Hierarchy Results on Stateless Multicounter $5' \rightarrow 3'$ Watson-Crick Automata

Benedek Nagy[1], László Hegedüs[1], and Ömer Eğecioğlu[2]

[1] Department of Computer Science, Faculty of Informatics,
University of Debrecen, Debrecen, 4032 Hungary
nbenedek@inf.unideb.hu, hegedus.laszlo@inbox.com

[2] Department of Computer Science, University of California,
Santa Barbara, CA 93106, USA
omer@cs.ucsb.edu

Abstract. We consider stateless counter machines which mix the features of one-head counter machines and a special type of two-head Watson-Crick automata (WK-automata). Our biologically motivated machines have heads that read the input starting from the two extremes. The reading process is finished when there are no more symbols between the heads. Depending on whether the heads are required to advance at each move, we distinguish between realtime and non-realtime machines. If every counter makes at most k alternations between nondecreasing and decreasing modes in every computation, then the machine is k-reversal. In this paper we concentrate on the properties of nondeterministic stateless WK-automata with counters. Results about deterministic versions can be found in (Eğecioğlu et al.: Stateless multicounter $5' \rightarrow 3'$ Watson-Crick Automata, BIC-TA 2010).

1 Introduction

A well investigated branch of DNA computing is the theory of Watson-Crick automata ([3,13]). These are finite state machines equipped with two read-only heads. They operate on DNA molecules, i.e., on double stranded sequences of bases. The strands of a DNA molecule have directions as a result of the underlying chemical bonds, determining the $5'$ and $3'$ ends of a strand. The two strands have opposite biochemical directions. Between the two strands, there is a one-to-one correspondence of the bases given by the so-called Watson-Crick complementarity relation. In this way a strand of the molecule uniquely defines the other, and therefore the DNA molecules can be described by ordinary strings (as, for instance, in [6,7]). In biology several enzymes are known to act on a DNA strand in direction from $5'$ to $3'$. Consequently, in the case of $5' \rightarrow 3'$ Watson-Crick automata, at the beginning of a computation, the reading heads start from opposite ends of the input and they can move opposite direction in computational point of view (but the same direction biochemically). These automata have been used to characterize linear context-free languages in [10]. In this paper we consider only $5' \rightarrow 3'$ Watson-Crick automata, and consequently use the terminology *WK-automata* and omit the qualification $5' \rightarrow 3'$.

J. Cabestany, I. Rojas, and G. Joya (Eds.): IWANN 2011, Part I, LNCS 6691, pp. 465–472, 2011.

Stateless machines (i.e. machines with only one state) have recently been connected to certain aspects of membrane computing [11,12]. As the name shows, a stateless machine (without additional storage) cannot store any information. Thus other methods are used, e.g., the automaton is equipped with some number of counters which are zero at the beginning and again zero at the end of a computation. In [5,14] the computing power of stateless multihead automata with respect to decision problems and head hierarchies were investigated.

If the machine is not allowed to make transitions without moving a head, then the model is called *realtime*. Otherwise the machine is *non-realtime* and can make transitions without moving the heads. We say that a counter is in increasing/decreasing mode if its value is increased/decreased at the last transition that changed its value. A counter machine is *k-reversal* if each counter makes at most k "full" alternations between increasing mode and decreasing mode and vice-versa on any computation (accepting or not), and is *reversal bounded* if it is k-reversal for some k.

Deterministic stateless (one-way) m-counter machines were investigated in [1], where hierarchies with respect to the number of counters and number of reversals were studied. Similar hierarchy results and characterizations are reported in [4] for the non-realtime versions. Hierarchies of the accepted language families by WK-automata are presented in [9] including stateless versions without counters. In this paper we concentrate on nondeterministic stateless realtime WK-automata with counters. Some results about the deterministic case can be found in [2]. We give examples and establish hierarchies of WK-automata with respect to counters and reversals.

2 Stateless Multicounter WK-Automata

We start with the description of the components of a stateless multicounter WK-automata. The input is of the form $\cent w\$$ with $w \in \Sigma^*$ and $\cent$ and $\$$ are endmarkers that are not in Σ. The machine has two read-only heads H_1, H_2. Head H_1 moves from left to right and H_2 moves from right to left. Originally, H_1 is on $\cent$ and H_2 is on $\$$. The machine is equipped with m counters, that are initially all zero. A move of the machine depends on the symbols under the heads and the signs of the counters (the automata can distinguish two cases: zero or positive). It consists of moving the heads and at the same time incrementing, decrementing, or leaving unchanged each counter. The input w is accepted by M if the counters are again zero when the heads *meet*.

The essence of when the heads H_1 and H_2 meet is captured best by making use of a function φ which indicates whether the heads are close or far apart in processing the input. This locality requirement can be justified in part by biological properties that give rise to WK-automata. For the model it suffices to know if there are zero, one, two, or more than two letters between the heads. Define

$$\varphi(M) = \begin{cases} p & \text{if there are } p \text{ letters between the two heads of } M \text{ and } p \leq 2, \\ \infty & \text{if there are more than two letters between the heads of } M. \end{cases}$$

We use the notation $\varphi(M)$ although φ is actually a function of the current positions of the heads of M, i.e., function of the actual configuration.

For a deterministic stateless multicounter WK-automaton M, a transition (a move)

$$((x, y; s_1, s_2, \ldots, s_m), p) \to (d_1, d_2; e_1, e_2, \ldots, e_m) \tag{1}$$

has the following parameters: $x, y \in \Sigma \cup \{¢, \$\}$ are the symbols under the heads H_1 and H_2, respectively; s_i is the sign of counter C_i: $s_i = 0$ if the i-th counter is zero, $s_i = 1$ if it is positive. $s_1 s_2 \cdots s_m$ is referred to as a *sign vector*; $d_1, d_2 \in \{0, 1\}$ indicate the direction of move of the heads with $d_1 + d_2 \leq p$. A value 0 signifies that the head stays where it is. $d_1 = 1$ means that H_1 moves one cell to the right, and $d_2 = 1$ means that H_2 moves one cell to the left; $e_i = +, -$, or 0, corresponding to the operations of increment, decrement, or leave unchanged the contents of the i-th counter. Here $e_i = -$ applicable only if $s_i = 1$. A move (1) is possible if and only if $\varphi(M) = p$. It should be noted that $\varphi(M)$ is not part of the system, nor it is a counter, just a technical parameter. M is *nondeterministic* if multiple choices are allowed for the right hand side of (1).

The machine is *realtime* if not both d_1 and d_2 are zero for any move of the machine. Otherwise it is *non-realtime.*

The machine is k-*reversal* if for a specified k, no counter makes more than k alternations between increasing mode and decreasing mode (i.e. k pairs of increase followed by decrease stages) in any computation, accepting or not. The machine is *reversal bounded* if it is k-reversal for some k.

We denote the set of all nondeterministic realtime k-reversal m counter non-realtime WK-automata by WKC_m^k, and the realtime versions by RWKC_m^k. The reversal bounded versions are given by

$$\mathrm{WKC}_m^* = \bigcup_{k=0}^{\infty} \mathrm{WKC}_m^k, \quad \mathrm{RWKC}_m^* = \bigcup_{k=0}^{\infty} \mathrm{RWKC}_m^k;$$

while WKC_m^∞ and RWKC_m^∞ denote the unbounded reversal versions. We use a "d" prefix to refer to the deterministic versions of these machines. So dRWKC_m^k denotes all deterministic realtime k-reversal m-counter machines. This notation is also used for the corresponding language classes.

The formal definition of a nondeterministic stateless multicounter WK-automaton is as follows.

Definition 1. *A nondeterministic stateless multicounter WK-automaton is a quadruple* $M = (\Sigma, \delta, ¢, \$)$ *where* Σ *is a nonempty alphabet,* δ *is a mapping from* $(\Sigma \cup \{¢, \$\})^2 \times \{1, 0\}^m \times \{0, 1, 2, \infty\}$ *to* $2^{(\{0,1\}^2 \times \{0,+,-\}^m)}$ *and* $¢, \$ \notin \Sigma$ *are two special symbols called endmarkers.*

3 Closure Properties of Nondeterministic Stateless Multicounter WK-Automata Languages

Some closure properties of languages accepted by nondeterministic stateless multicounter WK-automata follow.

Proposition 1. *The language family accepted by nondeterministic stateless realtime multicounter WK-automata is closed under the union operation.*

Proof. If M_1 is a k_1-reversal m_1-counter machine and M_2 is a k_2-reversal m_2-counter machine, then we can construct a $\max\{k_1,k_2\}$-reversal, $\max\{m_1+m_2\}+2$-counter machine which can simulate both M_1 and M_2. The two additional counters keep track of which machine is being simulated. ☐

Proposition 2. *Language families* $\mathrm{WKC}_m^x, \mathrm{RWKC}_m^x, \mathrm{dWKC}_m^x, \mathrm{dRWKC}_m^x$ *(*$m \in \mathbb{N}$*,* $x \in \mathbb{N} \cup \{*, \infty\}$*) are closed under reversal (mirror image) of words.*

Proof. Suppose a machine accepts w with k-counters and m-reversals. Then w^R can also be accepted with the same parameters, just the behaviour of the heads have to be exchanged. ☐

Proposition 3. *For* $|\Sigma| \geq 2$*, language families accepted by nondeterministic stateless realtime multicounter WK-automata are not closed under concatenation operation.*

Proof. The idea of the proof is as follows. Let us consider the language of even palindromes $L = \{ww^R \mid w \in \{a,b\}^*\}$. It can be accepted by a realtime deterministic WK-automaton without counters [9]. Its concatenation with itself is $L \cdot L = \{ww^R uu^R \mid w,u \in \{a,b\}^*\}$. Since an RWKC_m^∞ machine has only memory by counters, there is no way to store a word of arbitrary length as w or u^R could be. Without storing any (or both) of these words our stateless machine is unable to check their reverse. ☐

Corollary 1. *Languages accepted by nondeterministic realtime stateless multicounter WK-automata are not closed under Kleene-closure.*

4 Hierarchies Regarding Nondeterministic Stateless Multicounter WK-Automata

Theorem 1. RWKC_m^k *is more powerful than* dRWKC_m^k*.*

Proof. Clearly all deterministic machines with m counters and k reversals are a special case of nondeterministic machines with the same number of counters and reversals.

For $n \geq 0$ we define the language

$$L_n = \{a^i b^{j_1} a b^{j_2} \cdots a b^{j_n} \mid i, j_k \geq 0\ ,\ k = 1,2,\ldots,n,$$
$$\text{and } i = j_\ell \text{ where } \ell = 1, \text{ or } \ell = 2, \text{ or } \ldots \ell = n\}.$$

Then $L' = \bigcup_{n=1}^{\infty} L_n$ cannot be accepted by a deterministic stateless realtime WK-automata with any number of reversals and any number of counters. The machine does not know for which $\ell \in \{1,2,\ldots,n\}$ the equality stands, so it must be tested for all possible values. It is clear that for any numbers $k, m \in \mathbb{N}$: $w \in L_{2(k+1)(m+1)+1}$ cannot be accepted by a machine in WKC_m^k.

Furthermore, in the nondeterministic case, only one counter is enough to accept the language. If $i = j_k$ for a k in $\{1, 2, \ldots, n\}$, then the right head can read the end of the input till the end of the subword b^{j_k} without changing the counter, then at the beginning of that block (i.e., at the previous symbol a or \$ in case of $k = n$, i.e., at the last block) the counter is changed to 1. If the counter is positive, both heads move while reading symbols. The left head reads the prefix a^i while the right head reads b^{j_k}. Thus they are reading the same number of symbols. At the end of the blocks the counter set to be zero again only if both blocks ended at the same step. Then the remainder of the input can be read by the first head to finish the input. □

The following theorem is a known result regarding to deterministic WK-automata.

Proposition 4. *[2] For fixed $i \geq 3$, languages in the form $L = \{a^{in} \mid n \geq 0\}$, cannot be accepted by any stateless deterministic realtime reversal bounded WK-automaton.*

These languages are clearly accepted by nondeterministic stateless WK-automata with k-reversals and m-counters, where k, m depend on the value of i. If the machine is 1-reversal, $m = \lceil \frac{i}{2} \rceil - 1$ counters are needed to accept L (for $i = 1, 2, \ldots$).

A linear grammar $G = (N, T, P, S)$ is in normal form if every rule has one of the following forms $A \to aBb, A \to aB, A \to Ba, A \to a$ with $A, B \in N$ and $a, b \in T$. This can be achieved by basic transformations of the rules.

Theorem 2. *Let $G = (N, T, P, S)$ be a linear grammar in normal form. Then $L(G)$ can be accepted by a nondeterministic stateless realtime WK-automaton with unbounded number of reversals and $m = |N| - 1$ counters so that $\sum_{i=1}^{m} c_i \leq 1$ at any time of the computation (where c_i is the value of the i-th counter).*

Proof. We give a sketch of the construction of the machine which accepts $L(G)$. Each counter represents a nonterminal which is not S. The symbol S is represented by the sign vector $(0, 0, \ldots, 0)$. Initially both heads make a step from the boundary markers without changing the values of the counters. For a rule $X \to aYb \in P$ the move

$$((a, b; 0, 0, \ldots, 0, s_X = 1, 0, \ldots, 0), \infty) \to$$

$$(1, 1; 0, 0, \ldots, 0, e_X = -, 0, e_Y = +, 0, \ldots, 0)$$

is added. The order of C_X and C_Y in the sequence may be different depending on how the counters were assigned to the nonterminals. If $X = Y$, then the counter is not changed. For a rule $X \to aY \in P$ similar moves are added (for $\varphi(M) > 1$. This time H_2 does not move: b can be any terminal symbol, so the moves should be constructed for all $b \in T$, and only the left head makes a step. The case of $X \to Ya \in P$ is very similar and $X \to a \in P$ can be handled with $\varphi(M) = 1$ by reading an a with the left input head and decreasing the counter C_X. □

Remark 1. It is known that counter machines with a finite control and two counters accept all recursively enumerable languages [8]. In this way we can add two additional counters to the automaton in Theorem 2 and get machine equivalent to the Turing-machine. Thus, reversal boundedness is a key factor at hierarchy of stateless multicounter WK-automata by limiting their power.

Let us define the following languages over $\Sigma = \{a, b, c\}$ for $j \geq 1$:

$$L_j = a^{n_1}ca^{n_2}c \cdots a^{n_j}cb^{n_1}c^{n_j}b^{n_2}c^{n_{j-1}} \cdots b^{n_{j-1}}c^{n_2}b^{n_j}c^{n_1},$$

where $n_i \geq 0$ for all $i = 1, \ldots, j$.

Theorem 3. *For any $j \geq 1$, L_j is in* dRWKC^1_j, *but not in* RWKC$^k_{j-1}$ *for any $k \geq 1$.*

Proof. First we remark that without the delimiters c between the subwords a^{n_i}, only one counter is enough to accept L_j.

It is easy to see that L_j can be accepted by a 1-reversal machine with j counters. The machine first counts the number of a's in the prefix a^{n_1} subword with counter 1, then, if a c is read, the second counter is increased by 1 to indicate, that the a^{n_2} subword is going to be read. After reading a^{n_2} and increasing counter 2 in each step, another c comes. This time, while reading it, the second counter is decreased by 1 (it contained $n_2 + 1$, because of the first c) and the third counter is increased to indicate, that now a^{n_3} is to be read and counted. In this way, the machine counts all these a^{n_i} subwords and after the jth one, while reading a c with the left head, it also reads the right end marker, \$ with the right head, while decreasing the jth counter by 1 (this is also because of the previously read c). At this point, the value of counter i ($i = 1, \ldots, j$) is n_i. After these steps, the subword $b^{n_1}c^{n_j}b^{n_2}c^{n_{j-1}} \cdots b^{n_{j-1}}c^{n_2}b^{n_j}c^{n_1}$ is processed while decrementing the corresponding counters.

We use induction on j to show, that L_j cannot be accepted by $j-1$ counters with any number of reversals. For $j = 1$, it is obvious, that the language $L_1 = a^ncb^nc^n$ cannot be accepted without counters. Note, that the expression of L_{k+1} can be constructed from the expression of L_k in the following way:

1. let w and x be the sequences $b^{n_1}c^{n_k} \cdots b^{n_k}c^{n_1}$ and $a^{n_1}ca^{n_2}c \cdots a^{n_k}c$ respectively
2. $L_{k+1} = a^\ell cxb^\ell h(w)c^\ell$, where h is a morphism such that $h(b) = c$ and $h(c) = b$.

Suppose, that for some $k \geq 1$, L_k cannot be accepted with $k-1$ counters with any number of reversals. For L_{k+1}, we have an additional counter. In this counter, we should store the number ℓ to test whether the first block of b's have the same number of symbols. So, the additional counter is needed to store ℓ. Then, after reading a^ℓ, or c^ℓ (or both) we have $k-1$ counters and the remaining part of the input to be processed (without the b^ℓ subword) is in L_k, except that the corresponding b's and the c's in the suffix are exchanged. But that does not help the machine to accept L_k. And we know that L_k cannot be accepted with $k-1$ counters. Thus there is no way to get to b^ℓ without loss of information, so L_{k+1} cannot be accepted with k counters. □

Corollary 2. dRWKC^1_m *and* RWKC^k_{m-1} *are incomparable (for* $k \geq 1, m \geq 2$*).*

Proof. L_m can be accepted by a 1-reversal m-counter machine, but cannot be accepted by any k-reversal $m-1$ counter machines for any $k \geq 1$. On the otherside, the language L' (after Theorem 1) is in RWKC^k_{m-1} for $k \geq 1, m \geq 2$, but it is not in dRWKC^1_m. □

5 Conclusions

Figure 1 contains some hierarchy results about WK-automata based on our new results and [2]. Arrows stand for proper inclusion. RE stands for the class of recursively enumerable languages. Some open problems are represented by dotted arrows. Another further direction of research can be to investigate such multicounter WK-automata where the heads do not sense when they meet, therefore both heads read the full input in an accepting computation. Other extension can also be considered: multicounter WK-automata which may read strings and not just letters in a single transition, for example.

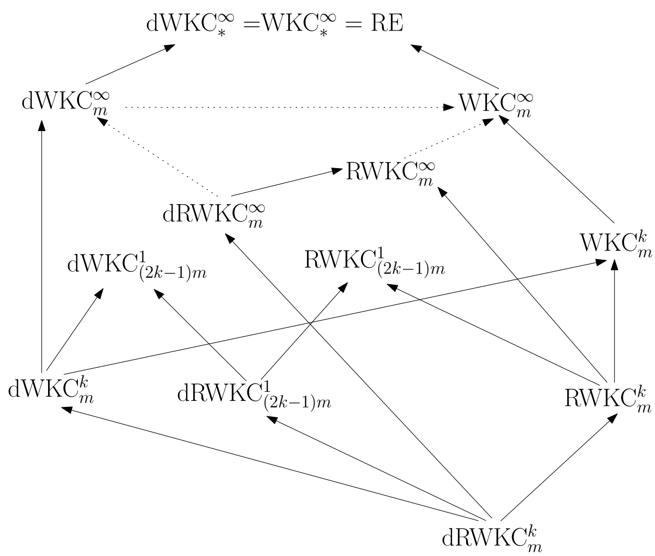

Fig. 1. Hierarchy of stateless multicounter $5' \to 3'$ WK automaton languages

Acknowledgements

The research is supported by the TÁMOP 4.2.1/B-09/1/KONV-2010-0007 project, which is implemented through the New Hungary Development Plan, co-financed by the European Social Fund and the European Regional Development Fund.

References

1. Eğecioğlu, Ö., Ibarra, O.H.: On stateless multicounter machines. In: Ambos-Spies, K., Löwe, B., Merkle, W. (eds.) CiE 2009. LNCS, vol. 5635, pp. 178–187. Springer, Heidelberg (2009)
2. Eğecioğlu, Ö., Hegedüs, L., Nagy, B.: Stateless multicounter $5' \rightarrow 3'$ Watson-Crick Automata. In: Fifth International Conference on Bio-Inspired Computing: Theories and Applications, pp. 1599–1606. IEEE Press, Los Alamitos (2010)
3. Freund, R.: Păun, Gh., Rozenberg, G., Salomaa, A.: Watson-Crick Finite Automata. In: Third Annual DIMACS Symposium on DNA Based Computers, Philadelphia, pp. 535–546 (1994)
4. Ibarra, O.H., Eğecioğlu, Ö.: Hierarchies and characterizations of stateless multicounter machines. In: Ngo, H.Q. (ed.) COCOON 2009. LNCS, vol. 5609, pp. 408–417. Springer, Heidelberg (2009)
5. Ibarra, O.H., Karhumäki, J., Okhotin, A.: On stateless multihead automata: Hierarchies and the emptiness problem. In: Laber, E.S., Bornstein, C., Nogueira, L.T., Faria, L. (eds.) LATIN 2008. LNCS, vol. 4957, pp. 94–105. Springer, Heidelberg (2008)
6. Leupold, P., Nagy, B.: $5' \rightarrow 3'$ Watson-Crick automata with several runs. In: Workshop on Non-Classical Models of Automata and Applications (NCMA), Wroclaw, Poland, pp. 167–180. book@ocg.at, Österreichischen Computer Gesellschaft (2009)
7. Leupold, P., Nagy, B.: $5' \rightarrow 3'$ Watson-Crick automata with several runs. Fundamenta Informaticae 104, pp. 71–91, extended version of [6] NCMA2009 (2010)
8. Minsky, M.L.: Computation: finite and infinite machines. Prentice-Hall, Englewood Cliffs (1967)
9. Nagy, B.: On a hierarchy of $5' \rightarrow 3'$ sensing WK finite automata languages. In: CiE 2009: Mathematical Theory and Computational Practice, Abstract Booklet, pp. 266–275. University of Heidelberg (2009)
10. Nagy, B.: On $5' \rightarrow 3'$ Sensing Watson-Crick Finite Automata. In: Garzon, M.H., Yan, H. (eds.) DNA 2007. LNCS, vol. 4848, pp. 256–262. Springer, Heidelberg (2008)
11. Păun, G.: Computing with Membranes. Journal of Computer and System Sciences 61/1, 108–143 (2000)
12. Păun, G.: Computing with Membranes: An Introduction. Springer, Berlin (2002)
13. Păun, G., Rozenberg, G., Salomaa, A.: DNA Computing: New Computing Paradigms. Springer, Heidelberg (1998)
14. Yang, L., Dang, Z., Ibarra, O.H.: On stateless automata and P systems. In: Pre-Proceedings of Workshop on Automata for Cellular and Molecular Computing, pp. 144–157 (2007)