_n, I_n, F_n)$ as follows.

$\mathbf{A}_n$ has as its set of states the set of words of length n over Σ together with w_2 if $|w_2| \neq n$; i.e., $Q_n = \Sigma^n \cup \{w_2\}$.

For $u \in \Sigma^n$, $v \in Q_n$ and $A \in \Sigma$ there exists an edge $(u, A, v) \in \Pi_n$ if and only if there exists a derivation $\delta : u \overset{*}{\underset{G}{\Longrightarrow}} v$ such that $act_n(\delta) = A$. (Note that then $A = u(n)$.) If $|w_2| \neq n$, then w_2 has no outgoing edges.

I_n consists of those $u \in Q_n$ for which there exists a derivation $\delta : w_1 \overset{*}{\Longrightarrow} u$ such that $act_n(\delta) = \Lambda$. (Note that if $|w_1| = n$, then $I_n = \{w_1\}$.)

$F_n = \{w_2\}$.

$L(\mathbf{A}_n)$ is regular and obviously $L(\mathbf{A}_n) = \{act_n(\delta) \mid \delta : w_1 \overset{*}{\underset{G}{\Longrightarrow}} w_2\}$. ■

Remark 1.3. The effectiveness of the construction of the automaton $\mathbf{A}_n$ from the above proof relies on deciding whether or not, for $u, v \in \Sigma^n$, there exists a derivation $\delta : u \overset{*}{\underset{G}{\Longrightarrow}} v$ with $act_n(\delta) = A$, $A = u(n)$. It is obvious that such a derivation cannot have lines of length n other than its first (we mean $\delta(0)$) and last lines. Moreover it is easily seen that, for all p, j such that $p \leq j < lg(\delta)$, $|\delta(p)| > n$ implies $|\delta(j)| > n$. Hence it follows that we may write $\delta = \delta_1 \otimes \delta_2$ where $\delta_1 : u \overset{*}{\Longrightarrow} w$, $\delta_2 : w \overset{*}{\Longrightarrow} v$ are such that $|\delta_1(i)| < n$ for $0 < i < lg(\delta_1)$ and $|\delta_2(i)| > n$ for $0 \leq i < lg(\delta_2)$. Note that if δ contains no lines shorter (longer) than n, then $lg(\delta_1) = 1(lg(\delta_2) = 0$ respectively). All lines in δ_2 have the first n symbols in common since they are not rewritten during δ_2. Thus $w = vz$ for some $z \in \Sigma^*$, $0 \leq |z| < maxr(G)$ and there exists a derivation $\mu : z \overset{*}{\Longrightarrow} \Lambda$ with $cont(\mu) = cont(\delta_2)$.

In this way the problem of deciding whether or not $(u, A, v) \in \Pi_n$ is reduced to the problem of deciding whether or not $z \overset{*}{\underset{G}{\Longrightarrow}} \Lambda$ for a given word z. The latter problem is easily seen to be decidable. The reasoning as above can be extended in a straightforward way to the problem of deciding whether or not $u_1 \overset{*}{\underset{G}{\Longrightarrow}} u_2$ for arbitrary $u_1, u_2 \in \Sigma^*$. (If $u_2 \neq \Lambda$, then take $n = |u_2|$; every derivation $\delta : u_1 \overset{*}{\underset{G}{\Longrightarrow}} u_2$ can be decomposed into an initial part leading to the first line of length n and a number of derivations of the form discussed above.) Thus the construction of the automaton $\mathbf{A}_n$ is effective. ∎

Corollary 1.3. Let G be a rb grammar and let $n \in \mathbf{N}^+$. Then $ACT_n(G)$ is regular. ∎

Example 1.1. (continued) A finite automaton accepting $ACT_3(G)$ can easily be constructed using the following diagram. This diagram has "virtual"nodes to represent derivations with lines of length less than 3. Note that $X \overset{*}{\underset{G}{\Longrightarrow}} \Lambda$ for every $X \in \Sigma$.

Figure 3.

The finite automaton for $ACT_3(G)$ looks as follows.

Figure 4.

∎

2. SPACE USED BY DERIVATIONS IN RB GRAMMARS

We have learned in the last section that for an arbitrary rb grammar $ACT_n(G)$ is regular for each n. As a matter of fact we are going to prove (in Section 3) that the regularity of the use of memory cells in rb grammars is much deeper than that: it turns out that for an arbitrary subset M of $\mathbf{N}^+$ the union $\bigcup_{m \in M} ACT_m(G)$ is also regular - this is strong regularity indeed !

In this section we prove an auxiliary technical result (Lemma 2.1) that is interesting on its own: in deriving a word v from a word u in a rb grammar it suffices to use (in addition to the space occupied by u and v) no more than some constant (for the grammar) extra amount of space.

The amount of space used by a derivation is formalized as follows.

Definition 2.1. Let $G = (\Sigma, P, \omega)$ be a rb grammar.

(1) The *breadth* of a derivation δ in G, denoted $brd(\delta)$, is $\max\{|\delta(i)| \mid 0 \leq i \leq lg(\delta)\}$.

(2) Let $u \overset{*}{\underset{G}{\Longrightarrow}} v$ for some $u, v \in \Sigma^*$. The (u,v)*-breadth*, denoted $brd(u,v)$, equals $\min\{brd(\delta) \mid \delta : u \overset{*}{\underset{G}{\Longrightarrow}} v\}$. ∎

Lemma 2.1. Let $G = (\Sigma, P, \omega)$ be a rb grammar. There exists an integer m_G such that for each pair $u, v \in \Sigma^*$, $u \overset{*}{\underset{G}{\Longrightarrow}} v$ implies that $brd(u,v) \leq m_G + \max\{|u|, |v|\}$.

Proof.

Let $m_G = \max\{brd(w_1, w_2) \mid w_1 \overset{*}{\underset{G}{\Longrightarrow}} w_2 \text{ and } |w_1|, |w_2| \leq maxr(G)\}$. Note that $maxr(G) \leq m_G$.

Let $\delta : u \overset{*}{\Longrightarrow} v$ be an arbitrary derivation in G. We will prove that there exists a derivation $\delta' : u \overset{*}{\Longrightarrow} v$ in G, with $brd(\delta') \leq m_G + M$, where $M = \max\{|u|, |v|\}$.

A line $\delta(i)$ of δ is called *special* if $M < |\delta(i)| \leq M + maxr(G)$. Let $\tau = (\delta(i_1), \delta(i_2), \ldots, \delta(i_t))$ be the subsequence of δ consisting of all special lines. Since $|\delta(j-1)| - 1 \leq |\delta(j)| < |\delta(j-1)| + maxr(G)$ for each $1 \leq j \leq lg(\delta)$, δ cannot have lines longer than $M + maxr(G)$ without having special lines. So if τ is empty then the lemma holds.

Assume now that τ is nonempty. Both $\delta_0 = (\delta(0), \ldots, \delta(i_1 - 1))$ and $\delta_{t+1} = (\delta(i_t + 1), \ldots, \delta(lg(\delta)))$ consist of lines not longer than M.

Consider now the subsequence $\delta_k = (\delta(i_k + 1), \ldots, \delta(i_{k+1} - 1))$ for some $k \in \{1,2,\ldots,t\}$. Either all these lines are not longer than M or they are all longer than $M + maxr(G)$.
Assume that δ_k is nonempty and that all its lines are longer than $M + maxr(G)$ - we say then that δ_k is an *external segment* of δ. We can write $\delta(i_k) = xw_1$ and $\delta(i_{k+1}) = xw_2$, where $|x| = M$, $|w_1|, |w_2| \leq maxr(G)$. Note that $w_1 \overset{*}{\underset{G}{\Longrightarrow}} w_2$, because $\delta(i_k) \overset{*}{\underset{G}{\Longrightarrow}} \delta(i_{k+1})$ and the derivation steps in-between do not influence (rewrite) x. From the definition of m_G it follows that there exists a derivation $\mu : w_1 \overset{*}{\Longrightarrow} w_2$, with $brd(\mu) \leq m_G$.

Consequently there exists a derivation $\mu' : \delta(i_k) = xw_1 \overset{*}{\Longrightarrow} xw = \delta(i_{k+1})$ with $cont(\mu) = cont(\mu')$ such that $brd(\mu') \leq m_G + M$.

Hence, by replacing the external segments of δ by "new" derivations as above, we obtain a derivation $\delta' : u \overset{*}{\Longrightarrow} v$, such that $brd(\delta') \leq m_G + M$. Thus the lemma holds. ▪

Example 1.1. (continued) Consider the following derivation $\delta : CAB \overset{*}{\Longrightarrow} \Lambda$:

Figure 5.

Special lines in this figure are indicated by short arrows. δ has one external segment; it is a derivation of $CAABBB$ from $CAABBB$. The "roof part" (i.e., the part above M, see Figure 6) represents the derivation
(BBB,$BBAB$,BBA,$BBBC$,$BBBBB$,$BBBB$,BBB).

Figure 6.

Replacing it by (BBB) yields the derivation δ'.

Figure 7.

■

3. UNIONS OF ACTIVE LANGUAGES

As we have indicated already, arbitrary unions of n-active languages in a rb grammar are regular. We prove this result in this section.

The reason that this result holds is that each n-active language is of a very special form (Lemma 3.3). We start by defining some classes of languages useful in our considerations.

Definition 3.1. A language K is *down-closed* if for each word $w \in K$ all sparse subwords of w are also in K. ■

The following well-known result (see [C], pp. 63-64) is very crucial in our proof of Lemma 3.3.

Proposition 3.1. Each down-closed language is regular. ■

We use DC_Θ to denote the family of down-closed languages over the alphabet Θ. For a regular substitution π, $DC_{\Theta,\pi}$ denotes the family $\{\pi(L) \mid L \in DC_\Theta\}$.

Lemma 3.2. Let π be a regular substitution over an alphabet Θ. Then

(1) each language in $DC_{\Theta,\pi}$ is regular, and

(2) $DC_{\Theta,\pi}$ is closed under arbitrary (possibly infinite) unions.

Proof. (1) Obvious. By Proposition 3.1 DC_Θ consists of regular languages and regularity is preserved under regular substitutions.

(2) DC_Θ is clearly closed under arbitrary unions. Consequently $DC_{\Theta,\pi}$ is closed under arbitrary unions because

$$\bigcup_{L \in \mathbf{F}} \pi(L) = \pi(\bigcup_{L \in \mathbf{F}} L)$$

for any language family $\mathbf{F}$ over Θ. ■

Lemma 3.3. Let $G = (\Sigma, h, \omega)$ be a rb grammar. There exist an integer n_G, an alphabet Θ and a regular substitution π of Θ into Σ^* such that $ACT_n(G) \in DC_{\Theta,\pi}$ for each $n > n_G$.

Proof.

Let m_G be a constant (dependent of G) satisfying the statement of Lemma 2.1, let $m_0 = m_G+1$ and let $n_G = m_0+|\omega|$. Let $n > n_G$.

Consider a derivation $\delta : \omega \overset{*}{\Longrightarrow} \Lambda$ and let $\delta(i)$ and $\delta(j)$, $i \leq j$, be two n-active lines of δ. We say that $\delta(i)$ and $\delta(j)$ are *n-related* if $|\delta(t)| > n-m_0$ for each $i \leq t \leq j$.

Assume now that $\delta(i)$ and $\delta(j)$ are two n-related lines of δ. Since all lines of δ between $\delta(i)$ and $\delta(j)$ are longer than $n-m_0$ they all have a common prefix x of length $n-m_0$. Hence for each $i \leq t \leq j$ we can write $\delta(t) = xw_t$, where $|x| = n-m_0$ and $w_i \Longrightarrow w_{i+1} \Longrightarrow \cdots \Longrightarrow w_j$. Moreover $|w_i| = |w_j| = m_0$.

Obviously the notion of n-related lines defines an equivalence relation on the (occurences of) n-active lines of δ. If we take the first $(\delta(i))$ and last $(\delta(j))$ line of each n-related equivalence class, then we obtain a "subsequence"

$\delta(i_1), \delta(j_1), \delta(i_2), \delta(j_2), \ldots, \delta(i_k), \delta(j_k)$

of δ, such that $0 \leq i_1 \leq j_1 < i_2 \leq j_2 < \cdots < i_k \leq j_k \leq lg(\delta)$; this subsequence is referred to as the *n-characteristic sequence of* δ. (Note that it may happen that $i_t = j_t$ because an equivalence class can consist of just one single n-active line of δ; in that case this line occurs twice in our sequence.)

Let $\Theta = \{<u,v> \mid u,v \in \Sigma^*, |u| = |v| = m_0\}$.

The word $<u_1,v_1><u_2,v_2> \cdots <u_k,v_k>$ over Θ such that u_t and v_t are the suffixes of length m_0 of $\delta(i_t)$ and $\delta(j_t)$ respectively, is called the *n-signature of* δ and denoted by $sig_n(\delta)$.

Now let $K_n = \{sig_n(\delta) \mid \delta : \omega \overset{*}{\Longrightarrow} \Lambda\}$.

The notions that we have introduced above can be illustrated as follows. Let δ be a derivation of the following form:

Figure 8.

Then the lines $\delta(i_1)$ and $\delta(j_1)$ are n-related, while the lines $\delta(j_1)$ and $\delta(i_2)$ are not n-related. Note that there are no other n-active lines related to $\delta(i_2)$.
The n-characteristic sequence of δ is the sequence
$\delta(i_1)$, $\delta(j_1)$, $\delta(i_2)$, $\delta(i_2)$, $\delta(i_3)$, $\delta(j_3)$.
The n-signature of δ is $<u_1,v_1><u_2,u_2><u_3,v_3>$.

First we will show that K_n is down-closed.

Claim 1. $K_n \in DC_\Theta$.

Proof of Claim 1.

In order to prove that K_n is down-closed, consider a derivation $\delta : \omega \overset{*}{\underset{G}{\Longrightarrow}} \Lambda$ with $sig_n(\delta) = <u_1,v_1><u_2,v_2> \cdots <u_k, v_k>$, $k \geq 1$.
Let $\delta(i_1)$, $\delta(j_1)$, ..., $\delta(i_k)$, $\delta(j_k)$ be the n-characteristic sequence of δ. Since $\delta(j_t)$ and $\delta(i_{t+1})$, $1 \leq t < k$, are not n-related they are separated by a line $\delta(l_t)$, $j_t < l_t < i_{t+1}$, with $|\delta(l_t)| \leq n - m_0$. Furthermore if we set $l_0 = 0$ and $l_k = lg(\delta)$, then obviously we have $l_0 < i_1$, $j_k < l_k$, $|\delta(l_0)| = |\omega| < n - m_0$ and $|\delta(l_{k+1})| = |\Lambda| < n - m_0$.

For each $1 \leq t \leq k$ we have $\delta(l_{t-1}) \overset{*}{\Longrightarrow} \delta(l_t)$. Thus, by Lemma 2.1, there exists a derivation $\mu_t : \delta(l_{t-1}) \overset{*}{\Longrightarrow} \delta(l_t)$ with $brd(\mu_t) \leq m_G + \max\{|\delta(l_{t-1})|, |\delta(l_t)|\} \leq m_G + n - m_0 = n - 1$. Hence $act_n(\mu_t) = \Lambda$ or, in other words, μ_t does not contain n-active lines.

So if we repace the lines $\delta(l_{t-1}), \ldots, \delta(l_t)$ from δ by the lines of μ_t, then we obtain a new derivation $\delta_t' : w \overset{*}{\Longrightarrow} \Lambda$ such that
$sig_n(\delta_t') = <u_1,v_1> \cdots <u_{t-1},v_{t-1}><u_{t+1},v_{t+1}> \cdots <u_k,v_k>$.

Consequently by erasing an arbitrary symbol in a word of K_n we obtain a word in K_n. Thus K_n is down-closed. ■

In order to illustrate the construction used above consider again the derivation depicted in Figure 8. According to our construction it is possible to replace the subderivation $(\delta(l_1), \delta(l_1+1), \ldots, \delta(l_2))$ of δ by a derivation μ_2 of $\delta(l_2)$ from $\delta(l_1)$ which looks as follows:

Figure 9.

The resulting derivation δ_2' has the n-signature $<u_1,v_1><u_3,v_3>$.

Let $\pi : \Theta \to \Sigma^*$ be the substitution defined by

$\pi(<u,v>) = \{wA \mid A = v(m_0)$ and $w = act_{m_0}(\mu)$ for some $\mu : u \overset{*}{\Longrightarrow} v\}$

Theorem 1.2 implies that π is a regular substitution.

In order to prove that $ACT_n(G) \in DC_{\Theta,\pi}$ we will demonstrate that $ACT_n(G) = \pi(K_n)$. This in turn is accomplished by proving the following two claims (corresponding to the two inclusions involved).

Claim 2. $ACT_n(G) \subseteq \pi(K_n)$.

Proof of Claim 2.

Let $w \in ACT_n(G)$, so $w = act_n(\delta)$ for some derivation $\delta : \omega \overset{*}{\underset{G}{\Longrightarrow}} \Lambda$. Let $\delta(i_1), \delta(j_1), \ldots, \delta(i_k), \delta(j_k)$ be the n-characteristic sequence of δ and let $sig_n(\delta) = <u_1,v_1> \cdots <u_k,v_k>$.

If we choose $l_0, l_1, \ldots, l_k$ as in the proof of Claim 1, then $act_n(\delta) = act_n(\delta_1) \cdots act_n(\delta_k)$, where each derivation δ_t, $1 \le t \le k$, is of the form $\delta_t = (\delta(l_{t-1}), \delta(l_{t-1}+1), \ldots, \delta(l_t))$.
Moreover, for each $1 \le t \le k$, $\delta_t = \delta_t^1 \otimes \delta_t^2 \otimes \delta_t^3$
where $\delta_t^1 = (\delta(l_{t-1}), \ldots, \delta(i_t))$, $\delta_t^2 = (\delta(i_t), \ldots, \delta(j_t))$ and $\delta_t^3 = (\delta(j_t), \ldots, \delta(l_t))$.
Thus $act_n(\delta_t) = act_n(\delta_t^1) act_n(\delta_t^2) act_n(\delta_t^3) = act_n(\delta_t^2) \cdot \delta(j_t)(n)$.

The claim now follows by observing that

$act_n(\delta_t^2) \cdot \delta(j_t)(n) = act_{m_0}(\mu_t) \cdot v_t(m_0)$ for the derivation $\mu_t : u_t \overset{*}{\underset{G}{\Longrightarrow}} v_t$ with $cont(\delta_t^2) = cont(\mu_t)$ and consequently $act_n(\delta_t) \in \pi(<u_t, v_t>)$; thus $act_n(\delta) \in \pi(sig_n(\delta)) \subseteq \pi(K_n)$. ■

Claim 3. $\pi(K_n) \subseteq ACT_n(G)$.

Proof of Claim 3.

If $wA \in \pi(<u,v>)$, where $A = v(m_0)$, then there exists a derivation $\mu : u \overset{*}{\Longrightarrow} v$ with $act_{m_0}(\mu) = w$. Thus for an arbitrary $x \in \Sigma^*$ with $|x| = n - m_0$ there exists a derivation $\delta' : xu \overset{*}{\Longrightarrow} xv$ with $act_n(\delta') = w$.

This enables us to replace in a derivation $\delta : \omega \overset{*}{\Longrightarrow} \Lambda$ each of the subderivations $\delta_t = (\delta(l_{t-1}), \ldots, \delta(l_t))$ (in the notation as above) by a corresponding derivation $\delta_t' : \delta(l_{t-1}) \overset{*}{\Longrightarrow} \delta(l_t)$ such that $act_n(\delta_t') = z_t$, where z_t is an arbitrarily chosen element of $\pi(<u_t, v_t>)$.
From these observations the claim easily follows. ■

So we have shown that $ACT_n(G) = \pi(K_n)$ for each $n > n_G = m_0 + |\omega|$, where K_n is a down-closed language over the alphabet Θ. Hence the lemma holds. ■

We are ready now to prove the main result of this section.

Theorem 3.4. Let G be a rb grammar. Then $\bigcup_{i \in I} ACT_i(G)$ is regular for arbitrary $I \subseteq \mathbf{N}^+$.

Proof.

Let $I \subseteq \mathbf{N}^+$. According to Lemma 3.3 there exists a constant n_G for G such that $ACT_n(G) \in DC_{\Theta,\pi}$ for every $n > n_G$, where π is a suitably chosen regular substitution over an alphabet Θ.

Let $I_1 = \{i \in I \mid i > n_G\}$ and $I_2 = \{i \in I \mid i \leq n_G\}$. From Lemma 3.2 it follows that $\bigcup_{i \in I_1} ACT_i(G)$ is regular.

Since $\bigcup_{i \in I} ACT_i(G) = \bigcup_{i \in I_1} ACT_i(G) \cup \bigcup_{i \in I_2} ACT_i(G)$, I_2 is finite and the class of regular languages is closed under finite unions, Corollary 1.3 implies that

$\bigcup_{i \in I} ACT_i(G)$ is regular. ∎

4. FULL RECORDS OF RB GRAMMARS

The n-th active record of a derivation contains the information about the active behavior of the n-th memory cell - we observe only those moments of time when the n-th cell is active and then we record the symbol being there.

Another, very natural, approach to recording the memory use is to apply a standard "snapshot" approach. Here observing the n-th memory cell means to record the contents of this cell during all consecutive steps of the derivation. In this way we obtain, for a given derivation δ and for a given $n \in \mathbf{N}^+$, the *n-full record* of δ (where if a line of δ is shorter than n, then we insert the $\$$ symbol symbolizing the idle state of the given cell). This is formalized as follows.

Definition 4.1. Let $G = (\Sigma, P, \omega)$ be a rb grammar, and let $\$$ be a symbol not in Σ. Let $n \in \mathbf{N}^+$.

(1) Let δ be a derivation in G and let $k = lg(\delta)$. The *n-full record of* δ, denoted $fr_n(\delta)$, is the word $\psi_n(\delta(0))\psi_n(\delta(1)) \cdots \psi_n(\delta(k-1))$ where $\psi_n : \Sigma^* \to \Sigma \cup \{\$\}$ is the mapping defined by

$$\psi_n(u) = \begin{cases} u(n), & |u| \geq n, \\ \$ & \text{, otherwise.} \end{cases}$$

(2) The *n-full record language of* G, denoted $FR_n(G)$, equals

$\{fr_n(\delta) \mid \delta : \omega \overset{*}{\underset{G}{\Longrightarrow}} \Lambda\}$ ■

Remark 4.1. As in the case of the n-active record, in defining the n-full record of a derivation δ we do not consider the last line of δ. Thus if

$\delta_1 : u \overset{*}{\Longrightarrow} v$ and $\delta_2 ; v \overset{*}{\Longrightarrow} w$ are derivations in a given rb grammar G, then $fr_n(\delta_1 \otimes \delta_2) = fr_n(\delta_1) fr_n(\delta_2)$. ■

Example 4.1. Let $G = (\{A, B, C, D\}, P, AC)$ be the rb grammar with $P = \{A \to BC, B \to \Lambda, C \to DC, C \to \Lambda, D \to \Lambda\}$.

For each pair $k, l \in \mathbf{N}$ there exists a derivation $\delta_{k,l}$ of the form $(AC, ADC, AD^2C, \ldots, AD^kC, AD^k, \ldots, AD, A, BC, BDC, \ldots, BD^lC, BD^l, \ldots, BD, B, \Lambda)$. Obviously $fr_1(\delta_{k,l}) = A^{2k+2}B^{2l+2}$.

Since each derivation $\delta : AC \overset{*}{\Longrightarrow} \Lambda$ is of the above form we have $FR_1(G) = (AA)^+(BB)^+$.

Consider $\delta = \delta_{6,3}$. Then

$fr_2(\delta) = CD^{12}\$CD^6\$$,

$fr_5(\delta) = \$^3CD^6\$^7C\$^4$,

$fr_8(\delta) = \$^6C\$^{15}$ and

$fr_n(\delta) = \$^{22}$ for each $n \geq 9$.

It is easily seen that $ACT_1(G) = \{AB\}$, $ACT_2(G) = \{CD, C\}^2$ and $ACT_n(G) = \{CD, C, \Lambda\}^2$ for each $n \geq 3$. ∎

For a rb grammar G and two words u, v over its alphabet the (u,v)-*spectrum* is the set of *all* lengths of *all* derivations in G leading from u to v. We will prove that (u,Λ)-spectrum is ultimately periodic for each word v. This result certainly says something about the nature of derivations in rb grammars. Moreover it will be an essential tool in proving the regularity of full record languages.

We begin by formally defining spectra.

Definition 4.2. Let $G = (\Sigma, P, \omega)$ be a rb grammar. For two words $u, v \in \Sigma^*$ the *(u,v)-spectrum (in G)*, denoted $spec_G(u,v)$, is defined by

$spec_G(u,v) = \{lg(\delta) \mid \delta : u \overset{*}{\underset{G}{\Longrightarrow}} v\}$. ∎

As usual we will omit the index G whenever G is clear from the context.

To prove our result on spectra we need the following lemma.

Lemma 4.1. Let $G = (\Sigma, P, \omega)$ be a rb grammar and let $u \in \Sigma^*$. There exist constants $n_0, q_0 \in \mathbf{N}$ satisfying:
if $n \in spec(u,\Lambda)$ with $n \geq n_0$, then $\{n + c_n \cdot k \mid k \in \mathbf{N}\} \subseteq spec(u,\Lambda)$ for some $c_n \in \mathbf{N}$ with $1 \leq c_n \leq q_0$.

Proof.

There are several ways to prove the lemma. One way would be to consider some "arithmetic" properties concerning the length of lines in the derivation and to obtain a pumping property for "long enough" derivations - such a proof is presented in [EHR4]. Here we will briefly sketch a "standard" tree-based proof of the lemma. (Since we use standard reasoning we will present the ideas rather informally.)

A derivation $\delta : u \overset{*}{\underset{G}{\Longrightarrow}} \Lambda$ can be represented by a forest $T_1, \cdots, T_{|u|}$ of node-labelled trees: leaves are labelled by Λ, all other nodes are labelled by letters from Σ (if $|u| = 1$, then we deal with a single tree rather than a forest, otherwise each tree of the forest $T_1, \ldots, T_{|u|}$ represents a derivation $\delta_i : u(i) \overset{*}{\underset{G}{\Longrightarrow}} \Lambda$).

For example if $\delta = (AB, ABA, ABAB, ABA, AB, A, AB, A, AB, A, \Lambda)$ is a derivation in a rb grammar G, then it is represented by the following forest.

Figure 10.

Given such a representation one can recover the *unique* derivation it represents (because we deal with right-most rewriting).

The number of internal nodes in T_i equals $lg(\delta_i)$. If no path in T_i is longer than k (counting the number of edges), then T_i has at most $1 + m + m^2 + \cdots + m^{k-1} = \frac{m^k - 1}{m - 1}$ internal nodes, where $m = maxr(G)$. Hence if $lg(\delta) = lg(\delta_1) + \cdots + lg(\delta_{|u|})$ is larger than $|u| \cdot \frac{m^{\#\Sigma+1}}{m-1}$, then at least one of the trees, say T_t, contains a path π that has at least two nodes with the same label. Let $T^{(0)}$ and $T^{(1)}$ be subtrees of T_t that have their roots in nodes on π with identical labels such that $T^{(0)}$ is a subtree of $T^{(1)}$ and such that $T^{(1)}$ is minimally chosen. Manipulating these subtrees in the standard way within T_t it is possible to obtain for each $k \in \mathbf{N}$ a tree $T_t^{(k)}$ representing a derivation $\delta_t^{(k)} : u(t) \overset{*}{\underset{G}{\Longrightarrow}} \Lambda$ with $lg(\delta_t^{(k)}) = lg(\delta_t) + c \cdot k$, where c equals the number of internal nodes in $T^{(1)}$ minus the number of internal nodes in $T^{(0)}$. Due to the minimality of $T^{(1)}$ no path in this tree is longer than $\#\Sigma + 1$, thus $c < \frac{m^{\#\Sigma+1}}{m-1}$. Combining $T_t^{(k)}$ with the trees $T_1, \ldots, T_{t-1}, T_{t+1}, \ldots, T_{|u|}$ again we have the tree representation of a derivation $\delta^{(k)} : u \overset{*}{\underset{G}{\Longrightarrow}} \Lambda$ with $lg(\delta^{(k)}) = lg(\delta) + c \cdot k$. Hence the lemma holds. ∎

Corollary 4.2. Let $G = (\Sigma, P, \omega)$ be a rb grammar. For each $u \in \Sigma^*$, $spec(u, \Lambda)$ is an ultimately periodic set.

Proof. Let $u \in \Sigma^*$ and let n_0, q_0 be constants satisfying the statement of Lemma 4.1. Let $\bar{c}$ be a common multiple of the numbers 1, 2, ..., q_0. Thus, if $n \in spec(u, \Lambda)$ with $n \geq n_0$, then $\{n + \bar{c} \cdot k \mid k \in \mathbf{N}\} \subseteq$
$\{n + c_n \cdot k \mid k \in \mathbf{N}\} \subseteq spec(u, \Lambda)$ for some $c_n \in \mathbf{N}$ with $1 \leq c_n \leq q_0$.

Let $I = \{n \mid n \in spec(u, \Lambda)$ and $P = \{n \mid n \in spec(u, \Lambda), n \geq n_0$ and $n < n_0$ and $n - \bar{c} \notin spec(u, \Lambda)\}$.

Then I and P are finite sets (elements of P are all different modulo $\bar{c}$) and consequently $spec(u, \Lambda) = I \cup \{n + k \cdot \bar{c} \mid n \in P \text{ and } k \in \mathbf{N}\}$ is an ultimately periodic set. ■

Note that the tree oriented argument used in our proof of Lemma 4.1 works nicely because we consider derivations that derive Λ. If we consider derivations that derive a non-empty word, then a technical difficulty arises: replacing a sub-tree in a tree that corresponds to a derivation may lead to a tree that does not represent a derivation.

In this paper we will use Corollary 4.2. However the ultimate periodicity property holds in a more general sense than presented above. One can prove that (see [EHR4]):

Proposition 4.3. Let $G = (\Sigma, P, \omega)$ be a rb grammar. For all $u, v \in \Sigma^*$, $spec(u,v)$ is an ultimately periodic set. ■

Theorem 4.4 Let $G(\Sigma, P, \omega)$ be a rb grammar. Then $FR_n(G)$ is regular for each $n \in \mathbf{N}^+$.

Proof.

Let $n \in \mathbf{N}^+$. We construct a rl grammar $H_n = (\Theta_n, P_n, S, \Delta_n)$ as follows.

Let $\Sigma_n = \{\omega\} \cup \{x \mid x \in \Sigma^* \text{ and } 1 \le |x| \le n-1 + maxr(G)\}$. Besides letters from $\Sigma \cup \{\$\}$, H_n will have terminal symbols representing special suffixes of words in Σ_n

Let $\Delta_n = \Sigma \cup \{\$\} \cup \{<A,z> \mid$ there exists an $y \in \Sigma^n$ such that $y(n) = A$ and $yz \in \Sigma_n$.$\}$,

let $\Theta_n = \Delta_n \cup \{[x] \mid x \in \Sigma_n\}$ and let $S = [\omega]$.

For each $x \in \Sigma_n$, P_n contains the following productions.

(1) If $|x| < n$, then $[x] \rightarrow \$[y] \in P_n$ for each $y \in \Sigma^*$ such that $x \Longrightarrow y$ (for

simplicity we set $[\Lambda] = \Lambda$),

(2) if $|x| = n$, then $[x] \to A[y] \in P_n$, with $A = x(n)$, for each $y \in \Sigma^*$ such that $x \Longrightarrow y$, and

(3) if $|x| > n$, then $[x] \to \langle A,z\rangle[y] \in P_n$, where $A = x(n)$ and $y \in \Sigma^n$ with $x = yz$.

Finally let σ be the substitution on Δ_n defined by $\sigma(A) = \{A\}$, $\sigma(\$) = \$$ and $\sigma(\langle A,Z\rangle) = \{A^k \mid k \in spec(z,\Lambda)\}$. We have to prove that $FR_n(G) = \sigma(L(H_n))$.

Claim 1. $FR_n(G) \subseteq \sigma(L(H_n))$.

Proof of Claim 1.

Let $\delta : \omega \overset{*}{\Longrightarrow} \Lambda$ be a derivation in G. Clearly there exists a unique decomposition $\delta = \delta_1 \otimes \delta_2 \otimes \cdots \otimes \delta_r$ of δ such that for each $1 \le k \le r$

(i) either $|\delta_k(0)| \le n$ and $lg(\delta_k) = 1$,

(ii) or $|\delta_k(i)| > n$ for $0 \le i < lg(\delta_k)$ and $|\delta_k(lg(\delta_k))| = n$.

For each of the derivations δ_k there exists a production $\pi_k = [\delta_k(0)] \to u_k[\delta_k(lg(\delta_k))]$ in P_n such that $fr_n(\delta_k) \in \sigma(u_k)$. (Note that $\delta_k(lg(\delta_k)) = \delta_{k+1}(0)$ if $k < r$ and $\delta_r(lg(\delta_r)) = \Lambda$, remember our convention that $[\Lambda] = \Lambda$.) The existence of π_k is seen as follows.

(1) If $|\delta_k(0)| < n$, then $lg(\delta_k) = 1$ and $fr_n(\delta_k) = \$$. P_n contains the production $[\delta_k(0)] \to \$[\delta_k(1)]$ and $\sigma(\$) = \{\$\}$.

(2) If $|\delta_k(0)| = n$, then $lg(\delta_k) = 1$ and $fr_n(\delta_k) = A$, where $A = \delta_k(0)(n)$. P_n contains the production $[\delta_k(0)] \to A[\delta_k(1)]$; $\sigma(A) = \{A\}$.

(3) If $|\delta_k(0)| > n$ then all lines of δ_k except for the last one are longer than n. Thus all lines of δ_k have the prefix $\delta_{k+1}(0)$;$k \ne r$ because the last line of δ_k does not equal Λ. Hence we may write $\delta_k(0) = \delta_{k+1}(0)z$ for some $z \in \Sigma^*$ and moreover there exists a derivation $\mu : z \overset{*}{\Longrightarrow} \Lambda$ with $cont(\mu) = cont(\delta_k)$. Let $A = \delta_k(0)(n) = \delta_{k+1}(0)(n)$. Then P_n contains the production

$[\delta_k(0)] \to <A,z>[\delta_{k+1}(0)]$ and $fr_n(\delta_k) = A^{lg(\delta_k)} = A^{lg(\mu)} \in \sigma(<A,z>)$.

It is now obvious that $(\pi_1, \ldots, \pi_r)$ can be used as the control sequence of a derivation $[\omega] \overset{*}{\Longrightarrow} u[\Lambda] = u$ in H_n with $u \in \Delta_n^*$ such that $fr_n(\delta) = fr_n(\delta_1) \cdot \cdots \cdot fr_n(\delta_r) \in \sigma(u)$. This proves the claim. ■

Claim 2. $\sigma(L(H_n)) \subseteq FR_n(G)$.

Proof of Claim 2.

For each $u \in L(H_n)$ and each $w \in \sigma(u)$ we can obtain a derivation $\delta : \omega \overset{+}{\Longrightarrow} \Lambda$ in G with $fr_n(\delta) = w$ by reversing the construction used in proving the previous claim. δ will be the composition of derivations $\delta_1, \ldots, \delta_r$ that are based on the control sequence $(\pi_1, \ldots, \pi_r)$ of a derivation of u in H_n. Note that, due to the form of productions in P_n, $r = |u|$.

Let $w = w_1 \cdots w_r$ with $w_k \in \sigma(u(k))$ for $k = 1, \ldots, r$.

(1,2) If $\pi_k = [x] \to A[y]$, where $|x| \le n$ and $A \in \Sigma \cup \{\$\}$, then $\delta_k = (x,y)$.

(3) If $\pi_k = [x] \to <A,z>[y]$ with $|x| > n$, $A = x(n)$ and $x = yz$, then $w_k = A^p$ for some $p \in spec(z,\Lambda)$. Choose any derivation $\mu : z \overset{*}{\underset{G}{\Longrightarrow}} \Lambda$ with $lg(\mu) = p$ and let $\delta_k : x = yz \overset{*}{\underset{G}{\Longrightarrow}} y$ be the derivation such that $cont(\delta_k) = cont(\mu)$.

As in the proof of Claim 1 it is easily seen that $\delta_1 \otimes \delta_2 \otimes \cdots \otimes \delta_r$ is a well-defined derivation $\delta : \omega \overset{*}{\Longrightarrow} \Lambda$ in G with

$fr_n(\delta) = fr(\delta_1) \cdot \cdots \cdot fr_n(\delta_r) = w_1 \cdots w_r = w$. Hence the claim holds. ■

From the two claims above we indeed get that $FR_n(G) = \sigma(L(H_n))$.

Note that σ is a regular substitution because the sets $\{A^k \mid k \in spec(z,\Lambda)\}$ are regular by Corollary 4.2. Hence, since $L(H_n)$ is regular and the class of regular languages is closed under regular substitution, $FR_n(G)$ is regular. ■

We end this section by demonstrating that, unlike in the case of active records, an arbitrary union of $FR_n(G)$ languages do not have to be regular. As a matter of fact we exhibit a rb grammar G for which $\bigcup_{n \in \mathbf{N}^+} FR_n(G)$ is not regular.

Theorem 4.5. There exists a rb grammar G such that $\bigcup_{n=1}^{\infty} FR_n(G)$ is not regular.

Proof.

Let $G = (\{A,B\}, P, A)$ be the rb grammar such that $P = \{A \to BA, A \to \Lambda, B \to \Lambda\}$.

All derivations from A to Λ are of the form
$\delta_k = (A, BA, B^2A, \ldots, B^kA, B^k, B^{k-1}, \ldots, B, \Lambda)$
for some $k \in \mathbf{N}$; then obviously $lg(\delta_k) = 2k+1$ and $brd(\delta_k) = k+1$.
Thus $fr_n(\delta_k) = \$^{2k+1}$ for each $n > k+1$.
For all $n \leq k+1$ we have $fr_n(\delta_k) = \$^{n-1}AB^{2(k+1-n)}\$^{n-1}$.

Consequently $\bigcup_{n=1}^{\infty} FR_n(G) \cap \$^*A\$^* = \{\$^nA\$^n \mid n \in \mathbf{N}\}$.

Since $\{\$^nA\$^n \mid n \in \mathbf{N}\}$ is not regular and the class of regular languages is closed under intersections, $\bigcup_{n=1}^{\infty} FR_n(G)$ is not regular. ■

Remark 4.2. (1) If (for the rb grammar from the proof of the previous theorem) we consider *arbitrary* unions $\bigcup_{n \in M} FR_n(G)$ for $M \subseteq \mathbf{N}^+$, then we can get even nonrecursive languages (by taking M nonrecursive):

$\bigcup_{n \in M} FR_n(G) \cap \$^*A\$^* = \{fr_n(\delta_k) \mid n \in M, k = n-1\} = \{\$^{n-1}A\$^{n-1} \mid n \in M\}$.

(2) In general, given a rb grammar G, $\bigcup_{n=1}^{\infty} FR_n(G)$ does not have to be even

context-free. An example of such a situation is the rb grammar given in Example 4.1. We have then $fr_n(\delta_{k,l}) \in \$^* C\$^* C\$^*$ if and only if $n = k+2 = l+2$.

Since $fr_{k+2}(\delta_{k,k}) = \$^k C\$^{2k+1} C\$^{k+1}$, we have

$$\bigcup_{n=1}^{\infty} FR_n(G) \cap \$^* C\$^* C\$^* = \{\$^k C\$^{2k+1} C\$^{k+1} \mid k \in \mathbf{N}\}$$

which is not a context-free language. Thus, because the class of context-free languages is closed under intersections with regular languages, $\bigcup_{n=1}^{\infty} FR_n(G)$ is not context-free. ∎

5. COORDINATED PAIR SYSTEMS

Right boundary grammars form a very basic building block in the general theory of grammars and automata presented in [R]. In particular, within this theory a push-down automaton is seen as a pair of "cooperating grammars", the first one rl (modelling the input and the finite state control) and the other one rb (modelling the infinite push-down store); such a pair is called a coordinated pair system.

In this section we will "transfer" our results concerning the (active and full records of the) use of memory in rb grammars to the level of cp systems (where the work of the rb component is coordinated by the right linear component). In this way investigating the use of memory in rb grammars is being used for learning about the use of memory in push-down automata.

We begin by recalling the notion of a coordinated pair system.

Definition 5.1. A *coordinated pair system, cp system* for short, is triple $G = (G_1, G_2, R)$, where
$G_1 = (\Sigma_1, P_1, S_1, \Delta)$ is a rl grammar,
$G_2 = (\Sigma_2, P_2, S_2)$ is a rb grammar with $S_2 \in \Sigma_2$ and
$R \subseteq P_1 \times P_2$, the set of *rewrites* of G. ▪

Definition 5.2. Let $G = (G_1, G_2, R)$ be a cp system, where
$G_1 = (\Sigma_1, P_1, S_1, \Delta)$ and $G_2 = (\Sigma_2, P_2, S_2)$.

(1) Let $x_1, y_1 \in \Sigma_1^*$ and $x_2, y_2 \in \Sigma_2^*$. If $x_1 \overset{\pi_1}{\underset{G_1}{\Longrightarrow}} y_1$ and $x_2 \overset{\pi_2}{\underset{G_2}{\Longrightarrow}} y_2$ for a rewrite $\pi = (\pi_1, \pi_2) \in R$, then we say that (x_1, x_2) *directly computes* (y_1, y_2) *in* G *using* π and we denote this by $(x_1, x_2) \overset{\pi}{\underset{G}{\Longrightarrow}} (y_1, y_2)$.

$\overset{*}{\underset{G}{\Longrightarrow}}$ denotes the reflexive and transitive closure of $\underset{G}{\Longrightarrow}$. If $(x_1, x_2) \overset{*}{\underset{G}{\Longrightarrow}} (y_1, y_2)$, then we say that (x_1, x_2) *computes* (y_1, y_2) (*in* G).

(2) A *computation (in G)* is a sequence $\rho = (x_0, x_1, \ldots, x_n)$, $n \geq 0$, of elements from $\Sigma_1^* \times \Sigma_2^*$ such that $x_{i-1} \underset{G}{\Longrightarrow} x_i$ for each $1 \leq i \leq n$. We say that ρ *computes* x_n *from* x_0 and denote this by $\rho : x_0 \overset{*}{\Longrightarrow} x_n$.
n is called the *length of* ρ and is denoted by $lg(\rho)$. For $0 \leq i \leq n$ we use $\rho(i)$ to denote x_i. If $x_0 = (S_1, S_2)$ and $x_n = (w, \Lambda)$ for some $w \in \Delta^*$, then ρ is called *successful*.

(3) The *language of* G, denoted $L(G)$, is the set $\{w \in \Delta^* \mid (S_1, S_2) \overset{*}{\Longrightarrow} (w, \Lambda)\}$. ■

The formal notions describing the use of memory in rb grammars are extended to cp systems in an obvious way.

Definition 5.3. Let G be a cp system and let $n \in \mathbf{N}^+$.

(1) Let $\rho = (\rho(0), \rho(1), \ldots, \rho(k))$ be a computation in G. The *n-active record* of δ, denoted $act_n(\rho)$, is the word $\varphi_n(\rho(0))\varphi_n(\rho(1)) \cdots \varphi_n(\rho(k-1))$, where $\varphi_n : \Sigma_1^* \times \Sigma_2^* \to \Sigma_2 \cup \{\Lambda\}$ is the mapping defined by

$$\varphi_n((u,v)) = \begin{cases} v(n), & \text{if} |v| = n, \\ \Lambda & \text{, otherwise.} \end{cases}$$

(2) The *n-active language* of G, denoted $ACT_n(G)$, is the language $\{act_n(\rho) \mid \rho : (S_1, S_2) \Longrightarrow (w, \Lambda) \text{ for some } w \in \Delta^*\}$. ■

Definition 5.4. Let G be a cp system and let $n \in \mathbf{N}^+$.

(1) Let $\rho = (\rho(0), \rho(1), \ldots, \rho(k))$ be a computation in G. The *n-full record* of δ, denoted $fr_n(\rho)$, is the word $\psi_n(\rho(0))\ \psi_n(\rho(1)) \cdots \psi_n(\rho(k-1))$, where

$\psi_n : \Sigma_1^* \times \Sigma_2^* \to \Sigma_2 \cup \{\$\}$ is the mapping defined by

$$\psi_n((u,v)) = \begin{cases} v(n), & \text{if } |v| \leq n, \\ \$ & \text{, otherwise.} \end{cases}$$

(2) The *n-full record language* of G, denoted $FR_n(G)$, is the language $\{fr_n(\rho) \mid \rho : (S_1, S_2) \Longrightarrow (w,\Lambda) \text{ for some } w \in \Delta^*\}$. ■

Since in this paper we are mainly interested in the behavior of the second component of successful computations in a cp system we introduce the notion of an *internal* cp system. A cp system is internal if it has only chain-rules (i.e. of the form $A \rightarrow B$) and Λ-rules (i.e. of the form $A \rightarrow \Lambda$) on its first component. If, for a give cp system G, we erase all terminal symbols in all productions of the first component (and in the corresponding rewrites), then we obtain the *internal version* of G.

Definition 5.5. Let $G = (G_1, G_2, R)$ be a cp system with $G_1 = (\Sigma_1, P_1, S_1, \Delta)$.

(1) G is called *internal* if $\Delta = \emptyset$.

(2) The *internal version of G*, denoted $int(G)$, is the cp system $\tilde{G} = (\tilde{G}_1, G_2, \tilde{R})$ where $\tilde{G}_1 = (\Sigma_1 - \Delta, \tilde{P}_1, S_1, \emptyset)$ with

$\tilde{P} = \{(X \rightarrow Z) \mid (X \rightarrow wZ) \in P$ for some $X \in \Sigma_1 - \Delta$, $w \in \Delta^*$ and $Z \in (\Sigma_1 - \Delta) \cup \{\Lambda\}\}$

and $\tilde{R} = \{(X \rightarrow Z, \pi_2) \mid (X \rightarrow wZ, \pi_2) \in R$ for some $X \in \Sigma_1 - \Delta$, $w \in \Delta^*$ and $Z \in (\Sigma_1 - \Delta) \cup \{\Lambda\}\}$. ■

Hence $int(G)$ works precisely as G does except that it ignores the "input aspect" of G. Consequently as far as the use of memory is concerned one can consider $int(G)$ rather than G.

Lemma 5.1. For each cp system G and each $n \in \mathbf{N}^+$,

$ACT_n(G) = ACT_n(int(G))$ and

$FR_n(G) = FR_n(int(G))$. ■

6. CP SYSTEMS AND RB GRAMMARS

In the previous section we have reduced the considerations concerning (the use of memory in) cp systems to the considerations concerning internal cp systems. In this section we will further reduce the problem: we show that rather than to consider internal cp systems it suffices to consider rb grammars. As a matter of fact we demonstrate how to represent (successful computations in) an internal cp system by (successful derivations in) a suitable rb grammar.
This "representation theorem" yields then the regularity of the (active and full) use of memory in arbitrary cp systems.

Definition 6.1. Let Θ, Σ_1 and Σ_2 be alphabets and let $\psi : \Theta^* \to \Sigma_1^*$ and $\varphi : \Theta^* \to \Sigma_2$ be codings. Let $(Z, w) \in (\Sigma_1 \cup \{\Lambda\}) \times \Sigma_2^*$ and $u \in \Theta^*$. We say that u (ψ, φ)*–represents* (Z,w), denoted $u[\psi,\varphi>(Z,w)$, if the following holds.

(i) $Z = \Lambda$ if $u = \Lambda$ and $Z = \psi(last(u))$ otherwise.

(ii) $w = \varphi(u)$. ▪

Definition 6.2. Let $G = (G_1, G_2, R)$ be an internal cp system with $G_1 = (\Sigma_1, P_1, S_1, \emptyset)$, $G_2 = (\Sigma_2, P_2, S_2)$ and let $H = (\Theta, Q, T)$, $T \in \Theta$, be a rb grammar. Finally let $\psi : \Theta^* \to \Sigma_1^*$ and $\varphi : \Theta^* \to \Sigma_2^*$ be codings.

(1) Let ρ be a computation in G and let δ be a derivation in H. We say that $\delta(\psi,\varphi)$*–represents* ρ, denoted $\delta[\psi,\varphi>\rho$, if $lg(\delta) = lg(\rho)$ and $\delta(j)[\psi,\varphi>\rho(j)$ for each $0 \leq j \leq lg(\delta)$.

(2) G is (ψ,φ)*–represented by* H if the following holds.

(i) For each successful computation ρ in G there exists a successful derivation δ in H such that $\delta[\psi,\varphi>\rho$.

(ii) For each successful derivation δ in H there exists a successful computation ρ in G such that $\delta[\psi,\varphi>\rho$. ▪

Theorem 6.1. For every internal cp system G there exist codings ψ and φ and a rb grammar H such that G is (ψ,φ)-represented by H.

Proof.

Let G be an internal cp system with $G = (G_1, G_2, R)$, $G_1 = (\Sigma_1, P_1, S_1, \emptyset)$ and $G_2 = (\Sigma_2, ,P_2, S_2)$.
(The construction of a rb system H somewhat resembles the usual construction of a context-free grammar for a given push-down automaton.)

Let $H = (\Theta, Q, T)$ be defined as follows
$\Theta = \{[Y,A,Z] \mid A \in \Sigma_2, Y \in \Sigma_1 \cup \{\Lambda\}$ and $Z \in \Sigma_1\}$,
$T = [\Lambda, S_2, S_1]$ and
Q contains the following productions
$[Y,A,X] \to \Lambda$ if and only if $(X \to Y, A \to \Lambda) \in R$ for $A \in \Sigma_2$, $X \in \Sigma_1$, $Y \in \Sigma_1 \cup \{\Lambda\}$,
$[Y,A,X] \to [Y,B,Z]$ if and only if $(X \to Z, A \to B) \in R$ for
$A,B \in \Sigma_2$, $X, Z \in \Sigma_1$, $Y \in \Sigma_1 \cup \{\Lambda\}$, and
$[Y,A,X] \to [Y,B_1,Z_1][Z_1, B_2, Z_2] \cdots [Z_{r-1}, B_r, Z]$ if and only if
$(X \to Z, A \to B_1, B_2 \cdots B_r) \in R$ for $r \geq 2$,
$A, B_1, \ldots, B_r \in \Sigma_2$, $X, Y, Z, Z_1, \ldots, Z_{r-1} \in \Sigma_1 \cup \{\Lambda\}$.

The codings $\psi : \Theta^* \to \Sigma_1^*$ and $\varphi : \Theta^* \to \Sigma_1^*$ are given by $\psi([Y,A,Z]) = Z$ and $\varphi([Y,A,Z]) = A$ for $Y,Z \in \Sigma_1 \cup \{\Lambda\}$ and $A \in \Sigma_2$.

Claim 1. For every successful computation $\rho : (S_1, S_2) \overset{*}{\Longrightarrow} (\Lambda, \Lambda)$ in G there exists a successful derivation $\delta : T \overset{*}{\Longrightarrow} \Lambda$ in H such that $\delta[\psi, \varphi>\rho$.

Proof of Claim 1.

Let ρ be a successful derivation in G. For $0 \leq i \leq lg(\rho)$ we write $\rho(i) = (Z_i, w_i)$.
We will construct a successful derivation δ in H that (ψ, φ)-represents ρ as follows. First let $\delta(0) = T$. Then $\delta(0)[\psi,\varphi>(S_1, S_2) = \rho(0)$.

We proceed inductively. Assume that $\delta(p), p < lg(\rho)$, is already constructed. From the form of the productions in H it easily follows that $\delta(p)$ is of the form $[\Lambda, A_1, Y_1][Y_1, A_2, Y_2] \cdots [Y_{l-1}, A_l, Y_l]$ for some $l \geq 1$; thus $Z_j = \psi([Y_{l-1}, A_l, Y_l]) = Y_l$ and $w_j = \varphi(\delta(p)) = A_1 A_2 \cdots A_l$. Let π be the rewrite such that $\rho(p) \overset{\pi}{\underset{G}{\Longrightarrow}} \rho(p+1)$. We consider separately three cases.

(1) $\pi = (X \to Z, A \to B_1 B_2 \cdots B_r)$ for some $r \geq 1$, $Z \neq \Lambda$.

Obviously $X = Y_l$ and $A = A_l$. We use now a production $[Y_{l-1}, A_l, Y_l] \to [Y_{l-1}, B_1, W_1][W_1, B_2, W_2] \cdots [W_{r-1}, B_r, Z]$ to derive $\delta(p+1)$ from $\delta(p)$.

The variables $W_1, \ldots, W_{r-1}$ are determined as follows. For $i \in \{1, 2, \ldots, r-1\}$ let $\rho(s_i)$ be the element of the computation ρ in which the occurrence B_i "becomes active" on the second component; more precisely, s_i is the least integer larger than p such that $w_{s_i} = A_1 A_2 \cdots A_{l-1} B_1 \cdots B_i$ (thus $s_r = p+1$).

We then choose $W_i = Z_{s_i}$. Note that $W_i = \Lambda$ if $s_i = lg(\rho)$ - this is the case if $i = lg(\rho)-1$.

(2) $\pi = (X \to Z, A \to \Lambda)$, where $Z \neq \Lambda$.

Obviously $X = Y_l$ and $A = A_l$. Since ρ is a successful computation, $l \neq 1$. The occurrence A_{l-1} will be "active" in the next step of the computation ρ and, since $[Y_{l-2}, A_{l-1}, Y_{l-1}]$ was introduced by a production as under (1) above, $Y_{l-1} = Z$. So we can apply the production $[Y_{l-1}, A_l, Y_l] \to \Lambda$ to derive $\delta(p+1)$ from $\delta(p)$.

(3) $\pi(X \to \Lambda, A \to \Lambda)$.

Note that this rewrite can be applied in the last step of a computation only. Thus $l = 1$, $X = Y_1$ and $A = A_1$. So we can derive $(\Lambda,\Lambda) = \delta(p+1)$ from $\delta(p)$ using the production $[\Lambda, A_1, Y_1] \to \Lambda$.

It is easily seen that in all the three cases above the requirement $\delta(p+1)[\psi,\varphi>\rho(p+1)$ is satisfied. Hence the claim holds. ∎

Claim 2. For every successful derivation $\delta : T \overset{*}{\Longrightarrow} \Lambda$ in H there exists a successful computation $\rho : (S_1\,,\, S_2) \overset{*}{\Longrightarrow} (\Lambda, \Lambda)$ in G such that $\delta[\psi,\varphi{>}\rho$.

Proof of Claim 2.

The proof of this claim is rather obvious. Let δ be a successful derivation in H. For $0 \le i < lg(\delta)$ we define $\rho(i) = (\psi(last(\delta(i))),\ \varphi(\delta(i)))$ and
$\delta(lg(\delta)) = (\Lambda, \Lambda)$.
It can be easily checked that the so defined sequence ρ is a successful computation in G. ■

The theorem follows from the above two claims. ■

Remark 6.1. The above result allows one to replace a (internal) cp system by a rb grammar as for as *successful* computation are concerned. In [EHR3] a more general representation theorem is proved where both successful and not successful computations are considered. Then to represent a cp system one needs (in general) a finite number of rb grammars. ■

Let $\varphi : \Theta^* \to \Sigma^*$ be a coding and let $\$ \notin \Theta$. By $\varphi_{\$}$ we denote the coding of $(\Theta \cup \{\$\})^*$ into $(\Sigma \cup \{\$\})^*$ defined by $\varphi_{\$}(A) = \varphi(A)$ for every $A \in \Theta$ and $\varphi_{\$}(\$) = \$$. Using this notation we can express the relation between the record languages of a cp system G and a rb grammar representing $int(G)$ as follows.

Lemma 6.2. Let G be a cp system, let ψ and φ be codings and let H be a rb grammar such that $int(G)$ is $(\psi,\ \varphi)$-represented by H.
For every $n \in \mathbf{N}$, $ACT_n(G) = \varphi(ACT_n(H))$ and $FR_n(G) = \varphi_{\$}(FR_n(H))$.

Proof.

If ρ is a successful computation in $int(G)$ and δ is a successful derivation in H such that $\delta[\psi,\varphi{>}\rho$, then $act_n(\rho) = \varphi(act_n(\delta))$ for $n \in \mathbf{N}^+$, because φ is a coding.

Hence $ACT_n(int(G)) = \varphi(ACT_n(H))$ and consequently, by Lemma 5.1, $ACT_n(G) = \varphi(ACT_n(H))$ for each $n \in \mathbf{N}^+$.

Using the same arguments we see that

$FR_n(G) = FR_n(int(G)) = \varphi_{\$}(FR_n(H))$ for $n \in \mathbf{N}^+$. ∎

We can now carry our results concerning the use of memory in rb systems to corresponding results for cp systems.

Theorem 6.3. Let G be a cp system.

(1) For each $n \geq 1$, $ACT_n(G)$ is regular.

(2) For each $n \geq 1$, $FR_n(G)$ is regular.

Proof.

(1) Let, for a suitable pair of codings the internal cp system $int(G)$ be a (ψ, φ)-represented by a rb grammar H. Let $n \in \mathbf{N}^+$. Then according to the previous lemma $ACT_n(G) = \varphi(ACT_n(H))$. This language is regular because by Corollary 1.3 $ACT_n(H)$ is regular and the family of regular languages is closed under homomorphisms.

(2) Analogously it follows from the regularity of $FR_n(H)$ - Theorem 4.4 - that $FR_n(G) = \varphi_{\$}(FR_n(H))$ is a regular language. ∎

Theorem 6.4.

(1) For each cp system G and each $I \subseteq \mathbf{N}^+$, $\bigcup_{n \in I} ACT_n(G)$ is regular.

(2) There exists a cp system G such that $\bigcup_{n} \in \mathbf{N}\, FR_n(G)$ is not regular.

Proof.

(1) For a given cp system G let H be a rb grammar, let ψ and φ be codings such that $int(G)$ is (ψ, φ)-represented by H.

Then it follows from Theorem 3.4 that $\bigcup_{n \in I} ACT_n(G) = \bigcup_{n \in I} \varphi ACT_n(H) = \varphi(\bigcup_{n \in I} ACT_n(H))$ is regular for each $I \subseteq \mathbf{N}$.

(2) Every rb grammar $H = (\Sigma, P, A)$ with $A \in \Sigma$ can be transformed in a natural way into a cp system by "adding" a (dummy) first component with one nonterminal only. Formally, let $H_{cp} = (G_1, G_2, R)$ be the (internal) cp system with $G_1 = (\{S\}, P_1, S, \emptyset)$, $P_1 = \{S_1 \to S_1, S_1 \to \Lambda\}$, $G_2 = H$ and $R = P_1 \times P$. One easily verifies that H_{cp} is (ψ,φ)-represented by H, where $\psi : \Sigma^* \to \{S\}^*$ is the coding that maps each element of Σ to S and $\varphi : \Sigma^* \to \Sigma^*$ is the identity on Σ^*. Thus, for $n \in \mathbf{N}^+$, $FR_n(H_{cp}) = \varphi_{\$}(FR_n(H)) = FR_n(H)$.

If we construct in this way G_{cp} for the rb grammar G given in the proof of Theorem 4.5, then $\bigcup_{n=1}^{\infty} FR_n(G_{cp}) = \bigcup_{n=1}^{\infty} FR_n(G)$, which is not regular. ■

Remark 6.1. The results concerning rb grammars mentioned in Remark 4.2 carry over to cp systems using the construction described in the proof of Theorem 6.4.(2). Thus in general, given a cp system G, $\bigcup_{n=1}^{\infty} FR_n(G)$ does not have to be context-free. Moreover if we take arbitrary unions $\bigcup_{n \in M} FR_n(G)$ for $M \subseteq \mathbf{N}^+$, then we can get arbitrarily complex languages. ■

DISCUSSION

Within the framework of ects systems (see [R]) rb grammars form a basic building block in constructing various types of grammars and machines known from the literature. Hence there is a need for a fundamental research concerning rb grammars.

In this paper we have investigated the use of memory in rb grammars. We have chosen two specific ways of "tracing" the use of memory in rb grammars: either we record the sequence of active use of a particular memory cell during a derivation or we record the contents of the cell during all steps of a derivation. Then it turns out that *all* active records for a given cell for *all successful* derivations form a regular language (Theorem 1.2) and *all* full records for a given cell for *all* successful derivations form a regular language (Theorem 4.4). As a matter of fact active record languages have a very specific structure which makes the "overall active use of memory" in a rb grammar regular: the union over any set of memory cells of all active records for all successful computations is regular (Theorem 3.4)!!! This is not the case for full record languages: the union over an arbitrary set of memory cells of all full records for all successful derivations can be arbitrarily complex.

Cp systems form a subclass of ects systems that correspond very closely to (are another formulation of) push-down automata. A cp system is a "coordinated pair" of a right-linear grammar and a rb grammar: in this combination the rb grammar component represents the (infinite) memory structure of the system. From this point of view investigating the use of memory in rb systems is very natural (and very much needed).

In order to transfer our results on the active use of memory in rb grammars to the level of cp systems (where the work of the rb component is coordinated by the right-linear component) we prove a representation theorem

(Theorem 5.1) for cp systems which allows one (in the investigation of computations in cp systems) to consider a rb grammar rather than a cp system.

We believe that this paper illustrates the usefulness of the fundamental research concerning rb grammars and of the cp systems point of view at push-down automata. It seems to be easier (and more elegant) to prove basic results on the level of rb grammars and then transfer them to the level of cp systems (by a "once and forever" established representation theorem) rather than to prove the corresponding results directly for cp systems.

We consider this paper as a first step into the systematic investigation of the use of memory in rb grammars and cp systems. Clearly one can consider other than active and full ways of recording the use of a memory in rb grammars. How complicated are other types of "recording the memory" languages? Are they regular? Are their arbitrary unions regular?

We are presently working on a number of problems of this nature and hope to present the results of our research in a forthcoming paper.

ACKNOWLEDGEMENTS

The first and the third author gratefully acknowledge the support of NSF grant MCS 83-05245.

REFERENCES

[B] Buchi, J.R., "Regular canonical systems", *Arch. Math. Logik und Grundlagenforsch.*, v.6, pp. 91-111, 1964.

[C] Conway, J.H., *Regular Algebra and Finite Machines*, Chapman and Hall, London, 1971.

[ER] Ehrenfeucht, A. and Rozenberg, G., "A note on regular canonical systems," Techn. Rep. No. CU-CS-279-84, Dept. of Comp. Science, Univ. of Colorado at Boulder, 1984.

[EHR1] Ehrenfeucht, A., Hoogeboom, H.J. and Rozenberg, G., "Computations in coordinated pair systems", Techn. Rep. No. CU-CS-260-84, Department of Computer Science, University of Colorado at Boulder, 1984.

[EHR2] Ehrenfeucht, A., Hoogeboom, H.J. and Rozenberg, G., "Coordinated pair systems; Part 1: Dyck words and classical pumping", Techn. Rep. No. CU-CS-275-84, Department of Computer Science, University of Colorado, Boulder, 1984.

[EHR3] Ehrenfeucht, A., Hoogeboom, H.J. and Rozenberg, G., "On the active use of memory in right-boundary grammars and push-down automata", Techn. Rep. No. CU-CS-298-85, Dept. of Comp. Science, Univ. of Colorado at Boulder, 1985.

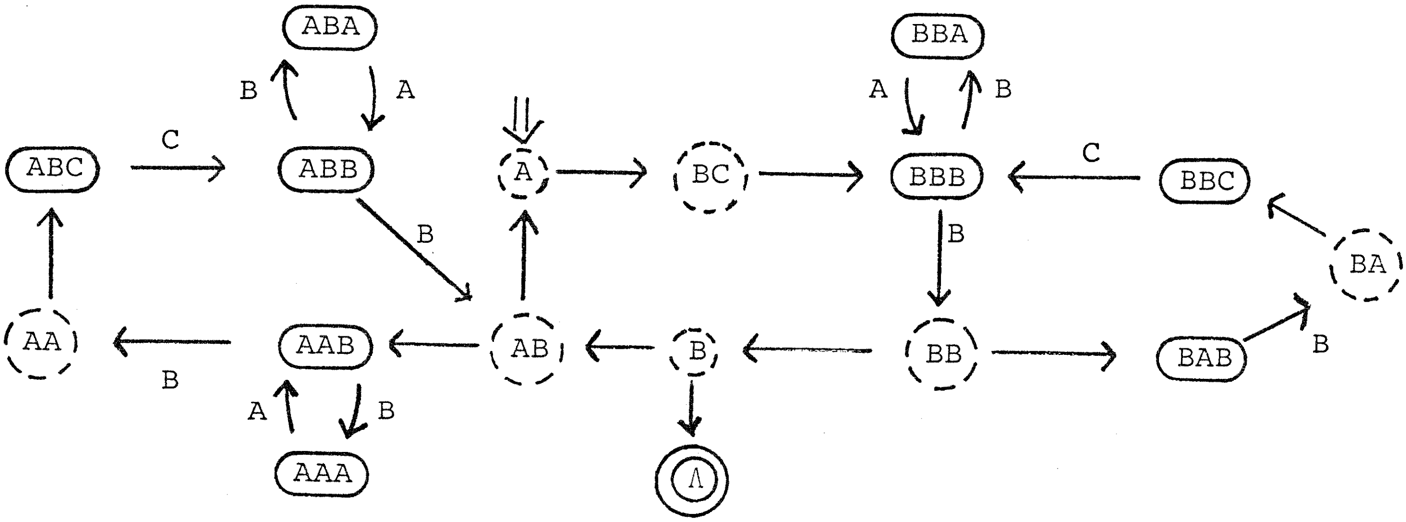
[EHR4] Ehrenfeucht, A., Hoogeboom, H.J. and Rozenberg, G., "On the full use of memory in right-boundary grammars and push-down automata", Techn. Rep. No. CU-CS-299-85, Dept. of Comp. Science, Univ. of Colorado at Boulder, 1985.

[H] Harrison, M., *Introduction to formal language theory*, Addison-Wesley, Reading, Massachusetts, 1978.

[R] Rozenberg, G., "On coordinated selective substitutions: Towards a unified theory of grammars and machines", *Theoretical Computer Science*, Vol.37, 1985.

[S1] Salomaa, A., *Theory of automata*, Pergamon Press, Oxford, 1969.

[S2] Salomaa, A., *Formal languages*, Academic Press, London-New York, 1973.



ABA
B
A
ABC
C
ABB
A
BC
BBA
A
B
BBB
C
BBC
B
B
BA
AA
B
AAB
AB
B
BB
BAB
B
A
B
AAA
Λ

Fig. 3

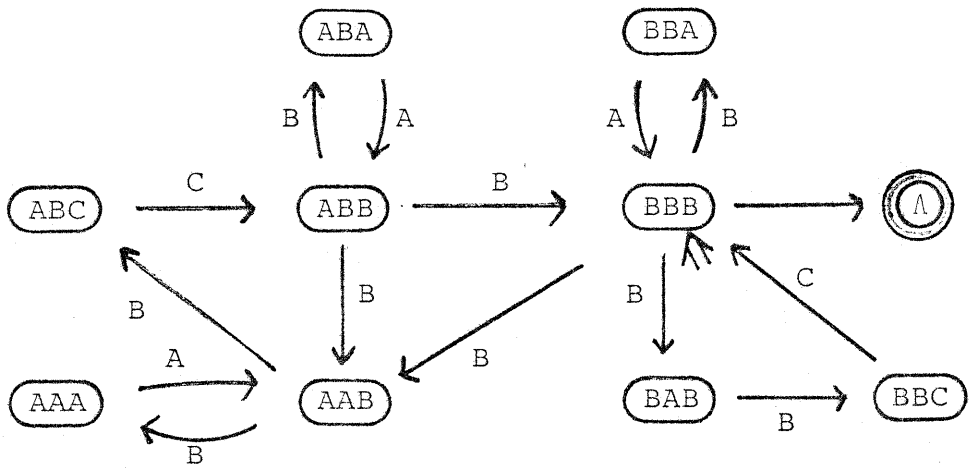


Figure 4.

Figure 10.