RAIRO-Theor. Inf. Appl. 55 (2021) 5
https://doi.org/10.1051/ita/2021007

RAIRO - Theoretical Informatics and Applications
www.rairo-ita.org

ON DETERMINISTIC 1-LIMITED $5' \to 3'$ SENSING WATSON–CRICK FINITE-STATE TRANSDUCERS

Benedek Nagy[1] and Zita Kovács[2,*]

Abstract. Finite automata and finite state transducers belong to the bases of (theoretical) computer science with many applications. On the other hand, DNA computing and related bio-inspired paradigms are relatively new fields of computing. Watson–Crick automata are in the intersection of the above fields. These finite automata have two reading heads as they read the upper and lower strands of the input DNA molecule, respectively. In $5' \to 3'$ Watson–Crick automata the two reading heads move in the same biochemical direction, that is, from the $5'$ end of the strand to the direction of the $3'$ end. However, in the double-stranded DNA, the DNA strands are directed in opposite way to each other, therefore $5' \to 3'$ Watson–Crick automata read the input from the two extremes. In sensing $5' \to 3'$ automata the automata sense if the two heads are at the same position, moreover, the computing process is finished at that time. Based on this class of automata, we define WK transducers such that, at each transition, exactly one input letter is being processed, and exactly one output letter is written on a normal output tape. Some special cases are defined and analyzed, *e.g.*, when only one of the reading heads is being used and when the transducer has only one state. We also show that the minimal transducer is uniquely defined if the transducer is deterministic and it has marked output, *i.e.*, the output letter written in a step identifies the reading head that is used in that transition. We have also used the functions 'processing order' and 'reading heads' to analyze these transducers.

Mathematics Subject Classification. 68Q45.

Received December 21, 2019. Accepted March 25, 2021.

1. Introduction

Finite automata and finite state transducers are very basic concepts of theoretical computer science [6, 27]. The Mealy and Moore automata are widely known old models of finite state transducers (see, *e.g.*, [11]).

On the other hand, Watson–Crick automata (WK automata for short) are introduced in [4] as a specific new computing model connected to DNA computing [23]. The DNA molecules are build up from nucleotides; there are four types of them: Adenine, Cytosine, Guanine, Thymine. A single-stranded DNA can be considered as a string over the alphabet $\{A, C, G, T\}$. The single-stranded DNA molecule has a direction: there is a $5'$ and a $3'$ end. Among the nucleotides the Watson–Crick complementarity relation defines pairs (A, T) and (C, G) such that two single-stranded DNA can form double-stranded DNA molecule if the Watson–Crick pairs are

Keywords and phrases: Watson–Crick transducers, $5' \to 3'$ WK automata, Sensing WK automata, Finite state transducers.

[1] Department of Mathematics, Faculty of Arts and Sciences, Eastern Mediterranean University, Famagusta, North Cyprus, via Mersin-10, Turkey.
[2] Department of Computer Science, Faculty of Informatics, University of Debrecen, 4032 Debrecen, Egyetem tér 1., Hungary.

[*] Corresponding author: kovacs.zitu@gmail.com

in the respective positions of the strands when their $5'$ to $3'$ direction is opposite. Consequently, finite state WK automata, which are working on DNA molecules, have two reading heads. The first head is reading the upper strand, while the second head is reading the lower strand. Therefore, the names first/upper head and second/lower head can be used to identify them. There are several variants of WK automata analyzed in [23]. We recall here the most important ones. From biological point of view there are good reasons to consider special WK automata with restrictions on their sets of states: A WK automaton is called all-final if each of its states is an accepting state. A more specific variant of WK automata is the stateless WK automaton. A stateless WK automaton has exactly one state. Based on the reading process, in arbitrary WK automata the heads may read strings in a transition, in simple WK automata in each transition at most one of the heads can read, while the 1-limited variants are further restricted, in these models, in each transition exactly one input letter is being read (by either head). One may also consider mixed constraints, *i.e.*, one for the set of states and other for the reading process. Deterministic and nondeterministic variants were also studied. In a double-stranded DNA, as we have mentioned, the $5'$ to $3'$ direction of the two strands are opposite. The sensing $5' \to 3'$ WK automata are introduced in [12]. In these automata, the two reading heads move in the same biochemical direction, that is, from the $5'$ end of the strand to the direction of the $3'$ end, respectively; this means that the heads read the input from the two extremes and they are moving in opposite mathematical/physical/computational direction. Moreover, the processing of the input is finished when the heads meet. It can be done, since by the Watson–Crick complementarity relation, at that time, every place of the input was read by one of the heads: the first part till the meeting point of the heads was read by the first head while the second part from this position to the end of the input was read by the second head. This model has a straight connection with the class Lin of linear languages: sensing $5' \to 3'$ WK automata accept exactly this class of the Chomsky hierarchy, and, in fact, it is equivalent to linear automata [10, 14]. Several variations were investigated and analyzed in [8, 9, 13, 15, 21, 22]. The above listed all-final, stateless, simple and 1-limited restrictions have also been studied for $5' \to 3'$ WK automata. Moreover, the determinism is a feature that is independent of the previously mentioned constraints. Deterministic sensing $5' \to 3'$ WK automata accept the language class $2detLin$ that is strictly included in Lin, it strictly includes the class Reg of regular languages, and it is incomparable with $detLin$, *i.e.*, the class accepted by one-turn pushdown automata. The features of $5' \to 3'$ WK automata are mixed with jumping automata, with automata with translucent letters and with pushdown automata in [7], in [18, 19], and in [16], respectively. In this paper, we go in another direction: we connect the idea of $5' \to 3'$ WK automata to the concept of transducers. We use the input tape as a Watson–Crick tape, *i.e.*, the input is a DNA molecule, while the output is a normal tape, as Watson–Crick transducers are considered in this form in [23]. However, in our transducers the reading heads move in the way as they move in a sensing $5' \to 3'$ WK automaton. Several variants of sensing $5' \to 3'$ WK transducers can be investigated; we start our analysis by relatively simple ones. We will study 1-limited WK transducers such that they read and write exactly one letter in every transition. Moreover, our transducers are completely defined: every input can be processed by them. We mainly focus on deterministic variants. (We note here that, in our previous study [17], we have used the term 'simple' for our WK transducers, however, in fact, they were more restricted, since the process went by reading the input letter by letter and also to build the output letter by letter, thus, we feel that the term '1-limited' also fits, and, therefore, in this paper we use this better-fit term.) Since transducers are not about accepting languages, the all-final restriction has no meaning. In contrast, using no states, *i.e.*, having a WK transducer with only one state, makes sense, and indeed we also study these stateless variants of deterministic sensing 1-limited $5' \to 3'$ WK transducers.

In the next section, we give our definitions: Mealy and Moore-type WK transducers and we prove their equivalences. Then, we show some properties of the mappings they could realize and also we define two functions, the 'reading heads' and the 'processing order' which will be helpful for further studies. In Section 3 we investigate special variants (such as the traditional finite state transducers, the stateless model or the transducers with marked output). In Section 4 we detail some more properties of our WK transducers and we also relate our transducers to computing by observing. Then, concentrating on some special variants of WK transducers, in Section 5 we define equivalent classes among the states of marked-output transducers, and this relation is used

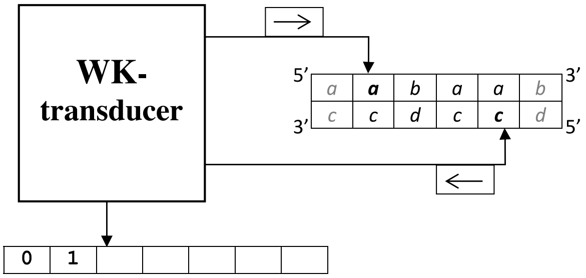


Figure 1. 1-limited $5' \to 3'$ sensing WK transducer.

in Section 6 to show that the minimal transducers are uniquely defined analogously to the normal, usual (Mealy and Moore) finite state transducers. Some conclusions and thoughts about future work closes the paper.

2. $5' \to 3'$ WK transducers

We assume that the reader is familiar with the basic concepts of automata theory, otherwise she/he is referred to [5, 6, 24]. Let V be an alphabet, *e.g.*, $V = \{A, C, G, T\}$. The Watson–Crick complementarity relation assigns A and T to each other, and C and G to each other. We describe a double-stranded DNA molecule over V by $\left[\begin{smallmatrix} w \\ w' \end{smallmatrix}\right]$ where $w, w' \in V^*$, $|w| = |w'|$ such that $\left[\begin{smallmatrix} w \\ w' \end{smallmatrix}\right] \in \{\begin{bmatrix} A \\ T \end{bmatrix}, \begin{bmatrix} T \\ A \end{bmatrix}, \begin{bmatrix} G \\ C \end{bmatrix}, \begin{bmatrix} C \\ G \end{bmatrix}\}^*$, *i.e.*, the two strands are Watson–Crick complements of each other. For two substrands that are not necessarily Watson–Crick complements of each other we use the notation $\binom{u}{v}$, where $u, v \in V^*$. The relation ϱ on V generalizes the Watson–Crick complementarity relation, and it defines DNA strings used in the next two definitions and later on. We use λ to denote the empty word.

Further in this section we define new types of finite state transducers. A sketch of a $5' \to 3'$ WK transducer is shown in Figure 1.

Definition 2.1. A *(nondeterministic)* $5' \to 3'$ *sensing finite Mealy-type Watson–Crick transducer (MeWK transducer)* $\mathbf{A}$ is a sextuple $(V, \varrho, T, Q, s_0, \delta)$ where V is the input (tape) alphabet, $\varrho \subseteq V \times V$ is a symmetric relation, T is the output alphabet, Q is the finite set of states, $s_0 \in Q$ is the initial state and δ is the transition mapping $Q \times \binom{V^*}{V^*} \to 2^{Q \times T^*}$, such that $\delta(q, \binom{u}{v})$ is non-empty only for finitely many triplets $(q, u, v) \in Q \times V^* \times V^*$. The interpretation of $(p, z) \in \delta(q, \binom{u}{v})$ is as follows: in state q, the upper head of the transducer reads the word u and the lower head of the transducer reads the word v, the transducer produces output z and it enters into the state p. A MeWK transducer is *1-limited* if $\delta(q, \binom{u}{v})$ is non-empty implies $|u| + |v| = 1$ and $|z| = 1$. Let $w \in \begin{bmatrix} V^* \\ V^* \end{bmatrix}$ be an input DNA string and $\mathbf{A}$ is a 1-limited MeWK transducer. A *configuration* of $\mathbf{A}$ is a triplet $(u, q, z) \in \begin{bmatrix} V^* \\ V^* \end{bmatrix} \times Q \times T^*$ including the remaining part of the input u, the current state q and the output z generated so far. The computation on w starts with the initial configuration (w, s_0, λ). The configuration $(\left[\begin{smallmatrix} a \\ a' \end{smallmatrix}\right] u \begin{bmatrix} b' \\ b \end{bmatrix}, q, z)$ can be followed by the configuration (u, q', zx) if $(q', x) \in \delta(q, \binom{a}{b})$ where $|ab| = 1$. It is denoted by $(\left[\begin{smallmatrix} a \\ a' \end{smallmatrix}\right] u \begin{bmatrix} b' \\ b \end{bmatrix}, q, z) \Rightarrow (u, q', zx)$. The reflexive and transitive closure of this relation is denoted by $\Rightarrow^*$. The 1-limited MeWK transducer is *sensing* if it finishes the process when its heads meet. Consequently, the *final configuration* on input w is $(\begin{bmatrix} \lambda \\ \lambda \end{bmatrix}, q, z)$ such that $(w, s_0, \lambda) \Rightarrow^* (\begin{bmatrix} \lambda \\ \lambda \end{bmatrix}, q, z)$ for some state $q \in Q$. The word $z \in T^*$ is called an *output* of $\mathbf{A}$ on w.

Analogously we extend the definition of Moore-automata.

Definition 2.2. A *(nondeterministic)* $5' \to 3'$ *sensing finite Moore-type Watson–Crick transducer (MoWK transducer)* $\mathbf{A}$ is a septuple $(V, \varrho, T, Q, s_0, \delta, \mu)$ where the first five components are the same as in Definition 2.1, moreover δ is the transition mapping $Q \times \binom{V^*}{V^*} \to 2^Q$, such that $\delta(q, \binom{u}{v})$ is non-empty only for finitely many triplets $(q, u, v) \in Q \times V^* \times V^*$ and μ is the output mapping $Q \to T^*$. The interpretation $p \in \delta(q, \binom{u}{v})$ is the following: in the state q the upper head of the transducer reads the word u and the lower head of the transducer reads the word v, the transducer enters into the state p, and right after a transition it produces output z, where $z = \mu(p)$. A MoWK transducer is *1-limited* if $\delta(q, \binom{u}{v})$ is non-empty implies $|u| + |v| = 1$ and $|\mu(p)| = 1$ for every $p \in Q$. For an input DNA string $w \in \begin{bmatrix} V^* \\ V^* \end{bmatrix}$ a *configuration* of $\mathbf{A}$ is the same as in Definition 2.1 and the computation on w starts with the initial configuration (w, s_0, λ). If $\mathbf{A}$ is 1-limited, the configuration $(\begin{bmatrix} a \\ a' \end{bmatrix} u \begin{bmatrix} b' \\ b \end{bmatrix}, q, z)$ can be followed by the configuration (u, q', zx) if $q' \in \delta(q, \binom{a}{b})$ and $\mu(q') = x$ where $|ab| = 1$. The notations $(\begin{bmatrix} a \\ a' \end{bmatrix} u \begin{bmatrix} b' \\ b \end{bmatrix}, q, z) \Rightarrow (u, q', zx)$, and $\Rightarrow^*$ are used subsequently. The 1-limited MoWK transducer is *sensing* if it finishes the process when its heads meet: the *final configuration* on input w is $(\begin{bmatrix} \lambda \\ \lambda \end{bmatrix}, q, z)$ such that $(w, s_0, \lambda) \Rightarrow^* (\begin{bmatrix} \lambda \\ \lambda \end{bmatrix}, q, z)$ for some state $q \in Q$. The word $z \in T^*$ is called an *output* of $\mathbf{A}$ on w.

When a MeWK transducer (MoWK transducer) $\mathbf{A}$ processes an input $w \in \begin{bmatrix} V^* \\ V^* \end{bmatrix}$, it generates an output $z \in T^*$, then a *mapping* can be defined $\gamma : \begin{bmatrix} V^* \\ V^* \end{bmatrix} \to 2^{T^*}$: $(w, z) \in \gamma$ if $(w, s_0, \lambda) \Rightarrow^* (\begin{bmatrix} \lambda \\ \lambda \end{bmatrix}, q, z)$ by the MeWK transducer (MoWK transducer) $\mathbf{A}$. Although the two types of transducers work in a slightly different way, the mapping from the set of possible inputs to the possible outputs are of the same nature. Let $\mathbf{A_1}$ be a MeWK transducer or a MoWK transducer, and $\mathbf{A_2}$ be a MeWK transducer or a MoWK transducer. Let γ_1 be the mapping defined by $\mathbf{A_1}$ and γ_2 be defined similarly by $\mathbf{A_2}$. We say that $\mathbf{A_1}$ is *equivalent* with $\mathbf{A_2}$ if the mappings γ_1, γ_2 are the same.

In [3] there are various ways to define deterministic WK automata playing with the complementary relation as well. However, at sensing $5' \to 3'$ WK automata, since exactly one of the strands is read at every pair when the input is fully processed, there is only one usual concept of determinism [20, 22]. Based on this fact, we are using the following definition for our WK transducers.

Definition 2.3. A MeWK transducer (MoWK transducer) with non-empty input is *deterministic* if at each possible configuration exactly one transition is possible.

Lemma 2.4. *For each 1-limited MeWK transducer* $\mathbf{A_1}$ *there is a 1-limited MoWK transducer* $\mathbf{A_2}$ *such that* $\mathbf{A_1}$ *and* $\mathbf{A_2}$ *are equivalent. Moreover, for each deterministic 1-limited MeWK transducer* $\mathbf{A_1}$ *there is a deterministic 1-limited MoWK transducer* $\mathbf{A_2}$ *such that* $\mathbf{A_1}$ *and* $\mathbf{A_2}$ *are equivalent.*

Proof. Let $\mathbf{A_1} = (V, \varrho, T, Q, s_0, \delta)$ be a MeWK transducer. Then we construct the MoWK transducer $\mathbf{A_2} = (V, \varrho, T, Q', s'_0, \delta', \mu)$, where Q', s'_0, δ', μ are defined in the following way. Let $Q' = Q \times T$. If $q' = (q, x) \in Q'$, then $\mu(q') = x$. Let us fix $s'_0 = (s_0, x)$ for a fixed letter $x \in T$ (any letter x can be chosen). If $(p, y) \in \delta(q, \binom{a}{b})$, then for every $x' \in T$ let $\delta'((q, x'), \binom{a}{b}) \ni (p, y)$, where $(p, y) \in Q'$. By this construction it is easy to check that $\mathbf{A_2}$ generates the same mapping as $\mathbf{A_1}$. Considering a deterministic MeWK transducer $\mathbf{A_1} = (V, \varrho, T, Q, s_0, \delta)$, let $\mathbf{A_2} = (V, \varrho, T, Q', s'_0, \delta', \mu)$ be defined as follows. Q', μ and s'_0 as above. If $(p, y) = \delta(q, \binom{a}{b})$, then for every $x' \in T$ let $(p, y) = \delta'((q, x'), \binom{a}{b})$, where $(p, y) \in Q'$. Consequently $\mathbf{A_2}$ is also deterministic and equivalent to $\mathbf{A_1}$. □

Lemma 2.5. *For each 1-limited MoWK transducer* $\mathbf{A_1}$ *there is a 1-limited MeWK transducer* $\mathbf{A_2}$ *such that* $\mathbf{A_1}$ *and* $\mathbf{A_2}$ *are equivalent. Moreover, if the MoWK-transducer* $\mathbf{A_1}$ *is deterministic, then there is a deterministic 1-limited MeWK-transducer* $\mathbf{A_2}$ *such that they are equivalent.*

Proof. Let $\mathbf{A_1} = (V, \varrho, T, Q, s_0, \delta, \mu)$ be a MoWK transducer. Then we construct the MeWK transducer $\mathbf{A_2} = (V, \varrho, T, Q, s_0, \delta')$, where δ' is defined in the following way. For every transition $p \in \delta(q, \binom{a}{b})$ and output $\mu(p)$ of $\mathbf{A_1}$, let $(p, \mu(p)) \in \delta'(q, \binom{a}{b})$ be a transition in $\mathbf{A_2}$. By this construction it is easy to check that $\mathbf{A_2}$ generates the same mapping as $\mathbf{A_1}$. Considering a deterministic MoWK transducer $\mathbf{A_1} = (V, \varrho, T, Q, s_0, \delta, \mu)$, let $\mathbf{A_2} =$

$(V, \varrho, T, Q, s_0, \delta')$ be defined as follows. For every transition $p = \delta(q, \binom{a}{b})$ and output $\mu(p)$ let $(p, \mu(p)) = \delta'(q, \binom{a}{b})$ is the transition mapping of $\mathbf{A_2}$. Consequently $\mathbf{A_2}$ is also deterministic and equivalent to $\mathbf{A_1}$. □

By the above lemmas, the family of 1-limited and sensing MeWK transducers are equivalent to the family of 1-limited and sensing MoWK transducers. Further we will use the notation WK transducers for 1-limited and sensing MeWK transducers but our results hold for 1-limited and sensing MoWK transducers also.

As we have already mentioned, in sensing $5' \to 3'$ WK transducers exactly one of the heads read an input letter of a pair $\left[\begin{smallmatrix} a \\ a' \end{smallmatrix}\right]$ in every position of the input and the other letter of this pair is not read by any of the heads. Assuming that ϱ is a bijection on V, as it is on the set of nucleotides $\{A, C, G, T\}$, the transducer has all information about the processed input double string. Unless otherwise mentioned, in the following, by a WK transducer we always mean a completely defined deterministic WK transducer with the identity relation $\varrho = \iota$ (for simplicity). Since we will read only one component of each pair on any position of the double-stranded DNA string in the deterministic case we can use the identity relation ι as the complementarity relation ϱ without loss of generality.

One important advantage of the identity relation in role of ϱ is that we can simplify our notation. Since the complement of any word w is itself, the input DNA string is $\left[\begin{smallmatrix} w \\ w \end{smallmatrix}\right]$, and can simply be written as w. According to this, the notation of δ and the configurations are also simplified. Furthermore, the mapping of a WK transducer is considered of the form $\gamma : V^* \to T^*$.

We have an analogous result to [22] where it is proven for 1-limited deterministic WK automata that in each state only one of the reading heads can move. In fact, the next lemma formally describes what determinism means for 1-limited WK transducers.

Lemma 2.6. *Let* $\mathbf{A}$ *be a deterministic WK transducer. Then* $Q = Q_1 \cup Q_2$ *with* $Q_1 \cap Q_2 = \emptyset$ *with the following properties. For every pair* $q \in Q_1$, $a \in V$*: there is a pair* $p \in Q, z \in T$ *such that* $\delta(q, \binom{a}{\lambda}) = (p, z)$ *and* $\delta(q, \binom{\lambda}{a}) = \emptyset$*. Furthermore, for every pair* $q \in Q_2$, $a \in V$*:* $\delta(q, \binom{a}{\lambda}) = \emptyset$ *and there is a pair* $p \in Q, z \in T$ *such that* $\delta(q, \binom{\lambda}{a}) = (p, z)$*.*

Proof. The proof goes by contradiction. Let us assume that $\mathbf{A}$ is deterministic and $Q = Q_1' \cup Q_2'$ such that for each state $q \in Q_1'$ there is $a \in V$ such that $\delta\left(q, \binom{a}{\lambda}\right) \neq \emptyset$ and for each state $p \in Q_2'$ there is $b \in V$ such that $\delta\left(p, \binom{\lambda}{b}\right) \neq \emptyset$. Contrary to the statement of the lemma, assume that $Q_1' \cap Q_2' \neq \emptyset$. Then there exists $q \in Q_1' \cap Q_2'$, *i.e.*, $\delta(q, \binom{a}{\lambda}) = (p_a, x_a)$ and $\delta(q, \binom{\lambda}{b}) = (p_b, x_b)$. In this case for the configuration $(aub, q, z) \in V^* \times Q \times T^*$ there are at least two possible transitions: $(\delta(q, \binom{a}{\lambda}) = (p_a, x_a)$ or $\delta(q, \binom{\lambda}{b}) = (p_b, x_b))$, so that the transducer is not deterministic. □

We continue this section by results about the possible mappings that WK transducers can do.

Theorem 2.7. *Let* $\mathbf{A} = (V, \iota, T, Q, s_0, \delta, \mu)$ *be a WK transducer and the mapping* $\gamma : V^* \to T^*$*:* $(w, z) \in \gamma$ *if* $(w, s_0, \lambda) \Rightarrow^* (\lambda, q, z)$ *be given by the WK transducer* $\mathbf{A}$*. Then* $|w| = |\gamma(w)| = |z|$*, i.e.,* γ *is "keeping the length".*

Proof. The proof goes by induction. Let $w \in V^*$ be the input string and $(w, s_0, \lambda) \Rightarrow^* (\lambda, q, z)$ be the configuration of $\mathbf{A}$ as in the theorem. Then the output of $\mathbf{A}$ on w is z. First, as the base case, we assume that $|w| = 0$ so $w = \lambda$ and therefore $z = \lambda$. Now, let $|w| = 1$. Since $\mathbf{A}$ is 1-limited, only one of the following transitions is allowed at state $s_0 \in Q$: either $\delta(s_0, \binom{w}{\lambda})$ or $\delta(s_0, \binom{\lambda}{w})$, and furthermore the output $|z| = 1$. Now, as the induction hypothesis, let us assume that $|w| = |\gamma(w)| \leq n$ holds for every possible $w \in V^*$ up to this length. Let us compute $|\gamma(w)|$ when $|w| = n + 1$. Let $w = w_1 a w_2$ where $(w_1 a w_2, s_0, \lambda) \Rightarrow^* (a, q', z') \Rightarrow (\lambda, q, z)$, *i.e.*, $\mathbf{A}$ processes first the n letters of $w_1 w_2$ by producing z' which has length n by the hypothesis. Since $\mathbf{A}$ is 1-limited, only one of the following transitions is allowed at state $q' \in Q$: either $\delta(q', \binom{a}{\lambda})$ or $\delta(q', \binom{\lambda}{a})$, let (q, x) be the pair assigned to the allowed transition. Hence $(w_1 a w_2, s_0, \lambda) \Rightarrow^* (\lambda, q, z)$ with $z = z'x$ and $|x| = 1$. The output of $\mathbf{A}$ on w is z such that $|z| = |z'x| = n + 1$. Thus, by induction on the length n, the statement of the theorem is proven. □

The mapping γ that can be defined by a sensing $5' \to 3'$ WK transducer has the property that the prefix of the output depends only on the prefix and the suffix of the input, more formally we have the following theorem.

Theorem 2.8. *Let $\mathbf{A} = (V, \iota, T, Q, s_0, \delta)$ be a WK transducer and γ be the mapping $V^* \to T^*$ defined by $\mathbf{A}$. Let $w \in V^*$ be an input word and let $z = \gamma(w)$. Then for any $z_1, z_2 \in T^*$ with $z = z_1 z_2$, w can be written of the form $w_1 w_2 w_3$ such that $\gamma(w_1 w_3) = z_1$ (moreover by the previous theorem: $|w_1 w_3| = |z_1|$ and consequently $|w_2| = |z_2|$).*

Proof. We will prove a stronger statement by induction on the length. So let $w \in V^*$ be an arbitrary input string. Then z is defined by the WK transducer $\mathbf{A}$: $\gamma(w) = z$. Let the length of z_1 be zero, *i.e.*, $z_1 = \lambda$, and thus, $w_1 = w_3 = \lambda$, $z_2 = z$, $w_2 = w$. In this case clearly $\gamma(\lambda) = \lambda$ and it makes our statement true.

As the induction hypothesis, let us assume that for any prefix z_1 of z ($z_1 \neq z$), there is a decomposition $w_1 w_2 w_3$ of w such that $\gamma(w_1 w_3) = z_1$. Let the length of z_1 be n (clearly, $n < |z|$). Then we are proving the statement for the prefix z_1' with length $|z_1'| = n + 1$. Then, let q be the state of $\mathbf{A}$ when it finishes the process of $w_1 w_3$ and let $w_2 = a w_2' b$ where $a \in V, b = \lambda$ if $q \in Q_1$ (*i.e.*, the upper head reads in state q) and $a = \lambda, b \in V$ if $q \in Q_2$ (*i.e.*, the lower head reads in state q). Then $|w_1 a b w_2| = n + 1$ and $\gamma(w_1 a b w_2) = z_1 x$ since the deterministic process first proceeds the prefix w_1 and the suffix w_2 of the input and in state q the letter ab is being read (since one of a and b is λ, ab denotes a string of a sole letter). In this way, $z_2 = x z_2'$ and therefore the statement is proven. Moreover, it is also true that the process goes letter by letter in a deterministic manner, and therefore, if $\gamma(w_1 w_3) = z_1$ such that w_1 is processed by the upper head and w_3 is processed by the lower head, then for any $w_2 \in V^*$, $\gamma(w_1 w_2 w_3) = z_1 z_2$ for an appropriate word $z_2 \in T^*$. □

Based on the previous result we see that WK transducers are a kind of "border-prefix" transducers, *i.e.*, they map the prefix/suffix (or their combination) to the prefix of the output.

Furthermore, we define two functions for WK transducers. The function *reading heads* (denoted by r) gives information about the working order of the heads of the transducer on any given input: it tells at every step which head was reading in the process. Formally:

Definition 2.9. Let $\mathbf{A}$ be a deterministic $5' \to 3'$ sensing WK transducer. The function associated to $\mathbf{A}$ of the form $r : V^* \to \{u, l\}^*$ is called *reading heads* if for every $w \in V^* : r(w) = r_1 r_2 ... r_n$, where $r_i = u$ if the upper head was reading at i-th step and $r_i = l$ if the lower head was reading at i-th step, for every $i \in \{1, 2, ..., n\}$.

Actually, function r also exhibits the strong relation established between linear grammars and languages with Watson–Crick automata, see, *e.g.*, [12, 21, 25, 26]. The function *processing order* (denoted by p) can be used to know for each letter of the input word in which step it was read. Its formal definition can be found below.

Definition 2.10. Let $\mathbf{A}$ be a deterministic $5' \to 3'$ sensing WK transducer. The function called *processing order* is of the form $p : V^* \to \mathbb{N}^*$ and for each word $w \in V^*$ (where $|w|$ is n) p takes each of the values of $\mathbb{N} \cap [1, n]$ exactly once: let $w = w_1 ... w_n$ where $w_1, ..., w_n \in V$, then $p(w)$ is the concatenation of the processing orders of the letters of w, *i.e.*, it is $p_1, ..., p_n$, where $p_j = i$ if the i-th step of the computation on w is either $(w_j v, q, z) \Rightarrow (v, q', zx)$ with $(q', x) = \delta(q, \binom{w_j}{\lambda})$ or $(v w_j, q, z) \Rightarrow (v, q', zx)$ with $(q', x) = \delta(q, \binom{\lambda}{w_j})$ for some $q \in Q$ with $|z| + 1 = i$.

Remark 2.11. For every WK transducer, for any input word $w \in V^* : |w| = |r(w)| = |p(w)|$.

In the next section we show 1-limited $5' \to 3'$ sensing Watson–Crick finite-state transducers with some special properties.

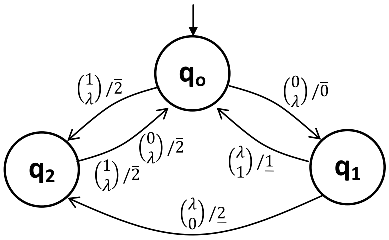
3. Special WK transducers

Since the model is still very general, in this paper, we start to analyze some more specific variants. In this section, we list these variants with some examples to highlight their power.

First let us analyze the variant with *marked-output*. In the following definition we still refer to the notation Q_1 and Q_2 defined in Lemma 2.6.

	q_0	q_1	q_2
0	$(q_1, \overline{0})$	$(q_2, \underline{2})$	$(q_0, \overline{2})$
1	$(q_2, \overline{2})$	$(q_0, \underline{1})$	$(q_0, \overline{2})$

FIGURE 2. Transition δ of $\mathbf{A}$.



q₀
q₁
q₂
$\binom{1}{\lambda}/\overline{2}$
$\binom{0}{\lambda}/\overline{0}$
$\binom{0}{\lambda}/\overline{2}$
$\binom{1}{\lambda}/\overline{2}$
$\binom{\lambda}{1}/\underline{1}$
$\binom{\lambda}{0}/\underline{2}$

FIGURE 3. WK transducer with marked-output.

Definition 3.1. Let $T_1 = \{\overline{a_1}, ..., \overline{a_j}\}$ and $T_2 = \{\underline{a_1}, ..., \underline{a_k}\}$ be two disjoint alphabets. Let $\mathbf{A}$ be a WK transducer and $T = T_1 \cup T_2$ with $T_1 \cap T_2 = \emptyset$. Let $q \in Q_1$, then $\delta(q, \binom{a}{\lambda}) = (p, x)$ with $x \in T_1$. Let $q \in Q_2$, then $\delta(q, \binom{\lambda}{a}) = (p, x)$ with $x \in T_2$. A transducer $\mathbf{A}$ with this property is called *WK transducer with marked-output.*

Notice that in every transition when the first reading head is used, then a letter from T_1 is written to the output, and when the second reading head is used, then a letter from T_2 is written to the output.

Example 3.2. Let $\mathbf{A} = (\{0, 1\}, \iota, \{\overline{0}, \overline{2}, \underline{1}, \underline{2}\}, \{q_0, q_1, q_2\}, q_0, \delta)$ be a WK transducer where the table shown in Figure 2 gives the transition function δ. In states q_0, q_2, the upper head reads an input letter and the output of the WK transducer is an overlined letter, furthermore, in state q_1, the lower head reads an input letter and the output of the WK transducer is an underlined letter. (Using the notations as before we have: $q_0, q_2 \in Q_1, q_1 \in Q_2$.) If for an input DNA string $w \in V^*$ the mapping $\gamma(w) = z$ has only letters $\overline{0}$ and $\underline{1}$, then w is in the form of $0^n 1^n$ or $0^{n+1} 1^n$. This means the string w started with some 0's and then followed by the same number or one less of 1's. The upper head reads the 0's and the lower head reads the 1's. If $x = \underline{2}$ or $x = \overline{2}$ for some letter $x \in z$, then w is not in the mentioned form. The graph of this WK transducer is shown in Figure 3.

For example, if the input string is $w_1 = 000111$, then $\gamma(w_1) = \overline{0}\underline{1}\overline{0}\underline{1}\overline{0}\underline{1}$ and if the input string is $w_2 = 00111$, then $\gamma(w_2) = \overline{0}\underline{1}\overline{0}\underline{1}\overline{2}$.

Therefore, one can identify which of the reading heads read a letter in a transition by the letter written to the output: for w_1 the first, second, first, second, first, second order and for w_2 the first, second, first, second, first order.

It is obvious that there is not any traditional Mealy/Moore transducer that is able to count arbitrarily large number of 0's and 1's, and thus the mapping γ of the previous example cannot be computed by these traditional models.

Theorem 3.3. *For any mapping $\gamma : V^* \to T^*$ defined by a WK transducer $\mathbf{A}$, there is a WK transducer $\mathbf{A}'$ with marked output, defining $\gamma' : V^* \to (T')^*$, and the homomorphism $h : T' \to T$ such that $\gamma(w) = h(\gamma'(w))$ for every $w \in V^*$.*

Proof. Let $\mathbf{A} = (V, \iota, T, Q, s_0, \delta)$ be a WK transducer defining γ. Further, let $T' = T_1 \cup T_2$ where $T_1 = \{\overline{x} \mid x \in T\}$ and $T_2 = \{\underline{x} \mid x \in T\}$, and $\mathbf{A}' = (V, \iota, T', Q, s_0, \delta')$, where δ' is defined as follows. For each transition $(p, x) \in \delta\left(q, \binom{a}{\lambda}\right)$, let $(p, \overline{x}) \in \delta'\left(q, \binom{a}{\lambda}\right)$ and for each transition $(p, x) \in \delta\left(q, \binom{\lambda}{a}\right)$ let $(p, \underline{x}) \in \delta\left(q, \binom{\lambda}{a}\right)$. Clearly, $\mathbf{A}'$ is a WK transducer with marked output. Now, by letting h a homomorphism that assigns $x \in T$ to each

$\overline{x} \in T_1$ and assigns $x \in T$ to each $\underline{x} \in T_2$, for any input $w \in V^*$, $\gamma(w)$, *i.e.*, the output word for w by the WK transducer $\mathbf{A}$ is the same as $h(\gamma'(w))$, *i.e.*, the homomorphic map of the output word of $\mathbf{A}'$. □

Based on the previous result we can say that WK transducers with marked output are almost (with resp. of a homomorphism) as powerful as our general WK transducers.

On the other hand, knowing the image $\gamma(w)$ of a word w (but not the WK transducer behind), generally, it is not straightforward to see in which transition which of the heads is moving. Since h is not injective, it could be a hard task to discover how the mapping is provided. However, knowing the output $\gamma(w)$ and also the value of the associated reading heads function, $r(w)$, we can state the following.

Proposition 3.4. *Let $w \in V^*$ be an input word. If the output $\gamma(w)$ and also $r(w)$ are given for a specific WK transducer $\mathbf{A}$, then one can construct also the marked output of $\mathbf{A}$ on w.*

Proof. Let us analyze the computation of $\mathbf{A}$ on the input w where $|w| = n$. Let $\gamma(w) = \gamma_1 ... \gamma_n$ and $r(w) = r_1 ... r_n$. We will use the notation $\gamma'(w) = \gamma'_1 ... \gamma'_n$ for the marked output of w. If at step i we have $r_i = u$ then $\gamma'_i = \overline{\gamma_i}$ and if $r_i = l$ then $\gamma'_i = \underline{\gamma_i}$. □

Example 3.5. Let $\mathbf{A}$ be a WK transducer such that after processing the input word $w = abaab$ we have $\gamma(w) = xyxyx$ and the output of the reading heads function $r(w) = uulul$. Based on Proposition 3.4 we can construct the marked output: $\gamma'(w) = \overline{xy}\underline{x}\overline{y}\underline{x}$.

3.1. Using only the upper head

If the transition function $\delta(q, \binom{a}{b})$ is non-empty only when $|a| = 1$, *i.e.*, $b = \lambda$ in every possible transition, then the transducer using only the upper head to read the input. It is easy to see that transducers with this property are equivalent to the classical (Mealy) transducers.

Note that WK transducers using only the upper reading head can also be seen as special cases of WK transducers with marked output choosing $T_1 = T$ and $T_2 = \emptyset$.

This together with Example 3.2 shows that WK transducers and WK transducers with marked output are more powerful than traditional finite state transducers (Mealy and Moore automata).

Moreover, we can see that there are special properties when we use only the upper head. First, we can see that the result stated in Theorem 2.8 is also specialized. In this case, we map the prefix of the input to the prefix of the output and the suffix, in fact, is not used.

Proposition 3.6. *Let $\mathbf{A} = (V, \iota, T, Q, s_0, \delta)$ be a special WK transducer where only the upper head reads the input and γ be the mapping $V^* \to T^*$ defined by $\mathbf{A}$. Let $w \in V^*$ be an input word and let $z = \gamma(w)$. Then for any $z_1, z_2 \in T^*$ with $z = z_1 z_2$, w can be written of the form $w_1 w_2$ such that $\gamma(w_1) = z_1$ (moreover by the Theorem 2.7: $|w_1| = |z_1|$ and consequently $|w_2| = |z_2|$).*

Furthermore, the functions reading heads r and processing order p also show the speciality of transducers having this special form.

Proposition 3.7. *Let $\mathbf{A} = (V, \iota, T, Q, s_0, \delta)$ be a special WK transducer where only the upper head reads the input and γ be the mapping $V^* \to T^*$ defined by $\mathbf{A}$. Let $w \in V^*$ be an input word such that $|w| = n$ and let $z = \gamma(w)$. Then $r(w)$ contains only the letter u and $p(w)$ has the numbers from 1 to n in increasing order, i.e., $r(w) = u^n$ and $p(w) = 1, 2, ..., n$.*

3.2. Using only the lower head

If the transition function $\delta(q, \binom{a}{b})$ is non-empty only when $|b| = 1$, *i.e.*, $a = \lambda$ in every possible transition, then the transducer using only the lower head to read the input. The next example shows the power of such WK transducers.

Alur and Cerný defined the streaming transducers in [1]. They argued that reversing a string is a hard task by finite state transducers. They solved this problem by introducing a single data variable and a single data

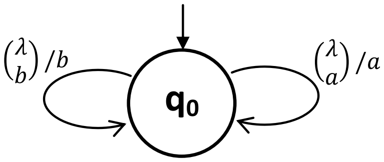


FIGURE 4. A 1-limited example for WK transducer, where only the lower head is used.

string variable and they highlighted the power of the streaming string transducer by this example. In our model the string reversing can be solved in the way presented in the next example.

Example 3.8. Let $\mathbf{A} = (\{a,b\}, \iota, \{a,b\}, \{q_0\}, q_0, \delta)$ be a WK transducer such that $\delta(q_0, \binom{\lambda}{a}) = (q_0, a)$, $\delta(q_0, \binom{\lambda}{b}) = (q_0, b)$. See Figure 4.

We also need only one state to do reversing the input strings, just like Alur and Cerný but we do not need any sophisticated data structure (variables).

We note here that the class of WK transducers mentioned in this subsection also belongs to the class of WK transducers with marked output by choosing $T_1 = \emptyset$ and $T_2 = T$.

Similarly to the previous special case when we used only the upper head, we can see that there are special properties when we use only the lower head also.

First, one can see that the statement of Theorem 2.8 can be strengthened. In fact, these transducers map the suffix of the input to the prefix of the output and the prefix of the input is not used. Formally, we can state the following.

Proposition 3.9. *Let $\mathbf{A} = (V, \iota, T, Q, s_0, \delta)$ be a special WK transducer where only the lower head reads the input and γ be the mapping $V^* \to T^*$ defined by $\mathbf{A}$. Let $w \in V^*$ be an input word and let $z = \gamma(w)$. Then for any $z_1, z_2 \in T^*$ with $z = z_1 z_2$, w can be written of the form $w_1 w_2$ such that $\gamma(w_2) = z_1$ (moreover by the Theorem 2.7: $|w_2| = |z_1|$ and consequently $|w_1| = |z_2|$).*

(Note: Since the lower head reads the input with the direction from the last letter to the first, then z has a more special property: the first letter of z_1 belongs to the last letter of w_2 etc.)

Moreover, the functions reading heads r and processing order p also shows the speciality of these transducers by having special forms.

Proposition 3.10. *Let $\mathbf{A} = (V, \iota, T, Q, s_0, \delta)$ be a special WK transducer where only the lower head reads the input and γ be the mapping $V^* \to T^*$ defined by $\mathbf{A}$. Let $w \in V^*$ be an input word such that $|w| = n$ and let $z = \gamma(w)$. Then $r(w)$ contains only the letter l and $p(w)$ has the numbers from 1 to n in decreasing order, i.e., $r(w) = l^n$ and $p(w) = n, n-1, ..., 1$.*

3.3. Stateless WK transducers

In this subsection we consider deterministic WK transducers with the set of states as a singleton set (*i.e.* $Q = \{q\}$). In fact, these transducers are the counterpart of the no-state (or stateless) 1-limited 5′ → 3′ WK automata. In these transducers there is no state memory, *i.e.*, they cannot store any information in their states.

By analysing these transducers, first of all, by Lemma 2.6, one can see that either only the upper, or only the lower head can be used in all transitions. In this sense, each no-state WK transducer belongs to one of the previously described classes, *i.e.*, it uses only its upper or only its lower head. On the other hand, since there is no state memory, we can state a more concrete result about them. As we will see, in this case, Moore-type WK transducers are more restricted than the Mealy-type WK transducers.

Theorem 3.11. *Let $\mathbf{A} = (V, \iota, T, \{q\}, q, \delta)$ be a stateless Mealy-type WK transducer. Then there are the following two possibilities:*

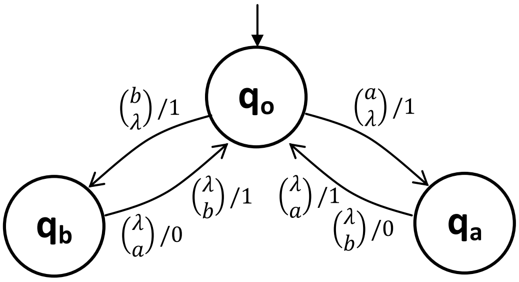


FIGURE 5. An example for WK transducer, where the heads reads alternately.

1. *If* **A** *uses only its upper head, then the mapping* γ *realizes just an alphabetic morphism (similar to the mappings of stateless Mealy automata), that is, for each* $a \in V$*, there is exactly one* $\gamma(a) \in T$*, and for each word* $w = w_1w_2...w_n$ *with* $w_1, w_2, ..., w_n \in V$*,* **A** *maps* $\gamma(w_1)\gamma(w_2)...\gamma(w_n)$ *to* w*.*
2. *If* **A** *uses only its lower head, then* γ *is a reversed alphabetic morphism, i.e., it is an alphabetic morphism of the reversal of the input word, that is, for each* $a \in V$*, there is exactly one* $\gamma(a) \in T$*, and for each word* $w = w_1w_2...w_n$ *with* $w_1, ..., w_n \in V$*,* **A** *maps* $\gamma(w_n)...\gamma(w_2)\gamma(w_1)$ *to* w*.*

Proof. Since **A** is a deterministic 1-limited transducer, at each step for every input letter a we have exactly one transition $\delta(q, a) = (q, x)$. Hence the output mapping $\gamma(a) = x$. **A** assigns $\gamma(w_i)$ to every w_i for $i = 1, ..., n$ and in the first case the upper head reads w letter by letter from the first letter till the last one so after reading w we have $\gamma(w) = \gamma(w_1)...\gamma(w_n)$. If only the lower head was used, then it reads w from the last letter till the first letter, hence $\gamma(w) = \gamma(w_n)...\gamma(w_1)$. □

We underline here that Example 3.8 shows a stateless MeWK transducer.

In contrast to Mealy-type transducers, the stateless constraint is much stronger for Moore-type transducers:

Theorem 3.12. *Let* $\mathbf{A} = (V, \iota, T, \{q\}, q, \delta, \mu)$ *be a stateless Moore-type WK transducer. Then, independently of the fact if* **A** *uses only its upper head or its lower head, there is a letter* $b \in T$ *such that the mapping* γ *just give the length of the input in unary code, i.e., for any* $w \in V^*$*,* $\gamma(w) = b^{|w|}$*.*

Proof. Since there is only one state q, the output in each step of the computation is $\mu(q) \in T$. Let $b = \mu(q)$. Then, **A** is just counting the length of the input word and puts as many b's to the output as the length of the input. □

Since, in fact, each of our stateless WK transducers is either using only its upper head or only its lower head, about the functions r and p Proposition 3.7 and 3.10 can be applied, respectively.

3.4. Using the heads alternately

The following example is connected to the well-known problem of words: one has to decide whether a word is the same as its reverse, *i.e.*, it is a palindrome or not.

Example 3.13. Let $\mathbf{A} = (\{a, b\}, \iota, \{0, 1\}, \{q_0, q_a, q_b\}, q_0, \delta)$ be a WK transducer such that $\delta(q_0, \binom{a}{\lambda}) = (q_a, 1)$, $\delta(q_0, \binom{b}{\lambda}) = (q_b, 1)$, $\delta(q_a, \binom{\lambda}{a}) = (q_0, 1)$, $\delta(q_b, \binom{\lambda}{b}) = (q_0, 1)$, $\delta(q_a, \binom{\lambda}{b}) = (q_0, 0)$ and $\delta(q_b, \binom{\lambda}{a}) = (q_0, 0)$. See Figure 5. In state q_0 the upper head reads an input letter and the output of the WK transducer is in this transition the letter $x = 1$, furthermore in state q_a, q_b the lower head reads an input letter and the output of the WK transducer in these transitions is the letter y: if the lower head reads the same letter which just read by the upper head in the previous transition, then the output letter is $y = 1$, in the other cases the output letter is $y = 0$. (Using the notations as before we have: $q_0 \in Q_1, q_a, q_b \in Q_2$.) If for an input DNA string $w \in V^*$ the mapping $\gamma(w) = z$ has only letters 1's, then w is a palindrome. If 0 is a letter of the output z, then w is not a palindrome.

For example, if the input string is $w_1 = abaaaba$, then $\gamma(w_1) = 1111111$ and if the input string is $w_2 = abbaba$, then $\gamma(w_2) = 111110$. This means w_1 is a palindrome and w_2 is not a palindrome.

In fact, by measuring the longest even prefix of the output built up by only 1's, *i.e.*, the maximal value of n such that $(11)^n$ is a prefix of the output, one gets the measure of how long palindrome border the input word has. Formally, for an input w it is the maximal length of v such that $w = vuv^{-1}$, where v^{-1} is the reversal of v. If w is a palindrome, then $n = \left\lfloor \frac{|w|}{2} \right\rfloor$, *i.e.*, the integer part of the halflength of w.

Proposition 3.14. *Let* $\mathbf{A} = (V, \iota, T, Q, q_0, \delta)$ *be a WK transducer which uses its reading heads alternately. Then* $\mathbf{A}$ *has a bipartite graph.*

Proof. The set of states is partitioned to Q_1 and Q_2 (as defined in Lem. 2.6). Moreover, in these transducers, if $\delta(q, a) = p$, then either $q \in Q_1$, $p \in Q_2$ or $p \in Q_1$, $q \in Q_2$. □

Although the WK transducers of this subsection are not WK transducers with marked output by default, since it is clearly known that the even/odd positions of the output are written by transitions using one and the other reading head respectively. We can write this fact more formally in the following way. Remember, that Q_1 is the subset of states Q where the first reading head is allowed to move.

Proposition 3.15. *Let* γ *be a mapping defined by the WK transducer* $\mathbf{A} = (V, \iota, T, Q, q_0, \delta)$ *that uses its reading heads alternately. Further, let* T'*, the mapping* γ' *and the homomorphism* h *be defined as in Theorem 3.3. Although* h *is not injective,* $\gamma'(w)$ *can be determined from* $\gamma(w)$ *based on the fact whether* $q_0 \in Q_1$ *or not (without actually computing the WK transducer* $\mathbf{A}'$ *with marked output).*

Proof. If $q_0 \in Q_1$, then the first letter of the output $h(w)$ is written in a transition by the first head, and thus, the corresponding first letter of $\gamma'(w)$ is in T_1 (overlined). Then based on the bipartite graph of the transducer, each even position of $\gamma'(w)$ is underlined and all odd positions are overlined.

In the case $q_0 \notin Q_1$, *i.e.*, $q_0 \in Q_2$, every odd position of $\gamma'(w)$ is written by a letter from the underlined output alphabet T_2 and each even position with an overlined letter. □

In fact, by partitioning the positions of the output letters, as odd and even positions, by using the alphabetical morphisms $h_o : T_1 \to T$ and $h_e : T_2 \to T$ if $q_0 \in Q_1$ ($h_o : T_2 \to T$ and $h_e : T_1 \to T$ if $q_0 \in Q_2$, respectively), they are bijections.

In the case of alternately reading heads, Theorem 2.8 can be applied and, in this special case, we can give the algorithm of how the prefix of the output can be constructed from the prefix and suffix of the input.

There are two versions of this transducer depending on which head starts reading the input. If the upper head starts reading, we say it is the case UL; and if the lower head stars reading, we say it is the LU case. Let $w = w_1 w_2 ... w_n \in V^*$ be an input word and let the output be $z = \gamma(w)$ with $z = z_1 z_2$ for some $z_1, z_2 \in T^*$. In the UL case the $z_1 = \gamma(w_1)\gamma(w_n)\gamma(w_2)\gamma(w_{n-1})....$ Thus, basically, the prefix of the output corresponds to the prefix and the suffix of the input letter by letter alternately: first one letter from the prefix, then one letter from the suffix, etc. In the LU case the difference is that the lower head starts the reading so z_1 is created in the way $z_1 = \gamma(w_n)\gamma(w_1)\gamma(w_{n-1})\gamma(w_2)....$

When the heads are reading alternately we can see that the reading heads function $r(w)$ can be either in the form *ulululul...* or in the form *lulululu....* In the UL case the $p(w)$ will have the following structure: first we can see the odd numbers in increasing order from 1 to $\lceil |w|/2 \rceil$ and then the even numbers in decreasing order from $\lfloor |w|/2 \rfloor$ to 2. In the LU case we can see the even numbers from 2 to $\lfloor |w|/2 \rfloor$ and then the odd numbers in decreasing order from $\lceil |w|/2 \rceil$ to 1.

4. Further properties of WK transducers

In this section, first, we further analyze the functions p and r associated to a WK transducer. We start this section by an example.

Example 4.1. Let $w = abbaa$ be an input of a WK transducer $\mathbf{A}$.

1. When only the upper head reads: $r(w) = uuuuu$ and $p(w) = 1, 2, 3, 4, 5$.
2. When only the lower head reads: $r(w) = lllll$ and $p(w) = 5, 4, 3, 2, 1$.
3. In case of alternately reading heads: $r_1(w) = ululu$ and $p_1(w) = 1, 3, 5, 4, 2$ or $r_2(w) = lulul$ and $p_2(w) = 2, 4, 5, 3, 1$.
4. An arbitrary WK transducer could have the functions r and p such that they assign the following values to w: $r(w) = uulul$ and $p(w) = 1, 2, 4, 5, 3$.

One may ask what are the possible values of the functions r and p that can occur for our WK transducers. Let us answer this question.

Theorem 4.2. *Let $m \in \{u, l\}^*$ be an arbitrary word. There exists a WK transducer $\mathbf{A} = (V, \iota, T, Q, q_0, \delta)$ and an input word $w \in V^*$ such that the associated function, reading heads r assigns m to w, i.e., $r(w) = m$.*

Proof. We prove the statement by construction. Let $m = m_1...m_n$ with $m_1, ..., m_n \in \{u, l\}$ be given, where the length of m is $|m| = n$. Let $\mathbf{A} = (\{a\}, \iota, \{b\}, \{q_0, q_1, ..., q_n\}, q_0, \delta)$ and $w = a^n$ such that there is only 1 transition from each state: from q_i only to q_{i+1} $(i < n)$ and from q_n only to q_n itself. Further by Lemma 2.6, we know that $Q = Q_1 \cup Q_2$ where all transitions in states of Q_1 are defined only for the upper head and for all states in Q_2 transitions are defined only for the lower head. Now, set $Q_1 = \{q_j \mid m_{j+1} = u\}$, and $Q_2 = Q \setminus Q_1$. Clearly, when w is processed by $\mathbf{A}$, in the first step the upper head is used if and only if $q_0 \in Q_1$ that is if and only if $m_1 = u$, etc. Consequently, it is easy to see that $r(w) = m$. □

The function processing order is onto and one-to-one, *i.e.*, every value of the target $\{1, 2, ..., n\}$ is used exactly once for the given input word $w = w_1 w_2 ... w_n$ (with $|w| = n$). In fact, it is a permutation of these values (*i.e.* $p(w) \in S_n$ where S_n is a symmetric group, the set of all permutations with n element), but not every permutation can occur for WK transducers. For being a processing order function output $p(w)$ for an input word w, the permutation should be in a special form according to the next theorem. We use $s_{i,j}$ to denote the sequence of numbers from position i till position j of a permutation s (where $i < j$). Further, we use $smin_{i,j}$ to denote the smallest number in the sequence of numbers $s_i, ..., s_j$. (Furthermore $smin_{i,i} = s_i$ that is when we have only one element s_i.)

Theorem 4.3. *Let $s = s_1...s_n \in S_n$ be an arbitrary permutation. There exists a WK transducer $\mathbf{A}$ and an input word $w = w_1...w_n$, where the remaining subword of the input w is denoted by w' with $|w'| = l+1$ and $w' = w_k...w_{k+l}$. The processing order function may assign s to w, i.e. $p(w) = s$ if and only if for every remaining subword w' either $s_k = smin_{k,k+l}$ or $s_{k+l} = smin_{k,k+l}$.*

Proof. First let $\mathbf{A} = (V, \iota, T, Q, q_0, \delta)$ be a WK transducer and $w = w_1...w_n \in V^*$ such that the assigned processing order function $p(w) = s$. It means that for every $i = 1, ..., n$ we have $p_i = s_i$. Since the transducer is 1-limited and deterministic we know that at first step either the upper head reads the first letter of w hence the configuration (w, q_0, λ) will be followed by $(w_2....w_n, q, x)$ where $(q, x) = \delta(q_0, \binom{w_1}{\lambda})$ and $p_1 = 1$; or the lower head reads the last letter of w hence the configuration (w, q_0, λ) will be followed by $(w_1....w_{n-1}, q, x)$ where $(q, x) = \delta(q_0, \binom{\lambda}{w_n})$ and $p_n = 1$ for some state $q \in Q$ and output letter $x \in T$. The minimum of the numbers $1, ..., n$ denoted by $smin_{1,n}$ is 1 and if the upper head starts the readings then $s_1 = p_1 = 1 = smin_{1,n}$ and if the lower head starts the reading then $s_n = p_n = 1 = smin_{1,n}$. At step i (where $i = 2, ..., n$) we have a remaining word $w' = w_k...w_{k+l}$ with the length $|w'| = n - i + 1$ and in the computation, the configuration (w', q', z) can be followed by either $(w_{k+1}...w_{k+l}, q'', zy)$ where $(q'', y) = \delta(q', \binom{w_k}{\lambda})$ by reading with the upper head the first letter of w' or $(w_k...w_{k+l-1}, q'', zy)$ where $(q'', y) = \delta(q', \binom{\lambda}{w_{k+l}})$ by reading the last letter of w' with the lower head, for some states $q', q'' \in Q$ and output letter $y \in T$. If the upper head reads w_k then $p_k = i$ and if the lower head reads w_{k+l} then $p_{k+l} = i$. Therefore $s_k = p_k = i$ or $s_{k+l} = p_{k+l} = i$ and since $smin_{k,k+l} = i$ we proved the first direction of the statement.

Second, we prove the existence of $\mathbf{A}$ by construction. Let $\mathbf{A} = (V, \iota, T, Q, q_0, \delta)$ be a WK transducer and $w = w_1...w_n \in V^*$ be an input word of $\mathbf{A}$ where the transition δ can be constructed based on s in the following way. We assume based on the second part of the theorem that either $s_1 = smin_{1,n}$ (first case) or $s_n = smin_{1,n}$ (second case). Hence at step 1, if $s_1 = smin_{1,n}$ then the configuration (w, q_0, λ) will be followed by $(w_2....w_n, q, x)$ where $(q, x) = \delta(q_0, \binom{w_1}{\lambda})$ and if $s_n = smin_{1,n}$ then the configuration (w, q_0, λ) will be followed by $(w_1....w_{n-1}, q, x)$ where $(q, x) = \delta(q_0, \binom{\lambda}{w_n})$ for some state $q \in Q$ and output letter $x \in T$. In the first case we have $p_1 = 1$ and in the second case we have $p_n = 1$. At the i-th step (where $i = 2, ..., n$) we have a remaining word $w' = w_k....w_{k+l}$ with length $l = n - i + 1$ moreover $smin_{k,k+l} = i$ and we know from the theorem that either $s_k = smin_{k,k+l}$ (first case) or $s_{k+l} = smin_{k,k+l}$. In the computation, the configuration (w', q'', z') will be followed by either (in the first case) $(w_{k+1}...w_{k+l}, q'', z'y)$ where $(q'', y) = \delta(q', \binom{w_k}{\lambda})$ or (in the second case) $(w_k...w_{k+l-1}, q'', z'y)$ where $(q'', y) = \delta(q', \binom{\lambda}{w_{k+l}})$ for some states $q', q'' \in Q$ and output letter $y \in T$. In the first case we have $p_k = i$ and in the second case we have $p_{k+l} = i$. □

Further, we see how the functions processing order and reading heads relates to each other.

4.1. Producing $p(w)$ from $r(w)$

Algorithm 1 gives the formal description of the process of producing the string value of the processing order function from the given string value of the reading heads function on an input word w for a given WK transducer.

Algorithm 1. *Computing the processing order from reading heads.*

```
input: $r$    /*the string value of the reading heads function on $w$ for a given WK transducer $\mathbf{A}$*/
output: $p$    /*the string value of the processing order function on $w$ for the WK transducer $\mathbf{A}$*/
  let $n = |w|$
  let $first = 1$
  let $last = n$
  for $i = 1..n$ do
    if $(r_i = u)$
        then
            let $p_{first} = i$
            let $first = first + 1$
        else      /* in case $(r_i = l)$ */
            let $p_{last} = i$
            let $last = last - 1$
```

4.2. Producing $r(w)$ from $p(w)$

Algorithm 2 gives the formal description of the method of producing the string value of the reading heads function from the given string value of the processing order function on an input word w for a given WK transducer.

If $p(w)$ is known for an input word w for a given WK transducer, then we can create two $r(w)$ string values because it is undecidable using only these data which head read the last letter. Since Algorithm 2 at first checks if the certain p_i value is equal to u, it means the output r will contain a u at the position of the last letter hence it gives the solution when it was read by the upper head.

Algorithm 2. *Computing the reading heads from processing order.*

```
input: $p$    /*the string value of the processing order function on $w$ for a given WK transducer $\mathbf{A}$*/
output: $r$    /*the string value of the reading heads function on $w$ for the WK transducer $\mathbf{A}$*/
  let $n = |w|$
  let $first = 1$
  let $last = n$
  for $i = 1..n$ do
    if $(p_{first} = i)$
        then
            let $r_i = u$
            let $first = first + 1$
    if $(p_{last} = i)$
        then
            let $r_i = l$
            let $last = last - 1$
```

If an arbitrary $r(w)$ is given then we can always create $p(w)$ without any restrictions. The order of the heads can be changed at any moment of the process depending on the given WK transducer and on its input as we shown in Theorem 4.2. However, the other case requires a little investigation since processing an input by reading it from the extremes implies that the numbers cannot be in an arbitrary order, as we have shown in Theorem 4.3.

4.3. Control word of the computation

An important language associated to a Watson–Crick automaton was defined in [23]: taking into account the transitions, not the recognized string. They considered a labeling on the edges of the graph of a Watson–Crick finite automaton $\mathbf{M}$, *i.e.*, to every transition of the automaton a label is assigned. Then the control word of a computation was given by the sequence of labels of transitions of the computation. The languages and language classes of control words of accepting computations were defined and analyzed based on this definition for variants of the automata. While these control word languages were extensively studied in [23] for traditional WK automata, so far they have not been studied for $5' \to 3'$ WK automata. Here, we define and use the control word language of the computations by deterministic sensing $5' \to 3'$ WK automata. In fact, deterministic 1-limited $5' \to 3'$ sensing WK automata and our WK transducers are in a close correspondence based on control words as we formalize below.

Definition 4.4. Let $\mathbf{A}$ be a deterministic $5' \to 3'$ sensing WK automaton or a WK transducer. Let Lab be the alphabet (set) of labels and let us assign a *label* to each transition of $\mathbf{A}$, *i.e.*, let the graph of the automaton be edge labelled. Let w be an input, then the sequence of labels of the transitions of the computation on w is called the *control word* of the computation, it is in fact, a word in Lab^*.

Proposition 4.5. *Let $\mathbf{A} = (V, \iota, T, Q, q_0, \delta)$ be a $5' \to 3'$ WK transducer in the following way: let the set of labels on the transitions be T such that on transition $(p, x) \in \delta\left(q, \binom{a}{b}\right)$ the label is exactly x. In this way the control word of a computation of $\mathbf{A}$ from the configuration (w, q_0, λ) to the configuration (λ, q, z) is exactly the output word z.*

Considering deterministic 1-limited all-final $5' \to 3'$ sensing WK automata $\mathbf{M} = (V, \iota, Q, q_0, \delta, Q)$ and the label set Lab such that there is a unique label $x \in Lab$ assigned for each transition $p \in \delta\left(q, \binom{a}{b}\right)$. The control words of the computations on any input word w can be observed as the words over Lab which are exactly the same as the output words of the $5' \to 3'$ Mealy WK transducer $\mathbf{A} = (V, \iota, Lab, Q, q_0, \delta')$ working on input w where δ' is defined as follows: $(p, x) \in \delta'(\left(q, \binom{a}{b}\right)$ if and only if $p \in \delta\left(q, \binom{a}{b}\right)$ and $x \in Lab$ is assigned to this transition.

Proof. The first statement of the proposition is obvious, while the second statement, the connection between the control words and control word language of deterministic 1-limited all-final $5' \to 3'$ WK automata and deterministic 1-limited $5' \to 3'$ WK transducers follow from the first statement and the construction shown in the proposition. □

Based on the previous proposition, WK transducers can also be connected to the concept of computing by observing ([2]) as the transition sequence in a bio-inspired system can be connected to a sequence of observations that can be done outside of the system during the process it computes. Actually, the WK transducers can make explicit the observation as the output of the computation.

Special subtypes of WK transducers are also presented: at WK transducers with marked-output it is explicitly given which of the heads moved in which transition. In some other cases one also can identify which of the reading heads are used in which transition. Here we can see another approach related to this issue. By using unique labels for every transition it is clear again that the output letter (label) identifies the reading head that moves during this transition.

In the next section we analyze transducers in which we can infer from the output which of the heads was moving in each transition.

5. Equivalence relation among the states of WK transducers

In this section we investigate equivalence relation among the states of a WK transducer and among the states of two WK transducers (with marked-output).

Definition 5.1. Let $\mathbf{A_1} = (V, \iota, T, Q, p_0, \delta)$ and $\mathbf{A_2} = (V, \iota, T, Q', q_0, \delta')$ be two not necessarily different WK transducers with common input and output alphabets. We say that $p \in Q$ and $q \in Q'$ are *distinguishable with word* $w \in V^*$, if we consider the configurations $(w, p, \lambda) \Rightarrow^* (\lambda, p', z_1)$ and $(w, q, \lambda) \Rightarrow^* (\lambda, q', z_2)$, then $z_1 \neq z_2$ holds.

If two states are distinguishable, intuitively, we may distinguish them by input words which are processed without state repetition, this fact can be formalised as follows.

Theorem 5.2. *Let $\mathbf{A} = (V, \iota, T, Q, p_0, \delta)$ be a WK transducer with marked-output. Let $n = |Q|$ and $p, q \in Q$. If p and q are distinguishable, then there is a word w of length at most $n - 1$ such that it distinguishes p and q.*

Proof. The proof goes by induction. For every non-negative integer k we define a relation η_k on the set of states Q of the given WK transducer as follows: $(p, q) \in \eta_k$ if and only if p and q cannot be distinguished by any words of length at most k. Denote c_k the number of η_k-classes. Naturally, $(p, q) \in \eta_0$ for every $p, q \in Q$, thus $c_0 = 1$, there is only one class.

Let us assume that η_k is known. Let us compute η_{k+1}. If $(p, q) \notin \eta_k$, then $(p, q) \notin \eta_{k+1}$. If $(p, q) \in \eta_k$, then $(p, q) \in \eta_{k+1}$ if the following condition holds for every $w \in V^*$ with $|w| = k$. Consider $w' = w_1 a b w_2 \in V^*$ where $w = w_1 w_2$ such that w_1 is read by the first head and w_2 is read by the second head, furthermore $|a| + |b| = 1$ (thus $|w'| = k+1$). Then the computation on w' from the states p and q go: $(w', p, \lambda) \Rightarrow^* (ab, p', z) \Rightarrow (\lambda, p'', zx)$ and $(w', q, \lambda) \Rightarrow^* (ab, q', z) \Rightarrow (\lambda, q'', zx)$.

In this way it is clear that η_{k+1} is a refinement of the classification η_k (allowing longer input there can be more classes of states).

There is a value $m \in \mathbb{N}$ such that $m \leq n - 1$ and $\eta_m = \eta_{m+1} = \eta_{m+i}$ for every $i \in \mathbb{N}$, because the set Q is finite. (If there would be no such value m, then in each step of the refinement we obtain more and more classes, and after n steps some classes with only one state would be divided which is clearly impossible.) Denote $\eta = \eta_m$ this final equivalence relation among the states.

The method to get this final classification also confirms the proof. □

The next theorem allows us to compare the states of two transducers.

Theorem 5.3. *Let $\mathbf{A_1} = (V, \iota, T, Q, p_0, \delta)$ and $\mathbf{A_2} = (V, \iota, T, Q', q_0, \delta')$ be two different WK transducers with marked-output, with common input and output alphabets. Let $\mathbf{A} = (V, \iota, T, Q \cup Q', p_0, \delta'')$, where δ'' is defined in the following way: $\delta''(q, a) = \delta(q, a)$ if $q \in Q$ and $\delta''(q, a) = \delta'(q', a)$ if $q \in Q'$. Let $p \in Q$ and $q \in Q'$. If p and q are distinguishable, then there is a word w of length at most $|Q| + |Q'| - 1$ such that it distinguishes p and q.*

Proof. The proof is a direct consequence of the definition of WK transducer $\mathbf{A}$ and Theorem 5.2. □

Definition 5.4. Let $\mathbf{A_1} = (V, \iota, T, Q, p_0, \delta)$ and $\mathbf{A_2} = (V, \iota, T, Q', q_0, \delta')$ be two not necessarily different WK transducers with common input and output alphabets. We say that $p \in Q$ and $q \in Q'$ are *equivalent* if they are not distinguishable with any word $w \in V^*$.

By Theorems 5.2 and 5.3 if one wants to know whether two states are equivalent, then it is enough to check their behavior for input words with at most a certain length.

Theorem 5.5. *Let $\mathbf{A_1} = (V, \iota, T, Q, p_0, \delta)$ and $\mathbf{A_2} = (V, \iota, T, Q', q_0, \delta')$ be two WK transducers with marked-output, with common input and output alphabets. $\mathbf{A_1}$ is equivalent to $\mathbf{A_2}$ if and only if p_0 and q_0 are equivalent.*

Proof. Let γ_1 be the mapping defined by $\mathbf{A_1}$ and γ_2 be defined similarly by $\mathbf{A_2}$. By definition, $\mathbf{A_1}$ and $\mathbf{A_2}$ are equivalent if γ_1 identical to γ_2. If p_0 is not distinguishable from q_0, then for every word $w \in V^*$ $\gamma_1(w) = \gamma_2(w)$. □

6. Canonical WK transducers

Canonical, *i.e.*, minimal WK transducers will be defined for WK transducers with marked-output. In this section we deal with WK transducers such that every of their state is reachable from the initial states. (For each state q' there exists an input string w such that $(w, q_0, \lambda) \Rightarrow^* (\lambda, q', z)$ where q_0 is the initial state and z is the output of the computation on w.)

Lemma 6.1. *Let $\mathbf{A_1} = (V, \iota, T, Q, p_0, \delta)$ and $\mathbf{A_2} = (V, \iota, T, Q', q_0, \delta')$ be two WK transducers with marked-output and the same input and output alphabets, such that every of their state is reachable from the initial states, respectively. The WK transducers $\mathbf{A_1}$ and $\mathbf{A_2}$ are equivalent if and only if p_0 and q_0 are equivalent, moreover for every state in Q there is an equivalent state in Q' and for every state in Q' there is an equivalent state in Q.*

Proof. By Theorem 5.5 it is clear that $\mathbf{A_1}$ and $\mathbf{A_2}$ are equivalent if their initial states are equivalent. Let us prove the other part of the statement by contradiction. Without loss of generality, let us assume that there is a state $p' \in Q$ such that there is no equivalent state in Q' with p'; but $\mathbf{A_1}$ and $\mathbf{A_2}$ are equivalent. By the assumption that p' can be reached from p_0, there is an input word $w \in V^*$ such that: $(w, p_0, \lambda) \Rightarrow^* (\lambda, p', z)$ and $(w, q_0, \lambda) \Rightarrow^* (\lambda, q', z)$ for a state $q' \in Q'$. Since q' and p' are not equivalent, there is a word u such that $(u, p', \lambda) \Rightarrow^* (\lambda, p'', z_1)$, $(u, q', \lambda) \Rightarrow^* (\lambda, q'', z_2)$ and $z_1 \neq z_2$. Then $(w_1 u w_2, p_0, \lambda) \Rightarrow^* (\lambda, p'', z z_1)$, $(w_1 u w_2, q_0, \lambda) \Rightarrow^* (\lambda, q'', z z_2)$ with some $w_1, w_2 \in V^*$ such that $w = w_1 w_2$. This contradicts to the fact that $\mathbf{A_1}$ and $\mathbf{A_2}$ are equivalent. □

Definition 6.2. Let $\mathbf{A} = (V, \iota, T, Q, p_0, \delta)$ be a WK transducer and γ be the mapping defined by $\mathbf{A}$. If we consider all the WK transducers which induce mapping γ and $\mathbf{A}$ has the minimal number of states among them, then we say that $\mathbf{A}$ is *minimal* (or *canonical*).

In the following theorem we will use the notation η which we have defined in the proof of Theorem 5.2.

Theorem 6.3. *Let $\mathbf{A_1} = (V, \iota, T, Q_1, p_0, \delta)$ be a WK transducer with marked output, γ_1 be the mapping defined by $\mathbf{A_1}$. Let $\mathbf{A_2} = (V, \iota, T, Q_2, p_0', \delta')$. If $\mathbf{A_2}$ is a minimal WK transducer and equivalent with $\mathbf{A_1}$, then the number of states of $\mathbf{A_2}$ is the same as the number of classes in η for $\mathbf{A_1}$.*

Proof. The proof consists of two parts. First we show that there is a WK transducer $\mathbf{A_2}$ that is equivalent with $\mathbf{A_1}$ and has the same number of states as the number of classes in η for $\mathbf{A_1}$.

Denote $\hat{p}$ the equivalence class of η containing p, *i.e.*, $\hat{p} = \{q | (p, q) \in \eta\}$. Let $Q_2 = \{\hat{p} | p \in Q_1\}$.

Let $\mathbf{A_2} = (V, \iota, T, Q_2, \hat{p_0}, \delta')$ be a WK transducer such that δ' is defined in the following way: $\delta'(\hat{p}, a) = (\hat{q}, x)$ if $\delta(p, a) = (q', x)$ where $q' \in \hat{q}$. (Note that $|Q_2| = |Q_1|$ when $\mathbf{A_1}$ is minimal.)

It is clear by the construction and the equivalence relation among the states that $\mathbf{A_2}$ defines the same mapping γ as $\mathbf{A_1}$. Moreover in $\mathbf{A_2}$ every class is represented by a unique state.

The second part of the proof goes by contradiction. We prove that $\mathbf{A_2}$ is minimal. Let us assume that $\mathbf{A_3} = (V, \iota, T, Q_3, p_0, \delta)$ is a minimal WK transducer and equivalent with $\mathbf{A_1}$, but $|Q_3| > |Q_2|$. Then it is a contradiction, since $\mathbf{A_2}$ is also equivalent with $\mathbf{A_1}$ and has smaller number of states.

Consider now the case when $\mathbf{A_3}$ has less states than $\mathbf{A_2}$, *i.e.*, $|Q_3| < |Q_2|$. This means that there is at least one state in $q_2 \in Q_2$ such that there is no equivalent state to q_2 in Q_3. Then $\mathbf{A_3}$ cannot be equivalent to $\mathbf{A_2}$ and to $\mathbf{A_1}$ by Lemma 6.1. Thus any minimal WK transducer that is equivalent to $\mathbf{A_1}$ has the same number of states that is the number identical to the number of classes of η for $\mathbf{A_1}$. □

7. Conclusions

Deterministic one-limited $5' \to 3'$ Mealy and Moore type Watson–Crick transducers were investigated. We have shown that for each Moore type WK transducer, there is an equivalent Mealy type WK transducer in the sense that the same mapping is realized by them and vice versa. Based on that, we mostly detailed only the Mealy type variants. In contrast, when the number of states is limited to one, *i.e.*, considering the stateless variants, it is shown that the Moore type transducers are more restricted than the Mealy type transducers. In fact, stateless Moore type WK transducers cannot do more than stateless Moore automata (with only one reading head). On the other hand, stateless Mealy type WK transducers can do everything that stateless Mealy automata (with one reading head) can do, moreover, they can also realize some other mappings. In fact, they can already do reversing the input word (that is defined as a hard task in [1] with – finite state – transducers). We have restricted our study to 1-limited and deterministic WK transducers, *i.e.*, in every configuration with nonempty input they read exactly one letter (by one of the two heads) and write exactly one letter on the output tape. Consequently, it is shown that at every state exactly one head can read (and move), *i.e.*, in all possible transitions from that state the same head can read a letter. It was shown that the mappings realized by this model has length preserving property and also a kind of border-to-prefix property in the sense that the prefix of the output depends only on a combination of the prefix and the suffix of the input.

By releasing the 1-limited condition, *i.e.*, allowing to read and/or to write longer strings or the empty word in a transition a generalization of the studied models can be understood which generalize also the streaming transducers (because of the two reading heads). Non-deterministic variants can also be interesting. While variants with non-injective WK-complementarity relation can also be analyzed, especially in the case when both reading heads must read their strands entirely.

We have also presented canonical $5' \to 3'$ WK transducers for variants with marked output. In these transducers, the output symbol does identify the reading head used in that transition. We have used the functions processing order and reading head to analyze WK transducers. Extension of the study of a larger class of WK transducers is also a task of a future research. We believe that a much larger class of mapping can be defined and analyzed by automata using our approach, as Lin and $2detLin$ are classes including much more interesting languages than Reg.

Acknowledgements. Some of the results of this paper were already presented at NCMA 2019: Eleventh Workshop on Non-Classical Models of Automata and Applications, Valencia, Spain (see [17]). The authors thank the comments of the reviewers, committee members and participants on the conference version. Comments of the anonymous reviewers of the journal version are also acknowledged. The work of the second author is supported by the EFOP-3.6.1-16-2016-00022 project. The project is co-financed by the European Union and the European Social Fund.

References

[1] R. Alur and P. Cerný, Expressiveness of streaming string transducers, in *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2010, December 15–18, 2010, Chennai, India* (2010) 1–12.

[2] M. Cavaliere, Computing by observing: A brief survey, in *Logic and Theory of Algorithms, 4th Conference on Computability in Europe, CiE 2008, Athens, Greece, June 15-20, 2008, Proceedings* (2008) 110–119.

[3] E. Czeizler, E. Czeizler, L. Kari and K. Salomaa, Watson-Crick automata: determinism and state complexity, in *10th International Workshop on Descriptional Complexity of Formal Systems, DCFS 2008, Charlottetown, Prince Edward Island, Canada, July 16–18, 2008*, edited by C. Câmpeanu and G. Pighizzini. University of Prince Edward Island (2008) 121–133.

[4] R. Freund, Gh. Păun, G. Rozenberg and A. Salomaa, Watson-Crick finite automata, in *DNA Based Computers, Proceedings of a DIMACS Workshop, Philadelphia, Pennsylvania, USA, June 23-25, 1997* (1997) 297–328.

[5] M.A. Harrison, Introduction to Formal Language Theory. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st ed. (1978).

[6] J.E. Hopcroft and J.D. Ullman, Introduction to Automata Theory, Languages and Computation. Addison-Wesley (1979).

[7] R. Kocman, B. Nagy, Z. Krivka and A. Meduna, A jumping 5' → 3' Watson-Crick finite automata model, in *Tenth Workshop on Non-Classical Models of Automata and Applications, NCMA 2018, Košice, Slovakia, August 21-22, 2018* (2018) 117–132.

[8] P. Leupold and B. Nagy, 5' → 3' Watson-Crick automata with several runs, in *Workshop on Non-Classical Models for Automata and Applications – NCMA 2009, Wroclaw, Poland, August 31–September 1, 2009. Proceedings* (2009) 167–180.

[9] P. Leupold and B. Nagy, 5' → 3' Watson-Crick automata with several runs. *Fundam. Inform.* **104** (2010) 71–91.

[10] R. Loukanova, Linear context free languages, in *ICTAC 2007, Proc.*, edited by C. Jones, Z. Liu and J. Woodcock. Vol. 4711 of *LNCS*. Springer, Heidelberg (2007) 351–365.

[11] G.H. Mealy, A method for synthesizing sequential circuits. *Bell Syst. Tech. J.* **34** (1955) 1045–1079.

[12] B. Nagy, On 5' → 3' sensing Watson-Crick finite automata, in *DNA Computing, 13th International Meeting on DNA Computing, DNA13, Memphis, TN, USA, June 4–8, 2007*, Revised Selected Papers, Lecture Notes in Computer Science 4848, Heidelberg. Springer (2008) 256–262.

[13] B. Nagy, On a hierarchy of $5' \rightarrow 3'$ sensing WK finite automata languages, in *CiE 2009, Computability in Europe 2009: Mathematical Theory and Computational Practice, Abstract Booklet, University of Heidelberg, Germany* (2009) 266–275.

[14] B. Nagy, A class of 2-head finite automata for linear languages. *Triangle* **8** (2012) 89–99.

[15] B. Nagy, On a hierarchy of $5' \rightarrow 3'$ sensing Watson-Crick finite automata languages. *J. Log. Comput.* **23** (2013) 855–872.

[16] B. Nagy, $5'{\rightarrow}3'$ Watson-Crick pushdown automata. *Inf. Sci.* **537** (2020) 452–466.

[17] B. Nagy and Z. Kovács, On simple 5' → 3' sensing Watson-Crick finite-state transducers, in *Eleventh Workshop on Non-Classical Models of Automata and Applications, NCMA 2019, Valencia, Spain, July 2-3, 2019* (2019) 155–170.

[18] B. Nagy and F. Otto, Two-head finite-state acceptors with translucent letters, in *SOFSEM 2019, Proc.*, edited by B. Catania, R. Královič, J. Nawrocki and G. Pighizzini. Lecture Notes in Computer Science 11376, Springer, Heidelberg (2019) 406–418.

[19] B. Nagy and F. Otto, Linear automata with translucent letters and linear context-free trace languages. *RAIRO-ITA: Theor. Inf. Appl.* **54** (2020) 3.

[20] B. Nagy and S. Parchami, On deterministic sensing $5' \rightarrow 3'$ Watson-Crick finite automata – a full hierarchy in 2detLIN. *Acta Inf.* **58** (2021) 153–175.

[21] B. Nagy, S. Parchami and H.M.M. Sadeghi, A new sensing 5' → 3' Watson-Crick automata concept, in *Proceedings 15th International Conference on Automata and Formal Languages, AFL 2017, Debrecen, Hungary, September 4-6, 2017., EPTCS 252* (2017) 195–204.

[22] S. Parchami and B. Nagy, Deterministic sensing 5' → 3' Watson-Crick automata without sensing parameter, in *Unconventional Computation and Natural Computation – 17th International Conference, UCNC 2018, Fontainebleau, France, June 25–29, 2018, Proceedings*. Lecture Notes in Computer Science 10867. Springer, Heidelberg (2018) 173–187.

[23] Gh. Păun, G. Rozenberg and A. Salomaa, DNA Computing – New Computing Paradigms, Texts in Theoretical Computer Science. An EATCS Series. Springer (1998).

[24] G. Rozenberg and A. Salomaa (Eds.), Handbook of Formal Languages, Volume 1: Word, Language, Grammar. Springer (1997).

[25] J.M. Sempere, A note on the equivalence and complexity of linear grammars. *Grammars* **6** (2003) 115–126.

[26] J.M. Sempere, A representation theorem for languages accepted by Watson-Crick finite automata. *Bull. EATCS* **83** (2004) 187–191.

[27] S. Yu, Regular languages, in *Handbook of Formal Languages, Volume 1: Word, Language, Grammar*, edited by G. Rozenberg and A. Salomaa. Springer (1997) 41–110.