Acta Informatica manuscript No.
(will be inserted by the editor)

A Jumping $5' \to 3'$ Watson-Crick Finite Automata Model

Radim Kocman · Zbyněk Křivka · Alexander Meduna · Benedek Nagy

Received: date / Accepted: date

Abstract Jumping finite automata and sensing $5' \to 3'$ Watson-Crick finite automata are finite-state models of computation which allow to process the input word not only in the strictly left-to-right manner. In this paper a new combined model of them is presented. The accepting power of the new model is studied and compared with the original models and also other well-known language families. Furthermore, the paper investigates changes in the accepting power when commonly studied restrictions from Watson-Crick finite automata, e.g., all states are final, are applied on this combined model. At the end, the paper presents a comprehensive hierarchy of all related language families.

Keywords jumping automata · Watson-Crick automata · DNA computer · 2-head automata · discontinuous information processing · formal languages

1 Introduction

In recent years, research in formal language theory takes interest in models that process inputs or generate outputs in non-conventional ways compared to classical models of automata and grammars. Due to the direction of the ongoing development in computer science, the main focus is often on models that process information discontinuously and work in a parallel way. In this paper, we focus our attention on two groups of such models: *jumping finite automata* and *Watson-Crick finite automata*. To be more precise, our main focus is on their specific variations which have multiple heads working in parallel that non-conventionally

An early stage of this research was presented in NCMA 2018, see [12].

R. Kocman, Z. Křivka, A. Meduna
Centre of Excellence IT4Innovations, Faculty of Information Technology,
Brno University of Technology, Božetěchova 2, Brno, Czech Republic
E-mail: {ikocman,krivka,meduna}@fit.vutbr.cz

B. Nagy
Department of Mathematics, Faculty of Arts and Sciences,
Eastern Mediterranean University, Famagusta, North Cyprus, Mersin-10, Turkey
E-mail: nbenedek.inf@gmail.com

process the input sequence/string. Traditionally, when an automaton model utilizes several heads, either each head works on its own tape, or all heads read the same input string in a symbol-by-symbol left-to-right way. In contrast, there are also well-established models of grammars that generate strings in a parallel way, but this process is usually very different than the reading with several heads. In the grammars, the sentential form is repeatedly rewritten on several places at once until the process creates the final string. The studied multi-head variations of finite automata models have their behavior set somewhere between the mentioned traditional models. They utilize several heads, but these heads cooperate on a single tape to process the single input string. Therefore, every symbol in the input is read only once, and the heads do not work in the traditional symbol-by-symbol left-to-right way.

To give a better insight into this study, let us briefly introduce both groups of mentioned models. Jumping finite automata were first introduced relatively recently in [16,17]. Their jumping concept is in its core focused on discontinuous information processing. In essence, a jumping finite automaton works just like a classical finite automaton except it does not read the input string in a symbol-by-symbol left-to-right way. After the automaton reads a symbol, the head can jump over (skip) a portion of the tape in either direction. Once an occurrence of a symbol is read on the tape, it cannot be re-read again later. Generally, these models can very easily define even some non-context-free languages if the order of symbols is unimportant for the language. On the other hand, the resulting language families of these models are usually incomparable with the classical families of regular, linear, and context-free languages. In the following years, this jumping mechanism has proven to be a clever addition into the model of finite automata that delivers new interesting results and captures novel families of languages. Several follow-up series of papers have spawned from this idea that further explore its possibilities in various different ways: a continuing study of the initial models (see [2,6,7, 26]), a use of jumps in other classical models (see [8,13,15]), a different approach with more deterministic behavior (see [1,3–5]), and a use of jumps together with multiple heads (see [9–12]). If the jumping concept utilizes multiple heads, the heads can naturally jump on specific positions in the tape, and thus they can easily work on different places at once in parallel.

Watson-Crick (WK) finite automata are a more settled and already thoroughly studied group of biology-inspired models (see, e.g., [24]). In essence, a WK automaton also works just like a classical finite automaton except it uses a WK tape (i.e., double-stranded tape), and it has a separate head for each of the two strands in the tape. This is therefore a group of models that always naturally use two heads. The classical version of a WK automaton processes the input tape quite conventionally: each head works separately on its own strand of the tape, and both heads read the input in a traditional symbol-by-symbol left-to-right way. However, more recently, new variations of this model were introduced that process the input in non-conventional ways. In a $5' \to 3'$ *WK automaton* (see [18–21]), both heads read their specific strand in the biochemical $5'$ to $3'$ direction. In a computing point of view, this means that they read the double strand sequence in opposite directions. Furthermore, a $5' \to 3'$ WK automaton is *sensing* if the heads sense that they are meeting each other, and the processing of the input ends if for all pairs of the sequence one of the letters is read. The sensing $5' \to 3'$ WK automata generally accept the family of linear languages. This concept is also studied further in sev-

eral follow-up papers that explore alternative definitions and combinations with different mechanics (see [22, 23]).

Even though that these two groups are significantly different in their original definitions, their newer models sometimes work in a very similar way. Both concepts are also not mutually exclusive in a single formal model. This paper defines *jumping* $5' \to 3'$ *WK automata*—a combined model of jumping finite automata and sensing $5' \to 3'$ WK automata—and studies their characteristics. We primarily investigate the accepting power of the model and also the effects of common restrictions on the model.

2 Preliminaries

This paper assumes that the reader is familiar with the theory of automata and formal languages (see [14, 27]). This section recalls only the crucial notions used in this paper.

For a set Q, $\mathrm{card}(Q)$ denotes the cardinality of Q, and 2^Q denotes the power set of Q. For an alphabet (finite nonempty set) V, V^* represents the free monoid generated by V under the operation of concatenation. The unit of V^* is denoted by ε. Members of V^* are called *strings*. Set $V^+ = V^* - \{\varepsilon\}$; algebraically, V^+ is thus the free semigroup generated by V under the operation of concatenation. For $x \in V^*$, $|x|$ denotes the length of x, and $\mathrm{alph}(x)$ denotes the set of all symbols occurring in x; for instance, $\mathrm{alph}(0010) = \{0, 1\}$. For $x \in V^*$ and $a \in V$, $|x|_a$ denotes the number of occurrences of a in x. The *Parikh vector* associated to a string $x \in V^*$ with respect to the alphabet $V = \{a_1, a_2, \ldots, a_n\}$ is $\Psi_V(x) = (|x|_{a_1}, |x|_{a_2}, \ldots, |x|_{a_n})$. For $L \subseteq V^*$ we define $\Psi_V(L) = \{\Psi_V(x) : x \in L\}$. For $x, y \in V^*$, the *shuffle* of x and y, denoted by $\mathrm{shuffle}(x, y)$, is defined as $\mathrm{shuffle}(x, y) = \{x_1 y_1 x_2 y_2 \cdots x_n y_n : x = x_1 x_2 \cdots x_n,\ y = y_1 y_2 \cdots y_n,\ x_i, y_i \in V^*,\ 1 \leq i \leq n,\ n \geq 1\}$. Let X and Y be sets; we call X and Y to be *incomparable* if $X \not\subseteq Y$, $Y \not\subseteq X$, and $X \cap Y \neq \emptyset$.

A *general grammar* or, more simply, a *grammar* is quadruple $G = (N, T, S, P)$, where N and T are alphabets such that $N \cap T = \emptyset$, $S \in N$, and P is a finite set of rules of the form $x \to y$, where $x, y \in (N \cup T)^*$ and $\mathrm{alph}(x) \cap N \neq \emptyset$. If $x \to y \in P$ and $u, v \in (N \cup T)^*$, then $uxv \Rightarrow uyv$ $[x \to y]$, or simply $uxv \Rightarrow uyv$. In the standard manner, extend $\Rightarrow$ to $\Rightarrow^n$, where $n \geq 0$; then, based on $\Rightarrow^n$, define $\Rightarrow^+$ and $\Rightarrow^*$. The language generated by G, $L(G)$, is defined as $L(G) = \{w \in T^* : S \Rightarrow^* w\}$. We recognize several special cases of grammars: G is a *context-sensitive grammar* if every $x \to y \in P$ satisfies $x = \alpha A \beta$ and $y = \alpha y \beta$ such that $A \in N$, $\alpha, \beta \in (N \cup T)^*$, and $y \in (N \cup T)^+$. G is a *context-free grammar* if every $x \to y \in P$ satisfies $x \in N$. G is a *linear grammar* if every $x \to y \in P$ satisfies $x \in N$ and $y \in T^* N T^* \cup T^*$. G is a *regular grammar* if every $x \to y \in P$ satisfies $x \in N$ and $y \in TN \cup T$. A language L is context-sensitive, context-free, linear, or regular if and only if $L = L(G)$, where G is a context-sensitive, context-free, linear, or regular grammar, respectively.

A *finite automaton* is a quintuple $A = (V, Q, q_0, F, \delta)$, where V is an input alphabet, Q is a finite set of states, $V \cap Q = \emptyset$, $q_0 \in Q$ is the initial (or start) state, and $F \subseteq Q$ is a set of final (or accepting) states. The mapping δ is a transition function. If $\delta \colon Q \times (V \cup \{\varepsilon\}) \to 2^Q$, then the device is nondeterministic; if $\delta \colon Q \times V \to Q$, then the automaton is deterministic. A string w is accepted by a

finite automaton if there is a sequence of transitions starting from q_0, ending in a state in F, and the symbols of the sequence yield w. A language is regular if and only if it can be recognized by a finite automaton.

Let **FIN**, **REG**, **LIN**, **CF**, and **CS** denote the families of finite, regular, linear, context-free, and context-sensitive languages, respectively. Moreover, let $\mathbf{FIN}_{\varepsilon\text{-inc}}$ denote the family of finite languages which contain the empty string.

2.1 Jumping Finite Automata

A *general jumping finite automaton* (see [16,17]), a *GJFA* for short, is a quintuple $M = (Q, \Sigma, R, s, F)$, where Q is a finite set of states, Σ is an input alphabet, $Q \cap \Sigma = \emptyset$, $R \subseteq Q \times \Sigma^* \times Q$ is finite, $s \in Q$ is the start state, and $F \subseteq Q$ is a set of final states. Members of R are referred to as rules of M. If $(p, y, q) \in R$ implies that $|y| \leq 1$, then M is a *jumping finite automaton*, a *JFA* for short. A configuration of M is any string in $\Sigma^* Q \Sigma^*$. The binary jumping relation, symbolically denoted by $\curvearrowright$, over $\Sigma^* Q \Sigma^*$, is defined as follows. Let $x, z, x', z' \in \Sigma^*$ such that $xz = x'z'$ and $(p, y, q) \in R$; then, M makes a *jump* from $xpyz$ to $x'qz'$, symbolically written as $xpyz \curvearrowright x'qz'$. In the standard manner, extend $\curvearrowright$ to $\curvearrowright^n$, where $n \geq 0$; then, based on $\curvearrowright^n$, define $\curvearrowright^+$ and $\curvearrowright^*$. The language accepted by M, denoted by $L(M)$, is defined as $L(M) = \{uv : u, v \in \Sigma^*,\ usv \curvearrowright^* f,\ f \in F\}$. We say that M accepts w if and only if $w \in L(M)$. M rejects w if and only if $w \in \Sigma^* - L(M)$.

Double-jumping modes for GJFAs were introduced in [9,10], which perform two single jumps simultaneously. Both jumps always follow the same rule, however, they are performed on two different positions on the tape and thus handle different parts of the input string. Additionally, these jumps cannot ever cross each other (i.e., the initial mutual order of reading positions is preserved during the whole accepting process). The specific double-jumping modes then assign one of the three jumping directions to each of the two jumps—(1) to the left, (2) to the right, and (3) in either direction. We omit the precise formal definition.

2.2 Watson-Crick Finite Automata

In this part we recall some well-known concepts of DNA computing and related formal language theory. Readers who are not familiar with these topics should read [24].

Let V be an alphabet and $\rho \subseteq V \times V$ be its complementary relation. For instance, $V = \{A, C, G, T\}$ is usually used in DNA computing with the Watson-Crick complementary relation $\{(T, A), (A, T), (C, G), (G, C)\}$. The strings built up by complementary pairs of letters are double strands (of DNA).

A *Watson-Crick finite automaton* (or shortly, a *WK automaton*) is a finite automaton working on a Watson-Crick tape, that is, a double-stranded sequence (or molecule) in which the lengths of the strands are equal and the elements of the strands are pairwise complements of each other: $\left[\begin{smallmatrix} a_1 \\ b_1 \end{smallmatrix}\right]\left[\begin{smallmatrix} a_2 \\ b_2 \end{smallmatrix}\right] \ldots \left[\begin{smallmatrix} a_n \\ b_n \end{smallmatrix}\right] = \left[\begin{smallmatrix} a_1 a_2 \ldots a_n \\ b_1 b_2 \ldots b_n \end{smallmatrix}\right]$ with $a_i, b_i \in V$ and $(a_i, b_i) \in \rho$ $(i = 1, \ldots, n)$. The notation $\left[\begin{smallmatrix} w_1 \\ w_2 \end{smallmatrix}\right]$ is used only for strings w_1, w_2 with equal length and satisfying the complementary relation ρ. The set of all double-stranded strings with this property is denoted by $\mathrm{WK}_\rho(V)$. For double-stranded strings for which these conditions are not necessarily satisfied,

the notation $\binom{w_1}{w_2}$ is used throughout the paper. Formally, a WK automaton is $M = (V, \rho, Q, q_0, F, \delta)$, where V, Q, q_0, and F are the same as in finite automata, $\rho \subseteq V \times V$ is a symmetric relation, and the transition mapping $\delta\colon (Q \times \binom{V^*}{V^*}) \to 2^Q$ in such a way that $\delta(q, \binom{w_1}{w_2})$ ($q \in Q$, $w_1, w_2 \in V^*$) is nonempty only for finitely many values of $(q, \binom{w_1}{w_2})$.

The elementary difference between finite automata and WK automata, besides the doubled tape, is the number of heads. WK automata scan each of the two strands separately with a unique head. In classical WK automata, the processing of the input sequence ends if all pairs of the sequence are read with both heads. There are also some restricted variations of WK automata which are widely used in the literature (see, e.g., [24]):

- **N** : stateless, i.e., with only one state: if $Q = F = \{q_0\}$;
- **F** : all-final, i.e., with only final states: if $Q = F$;
- **S** : simple (at most one head moves in a step)
  $\delta\colon (Q \times (\binom{V^*}{\{\varepsilon\}} \cup \binom{\{\varepsilon\}}{V^*})) \to 2^Q$;
- **1** : 1-limited (exactly one letter is being read in a step)
  $\delta\colon (Q \times (\binom{V}{\{\varepsilon\}} \cup \binom{\{\varepsilon\}}{V})) \to 2^Q$.

Further variations such as **NS**, **FS**, **N1**, and **F1** WK automata can be identified in a straightforward way by using multiple constraints.

In $5' \to 3'$ *WK automata* (see [18–21,23]), both heads start from the $5'$ end of the appropriate strand. Physically/mathematically and from a computing point of view they read the double-stranded sequence in opposite directions, while biochemically they go to the same direction. A $5' \to 3'$ WK automaton is *sensing* if the heads sense that they are meeting (i.e., they are close enough to meet in the next step or there is a possibility to read strings at overlapping positions). In sensing $5' \to 3'$ WK automata, the processing of the input sequence ends if for all pairs of the sequence one of the letters is read. Due to the complementary relation, the sequence is fully processed; thus, the automaton makes a decision on the acceptance.

In the usual WK automata, the state transition is a mapping of the form $(Q \times \binom{V^*}{V^*}) \to 2^Q$. In a transition $q' \in \delta(q, \binom{w_1}{w_2})$, we call $r_l = |w_1|$ and $r_r = |w_2|$ the left and right *radius* of the transition (they are the lengths of the strings that the heads read from *left to right* and from *right to left* in this step, respectively). The value $r = r_l + r_r$ is the radius of the transition. Since $\delta(q, \binom{w_1}{w_2})$ is nonempty only for finitely many triplets of (q, w_1, w_2), there is a transition (maybe more) with the maximal radius for a given automaton. Let δ be extended by the sensing condition in the following way: Let r be the maximum of the values $r_l + r_r$ for the values given in the transition function of the original WK automaton. Then, let $\delta'\colon (Q \times \binom{V^*}{V^*} \times D) \to 2^Q$, where D is the *sensing distance set* $\{-\infty, 0, 1, \ldots, r, +\infty\}$. This set gives the distance of the two heads between 0 and r, $+\infty$ when the heads are further than r, or $-\infty$ when the heads are after their meeting point. Trivially, this automaton is finite, and D can be used only to control the sensing (i.e., the appropriate meeting of the heads). To describe the work of the automata, we use the concept of configuration. A configuration $\binom{w_1}{w_2}(q, s)\binom{w_1'}{w_2'}$ consists of the state q, the actual sensing distance s, and the input $\left[\begin{smallmatrix} w_1 w_1' \\ w_2 w_2' \end{smallmatrix}\right] \in \mathrm{WK}_\rho(V)$ in such a way that the first head (upper strand) has already processed the part w_1, while the second head (lower strand) has already processed w_2'. A step of the automaton, according to the state transition function, can be of the following two types:

(1) Normal steps : $\binom{w_1}{w_2 y}(q, +\infty)\binom{x w_1'}{w_2'} \Rightarrow \binom{w_1 x}{w_2}(q', s)\binom{w_1'}{y w_2'}$,
for $w_1, w_2, w_1', w_2', x, y \in V^*$ with $|w_2 y| - |w_1| > r$, $q, q' \in Q$,
if and only if $\begin{bmatrix} w_1 x w_1' \\ w_2 y w_2' \end{bmatrix} \in \mathrm{WK}_\rho(V)$ and $q' \in \delta(q, \binom{x}{y}, +\infty)$,
and $s = \begin{cases} |w_2| - |w_1 x| & \text{if } |w_2| - |w_1 x| \leq r; \\ +\infty & \text{in other cases.} \end{cases}$

(2) Sensing steps : $\binom{w_1}{w_2 y}(q, s)\binom{x w_1'}{w_2'} \Rightarrow \binom{w_1 x}{w_2}(q', s')\binom{w_1'}{y w_2'}$,
for $w_1, w_2, w_1', w_2', x, y \in V^*$,
if and only if $\begin{bmatrix} w_1 x w_1' \\ w_2 y w_2' \end{bmatrix} \in \mathrm{WK}_\rho(V)$ and $q' \in \delta(q, \binom{x}{y}, s)$,
and $s' = \begin{cases} s - |x| - |y| & \text{if } s - |x| - |y| \geq 0; \\ -\infty & \text{in other cases.} \end{cases}$

In the standard manner, extend $\Rightarrow$ to $\Rightarrow^n$, where $n \geq 0$; then, based on $\Rightarrow^n$, define $\Rightarrow^+$ and $\Rightarrow^*$. The accepted language, denoted by $L(M)$, can be defined by the final accepting configurations that can be reached from the initial one: A double strand $\begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$ is accepted by a sensing $5' \to 3'$ WK automaton M if and only if $\binom{\varepsilon}{w_2}(q_0, s_0)\binom{w_1}{\varepsilon} \Rightarrow^* \begin{bmatrix} w_1' \\ w_2' \end{bmatrix}(q_f, 0)\begin{bmatrix} w_1'' \\ w_2'' \end{bmatrix}$, for $q_f \in F$, where $\begin{bmatrix} w_1' \\ w_2' \end{bmatrix}\begin{bmatrix} w_1'' \\ w_2'' \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$ with the proper value of s_0 (it is $+\infty$ if $|w_1| > r$, elsewhere it is $|w_1|$); since the full input is processed by the time the heads meet.

From a biochemical point of view, a double-stranded sequence has no distinguishable start and end. Consequently, each word that is accepted by a WK automaton has a complement-symmetric pair which is also in the language. This fact does not cause any problem in connection to formal language theory. For instance, double strands having only A and C in a strand (and thus having T and G in the other) can represent languages over a binary alphabet: considering the pair $\begin{bmatrix} A \\ T \end{bmatrix}$ as letter a and $\begin{bmatrix} C \\ G \end{bmatrix}$ as letter b in the new alphabet V'.

Lastly, we briefly mention other closely related $5' \to 3'$ WK automata models. Besides the sensing version, the papers [18–21] also define the *full-reading sensing version*. The formal definition remains almost identical, however, the automaton continues with the reading when the heads meet, and both heads have to read their strand completely from the $5'$ end to the $3'$ end. The resulting behavior therefore combines some properties of classical WK automata and sensing $5' \to 3'$ WK automata. It can be easily seen that the full-reading sensing version is generally stronger than the sensing version. And finally, the paper [23] introduces a new version of sensing $5' \to 3'$ WK automata without the sensing distance. It shows that it is not strictly necessary to know the precise sensing distance and that we can obtain the same power even if we are able to recognize only the actual meeting event of heads. Nonetheless, this result does not hold in general if we consider restricted variations of these models.

3 Definitions

Considering the previously described sensing $5' \to 3'$ WK automata and full-reading sensing $5' \to 3'$ WK automata, there is quite a large gap between their behaviors. On the one hand, the definition of sensing $5' \to 3'$ WK automata states that we need to read only one of the letters from all pairs of the input sequence

before it is fully processed. However, this also limits the positioning of heads because they can read letters only until they meet. On the other hand, the definition of full-reading sensing $5' \to 3'$ WK automata allows the heads to traverse the whole input. Nonetheless, this also means that all pairs of the input sequence are read twice (which may be undesired). If we take into consideration other models, jumping finite automata utilize a mechanism that allows heads to skip (jump over) some symbols. Furthermore, the recently introduced double-jumping modes behave very similarly to $5' \to 3'$ WK automata. Due to this natural fit, it is our intention to fill and explore this gap by introducing the jumping mechanism into sensing $5' \to 3'$ WK automata. We want the heads to be able to traverse the whole input, but we also want to read all pairs of the input sequence only once.

With a simple direct approach, it is possible to fit the jumping mechanism straightforwardly into the original definition of sensing $5' \to 3'$ WK automata. (Note that we are now tracking only the meeting event of the heads from [23] and not the original precise sensing distance from [18–21].)

Definition 1 A *sensing* $5' \to 3'$ *WK automaton with jumping feature* is a 6-tuple $M = (V, \rho, Q, q_0, F, \delta)$, where V, ρ, Q, q_0, and F are the same as in WK automata, $V \cap \{\#\} = \emptyset$, $\delta \colon (Q \times \binom{V^*}{V^*} \times D) \to 2^Q$, where $D = \{\oplus, \ominus\}$ indicates the mutual position of heads, and the transition function assigns a nonempty set only for finitely many triplets of $(Q \times \binom{V^*}{V^*} \times D)$. We denote the head as $\blacktriangleright$-head or $\blacktriangleleft$-head if it reads from left to right or from right to left, respectively. We use the symbol $\oplus$ if the $\blacktriangleright$-head is on the input tape positioned before the $\blacktriangleleft$-head; otherwise, we use the symbol $\ominus$. A configuration $\binom{w_1}{w_2}(q, s)\binom{w_1'}{w_2'}$ has the same structure as in sensing $5' \to 3'$ WK automata; however, s indicates only the mutual position of heads, and a partially processed input $\binom{w_1 w_1'}{w_2 w_2'}$ may not satisfy the complementary relation ρ. A step of the automaton can be of the following two types: Let $w_1', w_2, x, y \in V^*$ and $w_1, w_2' \in (V \cup \{\#\})^*$.

(1) Reading steps: $\binom{w_1}{w_2 y}(q, s)\binom{x w_1'}{w_2'} \curvearrowright \binom{w_1 \{\#\}^{|x|}}{w_2}(q', s')\binom{w_1'}{\{\#\}^{|y|} w_2'}$, where $q' \in \delta(q, \binom{x}{y}, s)$, and s' is either $\oplus$ if $|w_2| > |w_1 x|$ or $\ominus$ in other cases.

(2) Jumping steps: $\binom{w_1}{w_2 v}(q, s)\binom{u w_1'}{w_2'} \curvearrowright \binom{w_1 u}{w_2}(q, s')\binom{w_1'}{v w_2'}$, where s' is either $\oplus$ if $|w_2| > |w_1 u|$ or $\ominus$ in other cases.

Note that the jumping steps are an integral and inseparable part of the behavior of the automaton, and thus they are not affected by the state transition function. In the standard manner, extend $\curvearrowright$ to $\curvearrowright^n$, where $n \geq 0$; then, based on $\curvearrowright^n$, define $\curvearrowright^+$ and $\curvearrowright^*$. The accepted language, denoted by $L(M)$, can be defined by the final accepting configurations that can be reached from the initial one: A double strand $[\binom{w_1}{w_2}]$ is accepted by a sensing $5' \to 3'$ WK automaton with jumping feature M if and only if $\binom{\varepsilon}{w_2}(q_0, \oplus)\binom{w_1}{\varepsilon} \curvearrowright^* \binom{w_1'}{\varepsilon}(q_f, \ominus)\binom{\varepsilon}{w_2'}$, for $q_f \in F$, where $w_1' = a_1 a_2 \dots a_n$, $w_2' = b_1 b_2 \dots b_n$, $a_i, b_i \in (V \cup \{\#\})$, and either $a_i = \#$ or $b_i = \#$, for all $i = 1, \dots, n$, for some $n \geq 0$.

From a practical point of view, however, this definition is not ideal. The automaton can easily end up in a configuration that cannot yield accepting results, and the correct positions of auxiliary symbols # need to be checked separately at the end of the process. Therefore, we present a modified definition that has the jumping mechanism more integrated into its structure. We are also using a simplification for complementary pairs and treat them as single letters; such a change

has no effect on the accepting power, and this form of input is more natural for formal language theory.

Definition 2 A *jumping* $5' \to 3'$ *WK automaton* is a quintuple $M = (V, Q, q_0, F, \delta)$, where V, Q, q_0, and F are the same as in WK automata, $V \cap \{\#\} = \emptyset$, the state transition function $\delta\colon (Q \times V^* \times V^* \times D) \to 2^Q$, where $D = \{\oplus, \ominus\}$ indicates the mutual position of heads, and δ assigns a nonempty set only for finitely many quadruples of $(Q \times V^* \times V^* \times D)$. A configuration (q, s, w_1, w_2, w_3) consists of the state q, the position of heads $s \in D$, and the three unprocessed portions of the input tape: (a) before the first head (w_1), (b) between the heads (w_2), and (c) after the second head (w_3). A step of the automaton can be of the following four types: Let $x, y, u, v, w_2 \in V^*$ and $w_1, w_3 \in (V \cup \{\#\})^*$.

(1) $\oplus$-reading: $(q, \oplus, w_1, xw_2y, w_3) \curvearrowright (q', s, w_1\{\#\}^{|x|}, w_2, \{\#\}^{|y|}w_3)$, where $q' \in \delta(q, x, y, \oplus)$, and s is either $\oplus$ if $|w_2| > 0$ or $\ominus$ in other cases.
(2) $\ominus$-reading: $(q, \ominus, w_1y, \varepsilon, xw_3) \curvearrowright (q', \ominus, w_1, \varepsilon, w_3)$, where $q' \in \delta(q, x, y, \ominus)$.
(3) $\oplus$-jumping: $(q, \oplus, w_1, uw_2v, w_3) \curvearrowright (q, s, w_1u, w_2, vw_3)$, where s is either $\oplus$ if $|w_2| > 0$ or $\ominus$ in other cases.
(4) $\ominus$-jumping: $(q, \ominus, w_1\{\#\}^*, \varepsilon, \{\#\}^*w_3) \curvearrowright (q, \ominus, w_1, \varepsilon, w_3)$.

In the standard manner, extend $\curvearrowright$ to $\curvearrowright^n$, where $n \geq 0$; then, based on $\curvearrowright^n$, define $\curvearrowright^+$ and $\curvearrowright^*$. The accepted language, denoted by $L(M)$, can be defined by the final accepting configurations that can be reached from the initial one: A string w is accepted by a jumping $5' \to 3'$ WK automaton M if and only if $(q_0, \oplus, \varepsilon, w, \varepsilon) \curvearrowright^* (q_f, \ominus, \varepsilon, \varepsilon, \varepsilon)$, for $q_f \in F$.

Even though the structure of this modified definition is considerably different from Definition 1, it is not very difficult to show that both models accept the same family of languages.

Proposition 3 *The models of Definitions 1 and 2 accept the same family of languages.*

Proof By construction (from both sides). Let $M_1 = (V_1, \rho, Q, q_0, F, \delta_1)$ from Definition 1 and $M_2 = (V_2, Q, q_0, F, \delta_2)$ from Definition 2. The states can remain identical. We can define bijection $\varphi\colon \rho \to V_2$. Let $\varphi(a_i, a_i') = x_i$ and $\varphi(b_i, b_i') = y_i$, where $a_i, a_i', b_i, b_i' \in V_1$, $(a_i, a_i'), (b_i, b_i') \in \rho$, $x_i, y_i \in V_2$, for all $i = 1, \ldots, n$, $n \geq 1$. Any $\delta_1(q, \binom{a_1 \ldots a_n}{b_1' \ldots b_m'}, s)$ can be converted into $\delta_2(q, x_1 \ldots x_n, y_1 \ldots y_m, s)$, for some $n, m \geq 0$, and vice versa. Now, we reason that both models accept the same inputs.

We say that a current configuration of M_1 is *potentially valid* if M_1 can still potentially reach the accepting configuration $\binom{w_1'}{\varepsilon}(q_f, \ominus)\binom{\varepsilon}{w_2'}$, $q_f \in F$, where $w_1' = a_1a_2 \ldots a_n$, $w_2' = a_1'a_2' \ldots a_n'$, $a_i, a_i' \in (V \cup \{\#\})$, and either $a_i = \#$ or $a_i' = \#$, for all $i = 1, \ldots, n$, $n \geq 0$. Observe that the condition regarding #'s can be checked continuously and individually for each pair $\binom{a_i}{a_i'}$ that was already visited with both heads. The following description thus considers only the configurations of M_1 that are still potentially valid.

First, we cover how M_1 can be simulated with M_2. The accepting process of M_1 can be divided into three distinct stages:

(1) Before the heads meet (the position of heads remains $\oplus$): The reading steps of M_1 clearly correspond with the $\oplus$-reading steps of M_2—the processed positions are marked with # in both models. Likewise, the jumping steps of M_1 clearly correspond with the $\oplus$-jumping steps of M_2—the visited positions are left unchanged for the other head in both models.

(2) The meeting point of heads (when the position changes from $\oplus$ to $\ominus$): The same steps as in (1) are still applicable. The difference is that in M_1 the heads can cross each other, but in M_2 the heads must meet each other precisely. However, the crossing situations in M_1 that lead to potentially valid configurations are quite limited. Assume that the $\blacktriangleright$-head reads/skips u and the $\blacktriangleleft$-head reads/skips v, then:
   (a) If $|u| > 1$ and $|v| > 1$, the resulting configuration cannot be potentially valid since some pair $\binom{a_i}{a'_i}$ was either read or skipped with both heads.
   (b) If $|u| > 1$ and $|v| = 0$: Considering a reading step, all symbols from u that are read after the meeting point must be skipped with the $\blacktriangleleft$-head. However, since jumping steps can occur arbitrarily, there is also an alternative sequence of steps in M_1 where the heads precisely meet, the $\blacktriangleleft$-head jumps afterwards, and the same configuration is reached. Moreover, any jumping step can be replaced with several shorter jumping steps.
   (c) If $|u| = 0$ and $|v| > 1$, the situation is analogous to (b).

   Thus, M_2 does not need to cover these crossing situations.

(3) After the heads met (the position of heads is $\ominus$): To keep the current configuration potentially valid, M_1 can use reading steps only on positions that were not yet read. Correspondingly, M_2 can use $\ominus$-reading steps on positions that do not contain #. Also, M_1 can effectively use jumping steps only on positions that were already read. Correspondingly, M_2 can use $\ominus$-jumping steps on positions that contain #.

Second, the simulation of M_2 with M_1 is trivial since any $\oplus/\ominus$-reading/jumping step of M_2 can be easily simulated with a reading/jumping step of M_1. Moreover, for the simulated steps, it is guaranteed that the condition regarding #'s holds.

Thus, when M_1 or M_2 accepts an input, there is a corresponding sequence of steps in the other model that accepts the same input as well. Consequently, M_1 and M_2 clearly accept the same inputs. A rigorous version of this proof is rather lengthy but straightforward, so we left it to the reader. □

Hereafter, we primarily use Definition 2.

4 Examples

To demonstrate the behavior of the automata, we present a few simple examples.

Example 4 Let us recall that $L = \{w \in \{a,b\}^* : |w|_a = |w|_b\}$ is a well-known nonlinear context-free language. We show that, even though the jumping directions in the model are quite restricted, we are able to accept such a language. Consider the following jumping $5' \to 3'$ WK automaton

$$M = (\{a,b\}, \{s\}, s, \{s\}, \delta)$$

with the state transition function δ: $\delta(s,a,b,\oplus) = \{s\}$ and $\delta(s,a,b,\ominus) = \{s\}$. Starting from s, M can either utilize the jumping or read simultaneously with both heads (the $\blacktriangleright$-head reads a and the $\blacktriangleleft$-head reads b), and it always stays in the sole state s. Now, consider the inputs *aaabbb* and *baabba*. The former can be accepted by using three $\oplus$-readings and one $\ominus$-jumping:

$$(s,\oplus,\varepsilon,aaabbb,\varepsilon) \curvearrowright (s,\oplus,\#,aabb,\#) \curvearrowright (s,\oplus,\#\#,ab,\#\#) \curvearrowright$$
$$(s,\ominus,\#\#\#,\varepsilon,\#\#\#) \curvearrowright (s,\ominus,\varepsilon,\varepsilon,\varepsilon).$$

The latter input is more complex and can be accepted by using one $\oplus$-jumping, two $\oplus$-readings, one $\ominus$-jumping, and one $\ominus$-reading:

$$(s, \oplus, \varepsilon, baabba, \varepsilon) \curvearrowright (s, \oplus, b, aabb, a) \curvearrowright (s, \oplus, b\#, ab, \#a) \curvearrowright$$
$$(s, \ominus, b\#\#, \varepsilon, \#\#a) \curvearrowright (s, \ominus, b, \varepsilon, a) \curvearrowright (s, \ominus, \varepsilon, \varepsilon, \varepsilon).$$

It is not hard to see that, by combining different types of steps, we can accept any input containing the same number of a's and b's, and thus $L(M) = L$.

Example 5 Consider the following jumping $5' \to 3'$ WK automaton

$$M = (\{a, b\}, \{s\}, s, \{s\}, \delta)$$

with the state transition function δ: $\delta(s, a, b, \oplus) = \{s\}$. Observe that this is almost identical to Example 4, however, we cannot use the $\ominus$-reading anymore. Consequently, we also cannot effectively use the $\oplus$-jumping because there is no way how to process remaining symbols afterwards. As a result, the accepted language changes to $L(M) = \{a^n b^n : n \geq 0\}$.

Lastly, we give a more complex example that uses all parts of the model.

Example 6 Consider the following jumping $5' \to 3'$ WK automaton

$$M = (\{a, b, c\}, \{s_0, s_1, s_2\}, s_0, \{s_0\}, \delta)$$

with δ: $\delta(s_0, a, b, \oplus) = \{s_1\}$, $\delta(s_1, \varepsilon, b, \oplus) = \{s_0\}$, $\delta(s_0, c, c, \ominus) = \{s_2\}$, and $\delta(s_2, \varepsilon, c, \ominus) = \{s_0\}$. We can divide the accepting process of M into two stages. First, before the heads meet, the automaton ensures that for every a on the left side there are two b's on the right side; other symbols are skipped with the jumps. Second, after the heads met, the automaton checks if the part before the meeting point has double the number of c's as the part after the meeting point. Thus, $L(M) = \{w_1 w_2 : w_1 \in \{a, c\}^*,\ w_2 \in \{b, c\}^*,\ 2 \cdot |w_1|_a = |w_2|_b,\ |w_1|_c = 2 \cdot |w_2|_c\}$.

5 General results

These results cover the general behavior of jumping $5' \to 3'$ WK automata without any further restrictions. Let **SWK**, **JWK**, **GJFA**, and **JFA** denote the language families accepted by sensing $5' \to 3'$ WK automata, jumping $5' \to 3'$ WK automata, general jumping finite automata, and jumping finite automata, respectively.

Considering previous results on other models that use the jumping mechanism (see [16, 17, 8–10]), it is a common characteristic that they define language families that are incomparable with the classical families of regular, linear, and context-free languages. On the other hand, sensing $5' \to 3'$ WK automata (see [18–21, 23]) are closely related to the family of linear languages. First, we show that the new model is able to accept all regular and linear languages. Furthermore, the accepting power goes beyond the family of linear languages.

Lemma 7 *For every regular language L, there is a jumping $5' \to 3'$ WK automaton M such that $L = L(M)$.*

Proof Consider a finite automaton $N = (V, Q, q_0, F, \delta_1)$ such that $L(N) = L$. We can construct the jumping $5' \to 3'$ WK automaton $M = (V, Q, q_0, F, \delta_2)$ where $\delta_2(q, a, \varepsilon, \oplus) = \delta_1(q, a)$ for all $q \in Q$, $a \in (V \cup \{\varepsilon\})$. Observe that with such a state transition function the $\oplus$-reading steps always look like this: $(q, \oplus, w_1, aw_2, w_3) \curvearrowright (q', s, w_1\{\#\}^{|a|}, w_2, w_3)$, where $q' \in \delta_2(q, a, \varepsilon, \oplus)$, $w_2 \in V^*$, $w_1, w_3 \in (V \cup \{\#\})^*$, and s is either $\oplus$ if $|w_2| > 0$ or $\ominus$ in other cases. There are no possible $\ominus$-reading steps. The $\oplus$-jumping can be potentially used to skip some symbols before the heads meet; nonetheless, the resulting configuration will be in the form (q, s, w_1, w_2, w_3) where $\text{alph}(w_1 w_3) \cap V \neq \emptyset$. Since there is no way how to read such symbols in w_1 and w_3, the configuration cannot yield an accepting result. Consequently, any input string will be read in M the same way as in N (the remaining #'s will be erased with the $\ominus$-jumping afterwards). Thus, $L(M) = L(N) = L$. □

Lemma 8 *For every sensing $5' \to 3'$ WK automaton M_1, there is a jumping $5' \to 3'$ WK automaton M_2 such that $L(M_1) = L(M_2)$.*

Proof This can be proven by construction. Consider any sensing $5' \to 3'$ WK automaton M_1. A direct conversion would be complicated, however, let us recall that **LIN** = **SWK** (see Theorem 2 in [21]). Consider a linear grammar $G = (N, T, S, P)$ such that $L(G) = L(M_1)$. We can construct the jumping $5' \to 3'$ WK automaton M_2 such that $L(M_2) = L(G)$. Assume that $q_f \notin (N \cup T)$. Define $M_2 = (T, N \cup \{q_f\}, S, \{q_f\}, \delta)$, where $B \in \delta(A, u, v, \oplus)$ if $A \to uBv \in P$ and $q_f \in \delta(A, u, \varepsilon, \oplus)$ if $A \to u \in P$ $(A, B \in N,\ u, v \in T^*)$. By the same reasoning as in the proof of Lemma 7, only the $\oplus$-reading can be effectively used before the heads meet. Consequently, it can be easily seen that M_2 reads all symbols in the same fashion as G generates them. Moreover, the heads of M_2 can meet each other with the accepting state q_f if and only if G can finish the generation process with a rule $A \to u$. Thus, $L(M_2) = L(G) = L(M_1)$. □

Theorem 9 ***LIN*** $=$ ***SWK*** $\subset$ ***JWK***.

Proof **SWK** $\subseteq$ **JWK** follows from Lemma 8. **LIN** = **SWK** was proven in [21]. **JWK** $\not\subseteq$ **LIN** follows from Example 4. □

The next two characteristics follow from the previous results.

Theorem 10 *Jumping $5' \to 3'$ WK automata without $\ominus$-reading steps accept linear languages.*

Proof Consider any jumping $5' \to 3'$ WK automaton $M = (V, Q, q_0, F, \delta)$ that has no possible $\ominus$-reading steps. Expanding the reasoning in the proof of Lemma 8, if there are no possible $\ominus$-reading steps, the $\oplus$-jumping cannot be effectively used, and we can construct a linear grammar that generates strings in the same fashion as M reads them. Define the linear grammar $G = (Q, V, q_0, R)$, where R is constructed in the following way: (1) For each $p \in \delta(q, u, v, \oplus)$, add $q \to upv$ to R. (2) For each $f \in F$, add $f \to \varepsilon$ to R. Clearly, $L(G) = L(M)$. □

Proposition 11 *The language family accepted by double-jumping finite automata that perform right-left and left-right jumps (see [10]) is strictly included in* ***JWK***.

Proof First, Theorem 3.18 in [10] shows that jumping finite automata that perform right-left and left-right jumps accept the same family of languages. Second, Theorem 3.7 in [10] shows that this family is strictly included in **LIN**. Finally, Theorem 9 shows that **LIN** is strictly included in **JWK**. □

Even though the model is able to accept some nonlinear languages, the jumping directions of the heads are quite restricted compared to general jumping finite automata. Consequently, there are some languages accepted by jumping $5' \to 3'$ WK automata and general jumping finite automata that cannot be accepted with the other model. To formally prove these results, we also need to introduce the concept of the debt of a configuration in jumping $5' \to 3'$ WK automata. First, we start with the formal definition of a reachable state.

Definition 12 Let $M = (V, Q, q_0, F, \delta)$ be a jumping $5' \to 3'$ WK automaton. Assuming some states $q, q' \in Q$ and a mutual position of heads $s \in \{\oplus, \ominus\}$, we say that q' is *reachable* from q and s if there exists a configuration (q, s, w_1, w_2, w_3) such that $(q, s, w_1, w_2, w_3) \curvearrowright^* (q', s', w_1', w_2', w_3')$ in M, $s' \in \{\oplus, \ominus\}$, $w_1, w_2, w_3, w_1', w_2', w_3' \in (V \cup \{\#\})^*$.

Next, we show that for any possible sequence of reading steps of a jumping $5' \to 3'$ WK automaton M that leads from the starting configuration σ to a configuration γ from which a final state is reachable, there exists $w' \in L(M)$ such that w' can be fully processed with the given sequence of reading steps from σ to γ and a limited number of additional reading steps. Note that potential jumping steps in σ to γ are unimportant for the result since jumping steps can occur arbitrarily and they do not process any symbols of the input.

Lemma 13 *Let $M = (V, Q, q_0, F, \delta)$ be a jumping $5' \to 3'$ WK automaton, and let $q \in Q$ and $s \in \{\oplus, \ominus\}$ such that $f \in F$ is reachable from q and s. When $(q_0, \oplus, \varepsilon, w, \varepsilon) \curvearrowright^* (q, s, w_1, w_2, w_3)$ in M, $w \in V^*$, $w_1, w_2, w_3 \in (V \cup \{\#\})^*$, there exists $w' \in L(M)$ such that M starting with w' can reach q and s' ($s' = s$ or $s' = \ominus$) by using the same sequence of $\oplus/\ominus$-reading steps as in $(q_0, \oplus, \varepsilon, w, \varepsilon) \curvearrowright^* (q, s, w_1, w_2, w_3)$ and the rest of w' can be processed with a limited number of steps bounded by some constant k for M.*

Proof First, if f is reachable from q and s, there has to exist a sequence of state transitions from $(Q \times \{\oplus, \ominus\})^+$ such that $(p_0, s_0) \cdots (p_n, s_n)$,
- $p_0 = q$, $s_0 = s'$, $p_n = f$, $s_n = \ominus$,
- $p_{i+1} \in \delta(p_i, a_i, b_i, s_i)$ or $p_{i+1} = p_i$,
- $s_{i+1} = s_i$ or $s_{i+1} = \ominus$, and
- $(p_i, s_i) = (p_j, s_j)$ implies $i = j$, $i, j = 0, \ldots, n$, $n \geq 0$ (all pairs are unique).

Obviously, this sequence is finite, and its maximum length is bounded by some constant k for M. Second, assume the complete sequence of state transitions of M, $(q_0, \oplus) \cdots (q, s') \cdots (f, \ominus)(f, \ominus)$, where $(q_0, \oplus) \cdots (q, s')$ includes the same sequence of reading steps as $(q_0, \oplus, \varepsilon, w, \varepsilon) \curvearrowright^* (q, s, w_1, w_2, w_3)$ and $(q, s') \cdots (f, \ominus) = (p_0, s_0) \cdots (p_n, s_n)$. Represent the complete sequence as $(p_0, s_0) \cdots (p_m, s_m)$, $p_{i+1} \in \delta(p_i, u_i, v_i, s_i)$ or $p_{i+1} = p_i$, $s_{i+1} = s_i$ or $s_{i+1} = \ominus$, $i = 0, \ldots, m$, $m \geq 0$. At first, for all $i = 0, \ldots, m$, set $a_i = \varepsilon$, $b_i = \varepsilon$, $c_i = \varepsilon$, $d_i = \varepsilon$. If $p_{i+1} \in \delta(p_i, u_i, v_i, s_i)$ is used, then if $s_i = \oplus$, set $a_i = u_i$ and $b_i = v_i$, otherwise if $s_i = \ominus$, set $c_i = u_i$ and $d_i = v_i$. It is not hard to see that $w' = a_0 \cdots a_m d_m \cdots d_0 c_0 \cdots c_m b_m \cdots b_0 \in L(M)$. □

Next, based on known M and $L(M)$, we can define the debt of a configuration of M. If we follow the computation of M on an input w, we can maintain the Parikh vector o of symbols already processed from w in the current configuration. Additionally, with the known $L(M)$, we can determine Parikh vectors for all $w' \in L(M)$. The debt of the configuration represents the minimum number of symbols that have to be added to o so that o matches some Parikh vector of $w' \in L(M)$. Note that we use ∞ to cover situations when no match is possible.

Definition 14 Let $M = (V, Q, q_0, F, \delta)$ be a jumping $5' \to 3'$ WK automaton, where $V = \{a_1, \ldots, a_n\}$. Following the computation of M on an input $w \in V^*$, let $o = (o_1, \ldots, o_n)$ be the Parikh vector built by the processed (read) symbols from w: At first, for the starting configuration, set $o = \Psi_V(\varepsilon)$. For the following configurations, whenever M makes a $\oplus/\ominus$-reading step from some q to q' according to $q' \in \delta(q, u, v, s)$, set $o = o + \Psi_V(uv)$. Using the Parikh mapping of $L(M)$, we define $\Delta(o) = \{\sum_{i=1}^{n}(m_i - o_i) : (m_1, \ldots, m_n) \in \Psi_V(L(M)),\ m_i \geq o_i,\ 1 \leq i \leq n\} \cup \{\infty\}$. Finally, we define the *debt* of the current configuration of M as $min(\Delta(o))$.

And finally, we can combine Lemma 13 and Definition 14 to show that any jumping $5' \to 3'$ WK automaton M has to accept all $w \in L(M)$ over configurations with some bounded debt.

Lemma 15 *Let L be a language, and let $M = (V, Q, q_0, F, \delta)$ be a jumping $5' \to 3'$ WK automaton. If $L(M) = L$, M accepts all $w \in L$ using only configurations that have their debt bounded by some constant k for M.*

Proof By contradiction. Assume that M does not accept all $w \in L$ exclusively using only configurations that have their debt bounded by some constant k for M, then M can accept some $w \in L$ over a configuration for which the debt cannot be bounded by any k. Let $V = \{a_1, \ldots, a_n\}$. Following the computation of M on $w \in L$, let $o = (o_1, \ldots, o_n)$ be the Parikh vector built by the processed symbols from w. Due to Lemma 13, when M is in a state q with a mutual position of heads s from which a final state $f \in F$ is reachable, there is $w' \in L(M)$ such that $\Psi(w') = (m_1, \ldots, m_n)$, $m_i \geq o_i$, $1 \leq i \leq n$, and $|w'| \leq \sum_{i=1}^{n}(o_i) + k'$, where k' is some constant for M. According to Definition 14, $w' \in L(M)$ implies $min(\Delta(o)) \leq k'$. Finally, M clearly cannot accept w over a state q and a mutual position of heads s from which no final state $f \in F$ is reachable. Consequently, when M accepts w, it must be done over configurations with the debt $\leq k'$. However, that is a contradiction with the premise that M can accept some $w \in L$ over a configuration for which the debt cannot be bounded by any k. □

Observe that the debt alone does not depend on the order of symbols in the words of $L(M)$, e.g., $\Psi_V(\{(abc)^n : n \geq 0\}) = \Psi_V(\{a^n b^n c^n : n \geq 0\})$, for $V = \{a, b, c\}$. However, when the debt is combined with the computational possibilities of M on an input w, we can show that a language L cannot be accepted by M if there is some $w \in L$ such that w cannot be fully processed over configurations of M with the debt bounded by some constant k.

Lemma 16 *There is no jumping $5' \to 3'$ WK automaton M such that $L(M) = \{a^n b^n c^n : n \geq 0\}$.*

Proof Basic idea. Considering any sufficiently large constant k, we show that M cannot process all symbols of $a^{10k}b^{10k}c^{10k}$ using only configurations that have their debt bounded by k.
Formal proof. By contradiction. Let $L = \{a^n b^n c^n : n \geq 0\}$, and let $M = (V, Q, q_0, F, \delta)$ be a jumping $5' \to 3'$ WK automaton such that $L(M) = L$. Due to Lemma 15, M must accept all $w \in L$ using only configurations that have their debt bounded by some constant k for M. Consider any k such that $k > max(\{|uv| : \delta(q, u, v, s) \neq \emptyset,\ u, v \in V^*\})$. Due to the structure of L, we can represent the debt of the configuration as $\langle d_a, d_b, d_c \rangle$, where d_a, d_b, d_c is the minimum number of symbols a, b, c that M must yet to read to get the balanced number of processed symbols. (For illustration, the initial configuration has the debt $\langle 0, 0, 0 \rangle$. When M reads a, the following configuration has the debt $\langle 0, 1, 1 \rangle$ because at least one b and one c have yet to be read to keep the number of processed symbols balanced.) When $(q_0, \oplus, \varepsilon, w, \varepsilon) \curvearrowright^* (q_f, \ominus, \varepsilon, \varepsilon, \varepsilon)$, $q_f \in F$, for all traversed configurations must hold $d_a + d_b + d_c \leq k$. Let $w = a^{10k}b^{10k}c^{10k}$.

First, we explore the maximum number of symbols that M can read from w before the heads meet. Starting from $(q_0, \oplus, \varepsilon, w, \varepsilon)$ $\langle 0, 0, 0 \rangle$ and until the position $\ominus$ is reached, M can use $\oplus$-reading steps to process symbols and $\oplus$-jumping steps to skip symbols. Consider the optimal reading strategy to process the maximum number of symbols from $a^{10k}b^{10k}c^{10k}$:
(1) M processes (with multiple steps) a^k and c^k and reaches $\langle 0, k, 0 \rangle$,
(2) M reads l symbols together in one step (balanced number of a's, b's, and c's) while keeping $\langle 0, k, 0 \rangle$, $l < k$,
(3) M processes b^{2k} and a^k (or c^k) and reaches $\langle 0, 0, k \rangle$ (or $\langle k, 0, 0 \rangle$).
No further reading is possible; this strategy processed $5k + l$ symbols.

Second, when the heads meet, $a^{>4k}b^{>4k}c^{>4k}$ has yet to be processed. The heads are next to each other, and M can use $\ominus$-reading steps to process symbols and $\ominus$-jumping steps to remove the auxiliary #'s. Consider one of the optimal reading strategies to process the maximum number of symbols:
(1) the heads are between b's and c's,
(2) the debt of the current configuration is $\langle 0, k, 0 \rangle$,
(3) M processes b^{2k} and c^k and reaches $\langle k, 0, 0 \rangle$.
No further reading is possible; this strategy processed $3k$ symbols. (Since there are still more than $2k$ of b's, M cannot reach a's to further balance the debt.)

Even when M follows the best reading strategies in both parts, it is not able to process more than $8k + l$ symbols; but the input contains $30k$ symbols. Consequently, there is no constant k that bounds the debt of configurations of M. But that is a contradiction with the assumption that there is a jumping $5' \to 3'$ WK automaton M such that $L(M) = \{a^n b^n c^n : n \geq 0\}$. □

Lemma 17 *There is no jumping $5' \to 3'$ WK automaton M such that $L(M) = \{w \in \{a, b, c\}^* : |w|_a = |w|_b = |w|_c\}$.*

Proof Let N be a jumping $5' \to 3'$ WK automaton, $L = \{w \in \{a, b, c\}^* : |w|_a = |w|_b = |w|_c\}$, and $K = \{a^n b^n c^n : n \geq 0\}$. Following the computation of N on an input w of the form $a^*b^*c^*$, let o be the Parikh vector built by the processed symbols from w. Observe that, for any configuration of N, the debt of the configuration $min(\Delta(o))$ is similar for $L(N) = L$ and $L(N) = K$ since it only depends on o and the quantities of symbols in the words of the language $L(N)$. Consequently, the proof that there is no M is analogous to the proof of Lemma 16. □

Proposition 18 ***JWK*** *is incomparable with* ***GJFA*** *and* ***JFA****.*

Proof The language $\{w \in \{a,b\}^* : |w|_a = |w|_b\}$ from Example 4 and the language $\{w \in \{a,b,c\}^* : |w|_a = |w|_b = |w|_c\}$ from Lemma 17 are accepted with (general) jumping finite automata (see Example 5 in [16]). The language $\{a^n b^n : n \geq 0\}$ from Example 5 is not accepted with (general) jumping finite automata (see Lemma 19 in [16]). □

The last group of results compares the accepting power of the model with the families of context-sensitive and context-free languages.

Theorem 19 ***JWK*** $\subset$ ***CS****.*

Proof Clearly, the use of two heads and the jumping behavior can be simulated by linear bounded automata, so **JWK** $\subseteq$ **CS**. From Lemma 16, **CS** $-$ **JWK** $\neq \emptyset$. □

Lemma 20 *There are some non-context-free languages accepted by jumping* $5' \to 3'$ *WK automata.*

Proof Consider the following jumping $5' \to 3'$ WK automaton

$$M = (\{a,b,c,d\}, \{s\}, s, \{s\}, \delta)$$

with the state transition function δ: $\delta(s,a,c,\oplus) = \{s\}$ and $\delta(s,d,b,\ominus) = \{s\}$. The accepting process has two stages. First, before the heads meet, the automaton reads the same number of a's and c's. Second, after the heads met, the automaton reads the same number of d's and b's. Thus, $L(M) = \{w_1 w_2 : w_1 \in \{a,b\}^*,\ w_2 \in \{c,d\}^*,\ |w_1|_a = |w_2|_c,\ |w_1|_b = |w_2|_d\}$.
By contradiction. Assume that $L(M)$ is a context-free language. The family of context-free languages is closed under intersection with regular sets. Let $K = L(M) \cap \{a\}^*\{b\}^*\{c\}^*\{d\}^*$. Clearly, there are some strings in $L(M)$ that satisfy this forced order of symbols. Furthermore, they all have the proper correlated numbers of these symbols. Consequently, $K = \{a^n b^m c^n d^m : n,m \geq 0\}$. However, K is a well-known non-context-free language (see Chapter 3.1 in [25]). That is a contradiction with the assumption that $L(M)$ is a context-free language. Therefore, $L(M)$ is a non-context-free language. □

Lemma 21 *There is no jumping* $5' \to 3'$ *WK automaton* M *such that* $L(M) = \{a^n b^n c^m d^m : n,m \geq 0\}$.

Proof Basic idea. We follow the proof structure of Lemma 16. Considering any sufficiently large constant k, we show that M cannot process all symbols of $a^{10k}b^{10k}c^{10k}d^{10k}$ using only configurations that have their debt bounded by k.
Formal proof. By contradiction. Let $L = \{a^n b^n c^m d^m : n,m \geq 0\}$, and let $M = (V, Q, q_0, F, \delta)$ be a jumping $5' \to 3'$ WK automaton such that $L(M) = L$. Due to Lemma 15, M must accept all $w \in L$ using only configurations that have their debt bounded by some constant k for M. Consider any k such that $k > max(\{|uv| : \delta(q,u,v,s) \neq \emptyset,\ u,v \in V^*\})$. Let $w = a^{10k}b^{10k}c^{10k}d^{10k}$.

First, we explore the limits of what the heads of M can read from the input after the meeting point. Consider the maximum number of b's that the $\blacktriangleleft$-head can read with $\ominus$-reading steps. Since a's are in front of b's, this number must be limited. The starting configuration can have the debt of k b's, the debt can

reach k a's, and only one step can read both types of symbols together. Thus, the maximum number of b's that the ◀-head can read with $\ominus$-reading steps is $< 3k$. In the same manner, the maximum number of c's that the ▶-head can read with $\ominus$-reading steps is $< 3k$.

Second, we explore the situation before the heads meet. Observe that the ▶-head can process less than $4k$ of a's and b's on its own with $\oplus$-reading steps and the ◀-head can process less than $4k$ of c's and d's on its own with $\oplus$-reading steps. Due to the previous limits, the ▶-head cannot jump over all remaining b's and the ◀-head cannot jump over all remaining c's. Thus, the heads cannot meet in a configuration that can process all remaining symbols.

Consequently, there is no constant k that bounds the debt of configurations of M. But that is a contradiction with the assumption that there is a jumping $5' \to 3'$ WK automaton M such that $L(M) = \{a^n b^n c^m d^m : n, m \geq 0\}$. □

Theorem 22 ***JWK*** *and* ***CF*** *are incomparable.*

Proof **JWK** $\not\subseteq$ **CF** follows from Lemma 20. **CF** $\not\subseteq$ **JWK** follows from Lemma 21. Lastly, **LIN** $\subset$ **JWK** and **LIN** $\subset$ **CF**. □

6 Results on restricted variations

In this section, we compare the accepting power of unrestricted and restricted variations of jumping $5' \to 3'$ WK automata. This paper considers the same standard restrictions as they are defined for Watson-Crick finite automata. Since these restrictions regulate only the state control and reading steps of the automaton, the jumping is not affected in any way. Let **JWK** denote the language family accepted by jumping $5' \to 3'$ WK automata. We are using prefixes **N**, **F**, **S**, **1**, **NS**, **FS**, **N1**, and **F1** to specify the restricted variations of jumping $5' \to 3'$ WK automata and appropriate language families.

In the field of DNA computing, the empty string/sequence usually does not belong to any language because it does not refer to a molecule. This paper is not so strict and thus considers the empty string as a possible valid input. Nonetheless, the following proofs are deliberately based on more complex inputs to mitigate the impact of the empty string on the results. Moreover, we distinguish between **FIN** and $\mathbf{FIN}_{\varepsilon\text{-inc}}$, when the difference is unavoidable.

Note that there are some inherent inclusions between language families based on the application of restrictions on the model. Additionally, several other basic relations can be established directly from the restriction definitions:

Lemma 23 *The following relations hold: (i)* ***N JWK*** $\subseteq$ ***F JWK****; (ii)* ***1 JWK*** $\subseteq$ ***S JWK****; (iii)* ***F1 JWK*** $\subseteq$ ***FS JWK****; (iv)* ***N1 JWK*** $\subseteq$ ***NS JWK****; (v)* ***NS JWK*** $\subseteq$ ***FS JWK****; (vi)* ***N1 JWK*** $\subseteq$ ***F1 JWK****.*

Proof These results follow directly from the definitions since the stateless restriction (**N**) is a special case of the all-final restriction (**F**) and the 1-limited restriction (**1**) is a special case of the simple restriction (**S**). □

6.1 On the simple restriction

Theorem 24 ***S JWK*** $=$ ***JWK***.

Proof
Basic idea. Any general reading step can be replaced with two simple reading steps and a new auxiliary state that together accomplish the same action.
Construction. Consider an arbitrary jumping $5' \to 3'$ WK automaton $M = (V, Q_1, q_0, F, \delta_1)$. We can construct the **S** jumping $5' \to 3'$ WK automaton N such that $L(N) = L(M)$. Define $N = (V, Q_2, q_0, F, \delta_2)$, where Q_2 and δ_2 are created in the following way:
(1) Set $Q_2 = Q_1$.
(2) For each $\delta_1(q, x, y, s) \neq \emptyset$ where $|x| = 0$ or $|y| = 0$, let $\delta_2(q, x, y, s) = \delta_1(q, x, y, s)$.
(3) For each $\delta_1(q, x, y, s) \neq \emptyset$ where $|x| > 0$ and $|y| > 0$, add a new unique state p to Q_2 and let $p \in \delta_2(q, x, \varepsilon, s)$ and $\delta_2(p, \varepsilon, y, s) = \delta_1(q, x, y, s)$.
It is clear that all original transitions that did not satisfy the simple restriction were transformed into the new suitable transitions according to the basic idea. Now we reason that this change has no effect on the accepted language. Observe that the mutual position of heads does not change between both transitions. Therefore, considering the definitions of $\oplus$-reading and $\ominus$-reading steps, the first reading cannot have any side effect on the second reading because the heads work on different parts of the input. Lastly, the new automaton can do additional jumps between the readings. Nonetheless, considering again the mutual position of heads, jumps with the same effect can also be done before the first reading or after the second reading. Thus, $L(N) = L(M)$. □

6.2 On the 1-limited restriction

Example 25 Consider the following jumping $5' \to 3'$ WK automaton $M = (\{a, b, c\}, \{s, f\}, s, \{f\}, \delta)$ with the state transition function δ:

$$\delta(s, a, b, \oplus) = \{s\}, \quad \delta(f, a, b, \oplus) = \{f\}, \quad \delta(f, a, b, \ominus) = \{f\},$$
$$\delta(s, cc, \varepsilon, \oplus) = \{f\}, \quad \delta(s, \varepsilon, cc, \oplus) = \{f\}.$$

It is clear that the first three transitions mimic the behavior of Example 4. The other two transitions ensure that the input is accepted only if it also contains precisely one substring cc. Therefore, $L(M) = \{w_1 ccw_2 : w_1, w_2 \in \{a, b\}^*, \ |w_1 w_2|_a = |w_1 w_2|_b\}$.

Lemma 26 *Let $M = (V, Q, q_0, F, \delta)$ be a **1** jumping $5' \to 3'$ WK automaton M, and let w be an input string. When M reads w, let us represent the sequence of used reading steps as $(u_1, v_1, u_1', v_1') \cdots (u_n, v_n, u_n', v_n')$, $u_i, u_i', v_i, v_i' \in (V \cup \{\varepsilon\})$, $i = 1, \ldots, n$, $n \geq 1$, where u_i and u_i' contain the symbol read by the $\blacktriangleright$-head with the $\oplus/\ominus$-reading step or ε and v_i and v_i' contain the symbol read by the $\blacktriangleleft$-head with the $\oplus/\ominus$-reading step or ε. Let $w_{\blacktriangleright\oplus} = u_1 \cdots u_n$, $w_{\blacktriangleright\ominus} = u_1' \cdots u_n'$, $w_{\blacktriangleleft\oplus} = v_n \cdots v_1$, $w_{\blacktriangleleft\ominus} = v_n' \cdots v_1'$. If $w \in L(M)$, $xy \in L(M)$ for all $x \in \text{shuffle}(w_{\blacktriangleright\oplus}, w_{\blacktriangleleft\ominus})$ and $y \in \text{shuffle}(w_{\blacktriangleleft\oplus}, w_{\blacktriangleright\ominus})$.*

Proof Since M satisfies the 1-limited restriction, exactly one symbol is always being read with a reading step. Therefore, for all i, only one of u_i, v_i, u'_i, v'_i is nonempty and it contains one symbol. If M accepts w and a head jumps over a symbol with a $\oplus$-jumping step, such a symbol is read later with the other head with a $\ominus$-reading step. Since jumping steps can occur arbitrarily between reading steps and since they do not depend on the state of M, it follows that every uv, where u is a shuffle of $w_{\blacktriangleright\oplus}$ and $w_{\blacktriangleleft\ominus}$ and v is a shuffle of $w_{\blacktriangleleft\oplus}$ and $w_{\blacktriangleright\ominus}$, has to belong into $L(M)$. □

Lemma 27 *There is no* ***1*** *jumping* $5' \to 3'$ *WK automaton* M *such that* $L(M) = \{w_1 ccw_2 : w_1, w_2 \in \{a,b\}^*,\ |w_1 w_2|_a = |w_1 w_2|_b\}$.

Proof Basic idea. We follow the proof structure of Lemma 16. Considering any sufficiently large constant k, we show that M cannot process all symbols of $a^{2k}b^{2k}cc\,b^{2k}a^{2k}$ using only configurations that have their debt bounded by k.
Formal proof. By contradiction. Let $L = \{w_1 ccw_2 : w_1, w_2 \in \{a,b\}^*,\ |w_1 w_2|_a = |w_1 w_2|_b\}$, and let $M = (V, Q, q_0, F, \delta)$ be a **1** jumping $5' \to 3'$ WK automaton such that $L(M) = L$. Due to Lemma 15, M must accept all $w \in L$ using only configurations that have their debt bounded by some constant k for M. Consider any $k \geq 2$. Let $w = a^{2k}b^{2k}ccb^{2k}a^{2k}$.

Consider restrictions on how M can accept w so that it does not also accept $w' \notin L$. Due to Lemma 26, to ensure that both c's are always next to each other, some parts of $w_{\blacktriangleright\oplus}$, $w_{\blacktriangleright\ominus}$, $w_{\blacktriangleleft\oplus}$, $w_{\blacktriangleleft\ominus}$ must remain empty.

Consider cases where two or three parts remain empty. To ensure the proper position of c's, either only one head can read or only $\oplus$-reading or $\ominus$-reading steps can be used. It is not hard to see that in these cases M can process at most $5k+2$ symbols from the input. (In the optimal strategy to process the maximum number of symbols, the heads meet between a's and b's and $\ominus$-reading steps are used.)

If only one part remains empty, the appropriate opposite part for the shuffle must contain both c's. Let us assume that $w_{\blacktriangleleft\ominus}$ remains empty. Consequently, $w_{\blacktriangleright\oplus}$ must contain at least $a^{2k}b^{2k}cc$. Consider the optimal reading strategy to process the maximum number of symbols into $w_{\blacktriangleright\oplus}$:
(1) the $\blacktriangleleft$-head reads $k-2$ times a (the initial debt contains 2 c's),
(2) $2k$ times the $\blacktriangleleft$-head reads b and the $\blacktriangleright$-head reads a,
(3) the $\blacktriangleright$-head reads $2k-4$ times b.
No further reading is possible; $w_{\blacktriangleright\oplus}$ does not contain all required symbols. The proof strategy and results are analogous for the other cases where $w_{\blacktriangleright\ominus}$, $w_{\blacktriangleleft\oplus}$, or $w_{\blacktriangleleft\ominus}$ remains empty.

Consequently, there is no constant k that bounds the debt of configurations of M. But that is a contradiction with the assumption that there is a **1** jumping $5' \to 3'$ WK automaton M such that $L(M) = \{w_1 ccw_2 : w_1, w_2 \in \{a,b\}^*,\ |w_1 w_2|_a = |w_1 w_2|_b\}$. □

Theorem 28 ***1 JWK*** $\subset$ ***JWK***.

Proof This theorem follows directly from Example 25 and Lemma 27. □

Example 29 Consider the following **1** jumping $5' \to 3'$ WK automaton $M = (\{a,b\}, \{s,p\}, s, \{s\}, \delta)$ with the state transition function δ:

$$\delta(s,a,\varepsilon,\oplus) = \{p\}, \quad \delta(p,\varepsilon,b,\oplus) = \{s\},$$
$$\delta(s,a,\varepsilon,\ominus) = \{p\}, \quad \delta(p,\varepsilon,b,\ominus) = \{s\}.$$

It is not hard to see that the resulting behavior is similar to Example 4. The automaton now reads a's and b's with separate steps and uses one auxiliary state that is not final. Consequently, $L(M) = \{w \in \{a,b\}^* : |w|_a = |w|_b\}$.

Lemma 30 *For every linear grammar G, there is a **1** jumping $5' \to 3'$ WK automaton M such that $L(G) = L(M)$.*

Proof By construction. Consider any linear grammar $G = (N, T, S, P)$. Every linear grammar has an equivalent grammar with rules in the form: (1) $S \to \varepsilon$, (2) $A \to aB$, (3) $A \to Ba$, (4) $A \to a$, where $A \in N$, $B \in (N - \{S\})$, and $a \in T$. Without loss of generality, assume that G satisfies this special form of rules and $q_f \notin (N \cup T)$. Define the **1** jumping $5' \to 3'$ WK automaton $M = (T, N \cup \{q_f\}, S, F, \delta)$, where F and δ are constructed in the following way:
(1) Set $F = \{q_f\}$. If $S \to \varepsilon \in P$, add S to F.
(2) For each $A \to aB \in P$, add B to $\delta(A, a, \varepsilon, \oplus)$.
(3) For each $A \to Ba \in P$, add B to $\delta(A, \varepsilon, a, \oplus)$.
(4) For each $A \to a \in P$, add q_f to $\delta(A, a, \varepsilon, \oplus)$.
Following the same reasoning as in Lemma 8, $L(M) = L(G)$. □

Theorem 31 ***LIN*** $\subset$ ***1 JWK***.

Proof This theorem follows directly from Example 29 and Lemma 30. □

6.3 On the all-final restriction

Lemma 32 *There is no **F** jumping $5' \to 3'$ WK automaton M such that $L(M) = \{ca^n cb^n c : n \geq 0\} \cup \{\varepsilon\}$.*

Proof By contradiction. Assume that there is an **F** jumping $5' \to 3'$ WK automaton $M = (V, Q, q_0, F, \delta)$ such that $L(M) = \{ca^n cb^n c : n \geq 0\} \cup \{\varepsilon\}$. Since M satisfies the all-final restriction, all states are final. Therefore, if in the first nonempty reading step the ▶-head reads u and the ◀-head reads v, then uv or vu belongs to $L(M)$. Consider any k such that $k > max(\{|uv| : \delta(q, u, v, s) \neq \emptyset,\ u, v \in V^*\})$. Let $w = ca^k cb^k c$. It is not hard to see that for any first nonempty reading step on w (which reads u and v) must hold $|uv|_c \leq 2$. However, for all $w' \in (L(M) - \{\varepsilon\})$ holds $|w'|_c = 3$. Therefore, if M accepts w, it also accepts $uv \notin L(M)$ or $vu \notin L(M)$. But that is a contradiction with the assumption that M exists. □

Theorem 33 ***F JWK*** $\subset$ ***JWK***.

Proof This theorem follows directly from Theorem 9 and Lemma 32. □

Proposition 34 ***F JWK*** *and* ***LIN*** *are incomparable.*

Proof **LIN** $\not\subseteq$ **F JWK** follows from Lemma 32. **F JWK** $\not\subseteq$ **LIN** follows from Example 4. Lastly, **F JWK** and **LIN** contain the language $\{a\}^*$. □

Lemma 35 *For every $L \in$ **F JWK** holds $\varepsilon \in L$.*

Proof Consider any **F** jumping $5' \to 3'$ WK automaton $M = (V, Q, q_0, F, \delta)$. Since $Q = F$, q_0 is a final state and $(q_0, \oplus, \varepsilon, \varepsilon, \varepsilon) \curvearrowright (q_0, \ominus, \varepsilon, \varepsilon, \varepsilon)$ can be done with a $\oplus$-jumping step; thus, $\varepsilon \in L(M)$. □

Proposition 36 ***F JWK*** *and* ***FIN*** *are incomparable.*

Proof **FIN** $\not\subseteq$ **F JWK** follows from Lemma 35. **F JWK** $\not\subseteq$ **FIN** follows from Example 4. Lastly, it is trivial to construct an **F** jumping $5' \to 3'$ WK automaton with two states that accepts the finite language $\{\varepsilon, a\}$. □

Theorem 37 $\boldsymbol{FIN}_{\varepsilon\text{-}inc} \subset$ ***F JWK***.

Proof First, **F JWK** $\not\subseteq$ $\mathbf{FIN}_{\varepsilon\text{-inc}}$ follows from Example 4. Second, by construction. Consider any alphabet V and $L = \{x_1, \ldots, x_n\} \in \mathbf{FIN}_{\varepsilon\text{-inc}}$ such that $x_i \in V^*$, $i = 1, \ldots, n$, $n \geq 1$. Define the **F** jumping $5' \to 3'$ WK automaton $M = (V, \{q_0, q_f\}, q_0, \{q_0, q_f\}, \delta)$, where δ is constructed in the following way: For each $x \in L$, set $\delta(q_0, x, \varepsilon, \oplus) = \{q_f\}$. It is clear that $L(M) = L$. □

Example 38 Consider the following **F** (in fact, even **N**) jumping $5' \to 3'$ WK automaton $M = (\{a, b, c\}, \{s\}, s, \{s\}, \delta)$ with the state transition function δ:

$$\delta(s, a, b, \oplus) = \{s\}, \quad \delta(s, a, b, \ominus) = \{s\},$$
$$\delta(s, cc, \varepsilon, \oplus) = \{s\}, \quad \delta(s, \varepsilon, cc, \oplus) = \{s\}.$$

This is a slightly modified version of Example 25 where the substring cc can occur arbitrarily many times. Therefore, $L(M) = \{w \in \{a, b, cc\}^* : |w|_a = |w|_b\}$.

Proposition 39 ***F JWK*** *and* ***1 JWK*** *are incomparable.*

Proof First, **1 JWK** $\not\subseteq$ **F JWK** follows from Theorem 31 and Lemma 32. Next, the proof structure of Lemma 27 can be straightforwardly applied on Example 38; therefore, **F JWK** $\not\subseteq$ **1 JWK**. Lastly, both families contain $\{a\}^*$. □

6.4 On the stateless restriction

Lemma 40 *There is no* ***N*** *jumping* $5' \to 3'$ *WK automaton* $M = (V, \{q_0\}, q_0, \{q_0\}, \delta)$ *such that* $L(M) \in$ ***FIN*** *and* $L(M) \neq \{\varepsilon\}$.

Proof First, due to Lemma 35, $L(M)$ must always contain ε. Second, by contradiction, assume that there is a **N** jumping $5' \to 3'$ WK automaton M_2 such that $L(M_2) \in$ **FIN** and $L(M_2)$ contains a nonempty string. Since there is only one state, any $\oplus/\ominus$-reading step can be repeated arbitrarily many times. Therefore, if in the first nonempty reading step the ▶-head reads u and the ◀-head reads v, then $u^i v^i$ or $v^i u^i$ belongs to $L(M_2)$ for all $i \geq 1$. Thus, if M_2 accepts a nonempty string, $L(M_2) \notin$ **FIN**. But that is a contradiction with the assumption that M_2 exists. Consequently, if $L(M) \in$ **FIN**, $L(M) = \{\varepsilon\}$. □

Theorem 41 ***N JWK*** $\subset$ ***F JWK***.

Proof From Lemma 23, **N JWK** $\subseteq$ **F JWK**. **F JWK** $\not\subseteq$ **N JWK** follows from Theorem 37 and Lemma 40. □

Proposition 42 ***N JWK*** *is incomparable with* ***LIN***, ***FIN***, *and* ***FIN***$_{\varepsilon\text{-inc}}$.

Proof **LIN**, **FIN**, **FIN**$_{\varepsilon\text{-inc}} \not\subseteq$ **N JWK** follows from Lemma 40. **N JWK** $\not\subseteq$ **LIN**, **FIN**, **FIN**$_{\varepsilon\text{-inc}}$ follows from Example 4. Lastly, **N JWK** and **LIN** contain the language $\{a\}^*$, and **N JWK**, **FIN**, and **FIN**$_{\varepsilon\text{-inc}}$ contain the language $\{\varepsilon\}$. □

Proposition 43 ***N JWK*** *and* ***1 JWK*** *are incomparable.*

Proof First, **1 JWK** $\not\subseteq$ **N JWK** follows from Theorem 31 and Lemma 40. Second, **N JWK** $\not\subseteq$ **1 JWK** follows from Example 38 and the proof of Proposition 39. Lastly, both families contain the language $\{a\}^*$. □

6.5 On the combined restrictions

Proposition 44 ***FS JWK*** $\subset$ ***F JWK***.

Proof Let $L = \{cca^ncc : n \geq 0\} \cup \{\varepsilon\}$. It is trivial to construct an **F** jumping $5' \to 3'$ WK automaton that accepts L. However, there is no **FS** jumping $5' \to 3'$ WK automaton that accepts L. By contradiction. Assume that there is an **FS** jumping $5' \to 3'$ WK automaton M such that $L(M) = L$. Using the basic premise of Lemma 32, all c's has to be read with the first nonempty reading step. Nonetheless, a single head cannot read all c's in one step if they are arbitrarily far away from each other—a contradiction with the assumption that M exists. □

Theorem 45 ***FIN***$_{\varepsilon\text{-inc}} \subset$ ***FS JWK***.

Proof **FS JWK** $\not\subseteq$ **FIN**$_{\varepsilon\text{-inc}}$ follows from $\{a\}^* \in$ **FS JWK**. The rest of the proof is analogous to Theorem 37. □

Example 46 Consider the following **FS** jumping $5' \to 3'$ WK automaton $M = (\{a, b, c\}, \{s, p\}, s, \{s, p\}, \delta)$ with the state transition function δ:

$$\begin{aligned}
\delta(s, a, \varepsilon, \oplus) &= \{p\}, \quad \delta(p, \varepsilon, b, \oplus) = \{s\},\\
\delta(s, a, \varepsilon, \ominus) &= \{p\}, \quad \delta(p, \varepsilon, b, \ominus) = \{s\},\\
\delta(s, cc, \varepsilon, \oplus) &= \{s\}, \quad \delta(s, \varepsilon, cc, \oplus) = \{s\},\\
\delta(p, cc, \varepsilon, \oplus) &= \{p\}, \quad \delta(p, \varepsilon, cc, \oplus) = \{p\}.
\end{aligned}$$

As a result, $L(M) = \{w \in \{a, b, cc\}^* : |w|_a = |w|_b \text{ or } |w|_a = |w|_b + 1\}$.
This automaton is just a combination of previous approaches from Examples 29 and 38. Note that $L(M)$ resembles the resulting language of Example 38.

Proposition 47 ***FS JWK*** *and* ***1 JWK*** *are incomparable.*

Proof First, **1 JWK** $\not\subseteq$ **FS JWK** follows from the language in the proof of Proposition 44. Second, the proof structure of Lemma 27 can be straightforwardly applied on Example 46; therefore, **FS JWK** $\not\subseteq$ **1 JWK**. Lastly, **FS JWK** and **1 JWK** contain the language $\{a\}^*$. □

Proposition 48 ***F1 JWK*** $\subset$ ***FS JWK***.

Proof From Lemma 23, **F1 JWK** $\subseteq$ **FS JWK**. It is trivial to construct an **FS** jumping $5' \to 3'$ WK automaton that accepts $\{aa\}^*$. However, there cannot be an **F1** jumping $5' \to 3'$ WK automaton that accepts only even-length inputs. □

Proposition 49 ***F1 JWK*** *and* ***LIN*** *are incomparable.*

Proof **LIN** $\not\subseteq$ **F1 JWK** follows from $\{aa\}^* \in$ **LIN**. Considering Example 46, there is an **F1** jumping $5' \to 3'$ WK automaton M such that $L(M) = \{w \in \{a,b\}^* : |w|_a = |w|_b \text{ or } |w|_a = |w|_b + 1\}$. Clearly, $L(M)$ is not a linear language. Lastly, **F1 JWK** and **LIN** contain the language $\{a\}^*$. □

Corollary 50 ***F1 JWK*** $\subset$ ***1 JWK***. □

Theorem 51 ***NS JWK*** $\subset$ ***REG***.

Proof **NS JWK** $\subseteq$ **REG** can be proven by construction. We show that for any **NS** jumping $5' \to 3'$ WK automaton we can construct a finite automaton that accepts the same language. Consider any **NS** jumping $5' \to 3'$ WK automaton $M = (V, \{q_0\}, q_0, \{q_0\}, \delta)$. The following claims hold:

Claim 1 Any $w \in L(M)$ can be expressed in the following special form $w = x_1 y_1' \dots \; x_n y_n' x_1' y_1 \dots x_m' y_m$, where $x_i, y_i', x_j', y_j \in V^*$, for all $i = 1, \dots, n$ and $j = 1, \dots, m$, for some $n, m \geq 1$, and for all x_i, y_i', x_j', y_j hold:

(i) either $x_i = \varepsilon$ or $\delta(q_0, x_i, \varepsilon, \oplus) = \{q_0\}$,
(ii) either $y_j = \varepsilon$ or $\delta(q_0, \varepsilon, y_j, \oplus) = \{q_0\}$,
(iii) either $x_j' = \varepsilon$ or $\delta(q_0, x_j', \varepsilon, \ominus) = \{q_0\}$,
(iv) either $y_i' = \varepsilon$ or $\delta(q_0, \varepsilon, y_i', \ominus) = \{q_0\}$.

Proof Due to the restrictions, parts (i), (ii), (iii), and (iv) cover all possible types of state transitions. The accepted input can be always divided into two parts, depending on the position where the heads of M meet each other when the automaton processes this input. The first part $x_1 y_1' \dots x_n y_n'$ is a combination of $\oplus$-readings with the ▶-head and $\ominus$-readings with the ◀-head. Likewise, the second part $x_1' y_1 \dots x_m' y_m$ is a combination of $\ominus$-readings with the ▶-head and $\oplus$-readings with the ◀-head. To get the uncertain reading order forced by the jumping steps, we also allow each part x_i, y_i', x_j', y_j to be empty. Therefore, all $w \in L(M)$ have to be able to satisfy this special form. ■

Claim 2 Any $w \in V^*$ that can be expressed in the previous special form belongs to $L(M)$.

Proof Considering the restrictions, M has only one state, and only one head can read in a step. Therefore, if there is a possible reading step, it can be used arbitrarily many times. Furthermore, the possible reading steps can change only when the heads meet each other. Also, since each head reads separately, there cannot be any dependence between the first and second part of the input in the special form. Consequently, any $w \in V^*$ that can be expressed in the form from Claim 1 has to belong to $L(M)$. ■

Considering both claims, it is easy to construct a finite automaton that accepts all inputs of this special form. **REG** $\not\subseteq$ **NS JWK** follows from Lemma 40. □

Proposition 52 ***N1 JWK*** $\subset$ ***NS JWK***.

Proof This proof is analogous to that of Proposition 48. □

Proposition 53 *The following relations hold:*
(i) ***NS JWK*** $\subset$ ***N JWK***;
(ii) ***NS JWK*** $\subset$ ***FS JWK***;
(iii) ***N1 JWK*** $\subset$ ***F1 JWK***.

Proof Examples 4 and 46 and Proposition 49 show that **N JWK**, **FS JWK**, and **F1 JWK** contain some non-regular languages. Considering Lemma 23 and Theorem 51, all three proposed relations directly follow. □

Proposition 54 ***FS JWK*** *and* ***N JWK*** *are incomparable.*

Proof First, **FS JWK** $\not\subseteq$ **N JWK** follows from Lemma 40. Second, let $L = \{a^n b^n : n \geq 0\}$. It is trivial to construct an **N** jumping $5' \to 3'$ WK automaton that accepts L. However, there is no **FS** jumping $5' \to 3'$ WK automaton that accepts L. By contradiction. Assume that there is an **FS** jumping $5' \to 3'$ WK automaton $M = (V, Q, q_0, F, \delta)$ such that $L(M) = L$. Due to the restrictions, if a head of M reads u in a step, it must hold that $|u|_a = |u|_b$. Otherwise, there would be $w' \in L(M)$ such that $|w'|_a \neq |w'|_b$. Consider any k such that $k > max(\{|v_1 v_2| : \delta(q, v_1, v_2, s) \neq \emptyset,\ v_1, v_2 \in V^*\})$. Let $w = a^{2k} b^{2k}$. Clearly, when M processes w, each head can read u such that $|u|_a = |u|_b$ no more than once. However, these balanced steps can therefore process only less than $2k$ symbols. Consequently, if M accepts w, it also accepts some $w' \notin L$—a contradiction with the assumption that M exists. Therefore, **N JWK** $\not\subseteq$ **FS JWK**. Lastly, **FS JWK** and **N JWK** contain the language $\{a\}^*$. □

Proposition 55 ***F1 JWK*** *and* ***NS JWK*** *are incomparable.*

Proof First, **F1 JWK** $\not\subseteq$ **NS JWK** follows from Lemma 40. Second, **NS JWK** $\not\subseteq$ **F1 JWK** follows from $\{aa\}^* \in$ **NS JWK**. Lastly, both families contain the language $\{a\}^*$. □

Proposition 56 ***REG*** *is incomparable with* ***F JWK***, ***N JWK***, ***FS JWK***, *and* ***F1 JWK***.

Proof First, Examples 4 and 46 and Proposition 49 show that **F JWK**, **N JWK**, **FS JWK**, and **F1 JWK** contain some non-regular languages. Second, let $L = \{ca^n cb^m c : n, m \geq 0\} \cup \{\varepsilon\}$. L is clearly a regular language. Considering the proof of Lemma 32 and the previous results, we can easily see that **F JWK**, **N JWK**, **FS JWK**, and **F1 JWK** cannot contain L. Lastly, all families contain the language $\{a\}^*$. □

Proposition 57 ***FIN*** *is incomparable with* ***FS JWK***, ***F1 JWK***, ***NS JWK***, *and* ***N1 JWK***.

Proof Considering previous results. First, **FS JWK**, **F1 JWK**, **NS JWK**, and **N1 JWK** cannot contain $\emptyset$. Second, **FS JWK**, **F1 JWK**, **NS JWK**, and **N1 JWK** contain $\{a\}^*$. Lastly, all families contain $\{\varepsilon\}$. □

Proposition 58 ***FIN***$_{\varepsilon\text{-}inc}$ *is incomparable with* ***F1 JWK***, ***NS JWK***, *and* ***N1 JWK***.

Proof Considering previous results. First, **F1 JWK**, **NS JWK**, and **N1 JWK** cannot contain $\{\varepsilon, aa\}$. Second, **F1 JWK**, **NS JWK**, and **N1 JWK** contain $\{a\}^*$. Lastly, all families contain $\{\varepsilon\}$. □

All the obtained results comparing the accepting power of different variations of jumping $5' \to 3'$ WK automata are summarized in Figure 1.

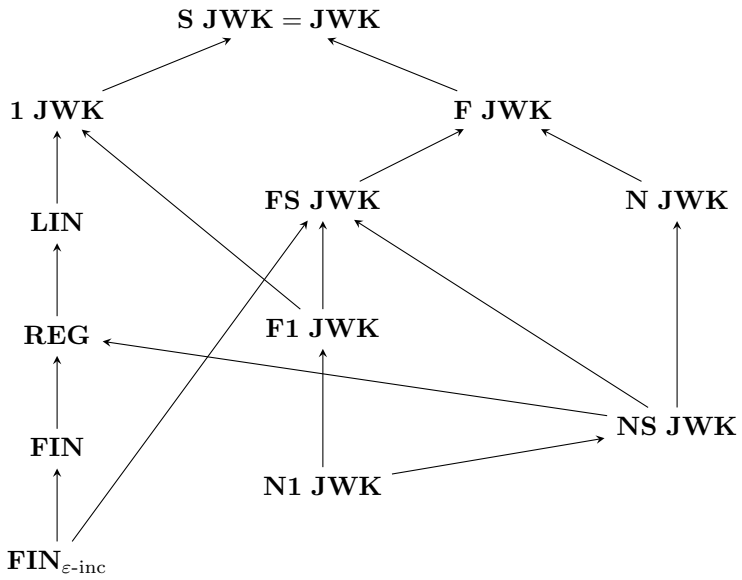

Fig. 1 A hierarchy of language families closely related to the unrestricted and restricted variations of jumping $5' \to 3'$ WK automata is shown. If there is an arrow from family X to family Y in the figure, then $X \subset Y$. Furthermore, if there is no path (following the arrows) between families X and Y, then X and Y are incomparable.

7 Conclusion

The results clearly show that, with the addition of the jumping mechanism into the model, the accepting power has been increased above sensing $5' \to 3'$ WK automata. The model is now able to accept some nonlinear and even some non-context-free languages. On the other hand, the jumping movement of the heads is restricted compared to jumping finite automata, and this limits its capabilities to accept languages that require a more sophisticated discontinuous information processing. Considering the comparison with full-reading sensing $5' \to 3'$ WK automata, the results are not yet clear and further research is required. However, we know that there are some languages, like $\{a^n b^n c^n : n \geq 0\}$, that cannot be

accepted by jumping $5' \to 3'$ WK automata and that are accepted by full-reading sensing $5' \to 3'$ WK automata (see [18–21]).

If we compare the hierarchies of language families related to the restricted variations of jumping $5' \to 3'$ WK automata and sensing $5' \to 3'$ WK automata (see [20,21,23]), there are several noticeable remarks. Most importantly, the 1-limited restriction (**1**) has a negative impact on the accepting power, which is usually not the case in sensing $5' \to 3'$ WK automata. Secondly, when several restrictions are combined together, the hierarchy structure resembles the alternative structure of sensing $5' \to 3'$ WK automata without the sensing distance. Lastly, almost all restricted variations, with the exception of **NS** and **N1**, are still able to accept some nonlinear languages, which cannot be accepted by any variation of sensing $5' \to 3'$ WK automata.

Finally, the reader may notice that the $\ominus$-jumping can be used only in situations where it is forced by the current configuration. Jumping finite automata usually immediately erase symbols from the configuration and do not use the auxiliary symbol #. It is therefore a question for future research whether we can safely remove this part from the model and keep the accepting power intact.

Acknowledgements This work was supported by The Ministry of Education, Youth and Sports of the Czech Republic from the National Programme of Sustainability (NPU II); project IT4Innovations excellence in science - LQ1602.

References

1. Beier, S., Holzer, M.: Decidability of right one-way jumping finite automata. In: DLT 2018, *LNCS*, vol. 11088, pp. 109–120 (2018). DOI https://doi.org/10.1007/978-3-319-98654-8_9
2. Beier, S., Holzer, M., Kutrib, M.: Operational state complexity and decidability of jumping finite automata. In: DLT 2017, *LNCS*, vol. 10396, pp. 96–108 (2017). DOI https://doi.org/10.1007/978-3-319-62809-7_6
3. Chigahara, H., Fazekas, S.Z., Yamamura, A.: One-way jumping finite automata. Int. J. Found. Comput. Sci. **27**(03), 391–405 (2016). DOI https://doi.org/10.1142/S0129054116400165
4. Fazekas, S.Z., Hoshi, K., Yamamura, A.: Enhancement of automata with jumping modes. In: AUTOMATA 2019, *LNCS*, vol. 11525, pp. 62–76 (2019). DOI https://doi.org/10.1007/978-3-030-20981-0_5
5. Fazekas, S.Z., Yamamura, A.: On regular languages accepted by one-way jumping finite automata. In: Eighth Workshop on Non-Classical Models of Automata and Applications (NCMA 2016) Short papers, pp. 7–14 (2016)
6. Fernau, H., Paramasivan, M., Schmid, M.L.: Jumping finite automata: Characterizations and complexity. In: CIAA 2015, *LNCS*, vol. 9223, pp. 89–101. Springer (2015). DOI https://doi.org/10.1007/978-3-319-22360-5_8
7. Fernau, H., Paramasivan, M., Schmid, M.L., Vorel, V.: Characterization and complexity results on jumping finite automata. Theor. Comput. Sci. **679**, 31–52 (2017). DOI https://doi.org/10.1016/j.tcs.2016.07.006
8. Křivka, Z., Meduna, A.: Jumping grammars. Int. J. Found. Comput. Sci. **26**(6), 709–731 (2015). DOI https://doi.org/10.1142/S0129054115500409
9. Kocman, R., Křivka, Z., Meduna, A.: On double-jumping finite automata. In: Eighth Workshop on Non-Classical Models of Automata and Applications (NCMA 2016), *OCG*, vol. 321, pp. 195–210 (2016)
10. Kocman, R., Křivka, Z., Meduna, A.: On double-jumping finite automata and their closure properties. RAIRO-Theor. Inf. Appl. **52**(2-3-4), 185–199 (2018). DOI https://doi.org/10.1051/ita/2018013
11. Kocman, R., Meduna, A.: On parallel versions of jumping finite automata. In: SDOT 2015, *AICS*, vol. 511, pp. 142–149 (2016). DOI https://doi.org/10.1007/978-3-319-46535-7_12

12. Kocman, R., Nagy, B., Křivka, Z., Meduna, A.: A jumping $5' \to 3'$ Watson-Crick finite automata model. In: Tenth Workshop on Non-Classical Models of Automata and Applications (NCMA 2018), *OCG*, vol. 332, pp. 117–132 (2018)
13. Madejski, G.: Jumping and pumping lemmas and their applications. In: Eighth Workshop on Non-Classical Models of Automata and Applications (NCMA 2016) Short papers, pp. 25–33 (2016)
14. Meduna, A.: Automata and Languages: Theory and Applications. Springer, London (2000)
15. Meduna, A., Soukup, O.: Jumping scattered context grammars. Fundam. Inf. **152**(1), 51–86 (2017). DOI https://doi.org/10.3233/FI-2017-1512
16. Meduna, A., Zemek, P.: Jumping finite automata. Int. J. Found. Comput. Sci. **23**(7), 1555–1578 (2012). DOI https://doi.org/10.1142/S0129054112500244
17. Meduna, A., Zemek, P.: Regulated Grammars and Automata. Springer (2014)
18. Nagy, B.: On $5' \to 3'$ sensing Watson-Crick finite automata. In: The 13th International Meeting on DNA Computing (DNA13), pp. 327–336 (2007)
19. Nagy, B.: On $5' \to 3'$ sensing Watson-Crick finite automata. In: DNA 13, *LNCS*, vol. 4848, pp. 256–262 (2008). DOI https://doi.org/10.1007/978-3-540-77962-9_27
20. Nagy, B.: $5' \to 3'$ sensing Watson-Crick finite automata. In: G. Fung (ed.) Sequence and Genome Analysis II – Methods and Applications, pp. 39–56. iConcept Press (2010)
21. Nagy, B.: On a hierarchy of $5' \to 3'$ sensing Watson-Crick finite automata languages. J. Log. Comput. **23**(4), 855–872 (2013). DOI https://doi.org/10.1093/logcom/exr049
22. Nagy, B., Otto, F.: Two-head finite-state acceptors with translucent letters. In: SOFSEM 2019, *LNCS*, vol. 11376, pp. 406–418 (2019). DOI https://doi.org/10.1007/978-3-030-10801-4_32
23. Nagy, B., Parchami, S., Mir-Mohammad-Sadeghi, H.: A new sensing $5' \to 3'$ Watson-Crick automata concept. In: AFL 2017, *EPTCS*, vol. 252, pp. 195–204 (2017). DOI https://doi.org/10.4204/EPTCS.252.19
24. Paun, G., Rozenberg, G., Salomaa, A.: DNA Computing: New Computing Paradigms. Springer-Verlag Berlin Heidelberg (1998)
25. Rozenberg, G., Salomaa, A.: Handbook of Formal Languages, Vol. 2: Linear Modeling: Background and Application. Springer-Verlag (1997)
26. Vorel, V.: On basic properties of jumping finite automata. Int. J. Found. Comput. Sci. **29**(1), 1–15 (2018). DOI https://doi.org/10.1142/S0129054118500016
27. Wood, D.: Theory of Computation: A Primer. Addison-Wesley, Boston (1987)