Deterministic and Nondeterministic Sensing $5' \rightarrow 3'$ Watson-Crick Automata without Sensing Parameter

Shaghayegh Parchami

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Applied Mathematics and Computer Science

Eastern Mediterranean University
April 2020
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Ali Hakan Ulusoy
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Doctor of Philosophy in Applied Mathematics and Computer Science.

Prof. Dr. Nazim Mahmudov
Chair, Department of Mathematics

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Doctor of Philosophy in Applied Mathematics and Computer Science.

Prof. Dr. Benedek Nagy
Supervisor

Examining Committee

1. Prof. Dr. Rashad Aliyev
2. Prof. Dr. Gergely Kovács
3. Prof. Dr. Elimhan Mahmudov
4. Prof. Dr. Benedek Nagy
5. Asst. Prof. Dr. Müge Saadetoğlu

ABSTRACT

Watson-Crick (abbreviated as WK) finite automata are working on double stranded DNA molecule that is also called Watson-Crick tape. Subsequently, these automata have two reading heads, one for each strand. While in traditional WK automata both heads read the whole input in the same physical direction, in $5' \to 3'$ WK automata the heads start from the two extremes (say $5'$ end of the strands) and read the input in opposite direction. In sensing $5' \to 3'$ WK automata the process on the input is finished when the heads meet. Since the heads of a WK automaton may read longer strings in a transition, in previous models a so-called sensing parameter took care for the proper meeting of the heads (not allowing to read the same positions of the input in the last step). We have investigated a new model which works without the sensing parameter (it is done by an appropriate change of the concept of the configuration). In this thesis, the nondeterministic and deterministic automata in the new model are studied. Consequently, for the nondeterministic part, the accepted language classes of variants are changed and various hierarchy results are proven. For the deterministic part, it is proven to accept the language class 2detLIN defined by the deterministic variant of the earlier version. However, using some of the restricted variants, e.g., all-final automata, the classes of the accepted languages are changed showing a finer hierarchy inside the class of linear context-free languages, hierarchy.

Keywords: nondeterministic Watson-Crick automata, deterministic Watson-Crick automata, $5' \to 3'$ WK automata, finite automata, deterministic computations, linear context-free languages

ÖZ

Watson-Crick (WK olarak kısaltılmıştır) sonlu otomataları, Watson-Crick bandı olarak da adlandırılan çift sarmallı DNA molekülü üzerinde çalışmaktadır. Bundan dolayı, bu otomatalar her bir tel için birer tane olmak üzere iki okuma kafasına sahiptir. Geleneksel WK otomatlarında her iki kafa da tüm girişi aynı fiziksel yönde okurken, $5' \to 3'$ WK otomatlarında kafalar iki uçtan başlar (iplikçiklerin 5 ′ ucu gibi) ve girişi ters yönde okur. $5' \to 3'$ WK otomat algılamasında, kafalar buluştuğunda girişteki işlem tamamlanır. Bir WK otomatının kafaları bir geçişte daha uzun dizeler okuyabildiğinden, önceki modellerde, kafaların düzgün bir şekilde toplanması için bir algılama parametresi kullanıldı (son adımda girişin aynı pozisyonlarını okumaya izin vermiyor). Algılama parametresi olmadan çalışan yeni bir model araştırdık (konfigürasyon kavramının uygun bir şekilde değiştirilmesi ile yapılır). Bu tezde yeni modeldeki deterministik ve deterministik olmayan otomatlar incelenmiştir. Sonuç olarak, deterministik olmayan kısım için, varyantların kabul edilen dil sınıfları değiştirilir ve çeşitli hiyerarşi sonuçları kanıtlanır. Deterministik kısım için, önceki versiyonun deterministik varyantı tarafından tanımlanan 2detLIN dil sınıfını kabul ettiği kanıtlanmıştır. Bununla birlikte, kısıtlı varyantların bazıları, örn., Tümüyle son otomatik veriler kullanılarak, kabul edilen dillerin sınıfları, doğrusal bağlamsız diller, hiyerarşi sınıfında daha ince bir hiyerarşi gösterecek şekilde değiştirilir.

Anahtar Kelimeler: Deterministik olmayan Watson-Crick otomatları, deterministik Watson-Crick otomatları, $5' \to 3'$ WK otomatları, sonlu otomatlar, deterministik hesaplamalar, doğrusal bağlamdan bağımsız diller

This thesis is dedicated to my husband Dr. Hamid Mirmohammadsadeghi

and my precious dauther Liana Mirmohammadsadeghi with love.

ACKNOWLEDGMENT

I would like to express my deep gratitude and great admiration to my esteemed supervisor Prof. Dr. habil. Benedek Nagy for his immense guidance, incredible patience, and support throughout the research. I am really lucky to be his student and thankful for that I have learnd a lot. Without his supervision and guidance this thesis would not have been accomplished. Worth extremly regard to Prof. Rashad Aliyev and Asst. Prof. Dr. Müge Saadetoğlu for serving me as committee members and their valuable suggestions during the past years.

I would like to wholeheartedly thank all my beloved family, especially my lovely mother Mrs. Parvin Naghdi for her unbounded love and two younger brothers, Dr. Mohsen Parchami and Dr. Milad Parchami who supported me and devoted their love in my whole time.

Last but not the least, I owed profound gratitude to my husband, Dr. Hamid Mirmohammadsadeghi, for his immeasurable giving, great sacrifice, ultimate love and unquestioning support, to my dearest daughter for simply being there and source of all love and contentment in me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

From the end of the last century, DNA computing has appeared as a relatively new computational paradigm [1, 26]. In contrast, automata theory is from the middle of the last century and it is one of the bases of computer science. An interesting combination of these two fields, the theory of Watson-Crick-automata (abbreviated as WK automata, from the initial of the name Watson and the last letter of the name Crick), was introduced in [5] as a branch of DNA computing. They have important relations to formal language and automata theory. To read more about these automata the book [26] and the survey [3] are recommended. WK automata work on double-stranded tapes called Watson-Crick tape (i.e., DNA molecule), whose strands are scanned separately by read only heads. The symbols in the corresponding cells of the double-stranded tapes are related by (the Watson-Crick) complementarity relation. Restricted classes having either or both restriction on the states, e.g., all states are final, or on the transitions, e.g., only one of the heads can read in a transition, are analysed. The relationships between various classes of the Watson-Crick automata are investigated in [5, 10, 26]. About applications of WK automata we may refer to [28]. The two strands of a DNA molecule have opposite $5' \to 3'$ orientation. Considering the reverse and the $5' \to 3'$ variants, they are more realistic in the sense, that both heads use the same biochemical direction (that is opposite physical directions) [5, 12–14]. A WK automaton is sensing if it has the information whether the heads are at the same position. Some variants of the $5' \to 3'$ Watson-Crick automaton with sensing parameter, i.e., with a feature which tells whether the upper and the lower heads are within a fixed small distance (or meet at the same position) are discussed in [15, 16, 18]. The heads of these automata start from the opposite ends of the input, assuming the complementary relation to be bijective (as it is in the nature), the automaton already has information about the whole input at the point where the heads meet. Consequently, the automaton makes the decision on acceptance at that point and the process on the input is finished. It was shown that the linear context-free languages and some of their subclasses (e.g., the class of even linear languages) can be characterised by these models [12–18]. Since the heads of a WK automaton may read longer strings in a transition, in these models the sensing parameter took care of the proper meeting of the heads sensing if the heads are close enough to meet in the next transition. This parameter could also be used to deny acceptance of some strings, e.g., by not allowing to read the last letter(s) to finish the process in that way. This idea led to the fact that there is no difference of the language classes accepted by arbitrary and all-final automata. The motivation of the new model, recently introduced in [22] is to erase the rather artificial term of sensing parameter from the model. Here, the accepted language classes of the new model are analyzed. Variations such as all-final, simple, 1-limited, and stateless $5' \rightarrow 3'$ Watson-Crick automata are also detailed. The deterministic counterpart is also investigated [24]. As one of the main results, we prove that the new deterministic model accept exactly the same class of languages, namely 2detLIN, that is accepted by the deterministic variant of the model with sensing parameter. This is an interesting class of languages containing, e.g., all even linear languages [2, 14, 16]. Even linear and other special subclasses of linear context-free languages are of recent interests of various research papers, see, e.g., [7, 11, 30], due to their learnability property in formal languages [4, 29]. Here, the accepted language classes of various restricted classes of WK automata and their relations are analyzed showing a finer hierarchy than the previous model has provided. A related model, deterministic stateless $5' \rightarrow 3'$ WK counter machine is analysed in [6]. Recently other 2-head automata models working in a similar manner as $5' \rightarrow 3'$ WK automata, with jumping feature [9], with tranclucent letter [21], with pushdown stack [19], and with output [20] were also investigated.

1.1 Layout of thesis

This thesis include of five chapters. Chapter 2 describes briefly the structure of DNA, Chomsky hierarchy. Also, the basic definitions and notations are explained. Chapter 3 represents nondeterministic sensing $5' \rightarrow 3'$ Watson-Crick automata which work without sensing parameter and all language classes accepted by various types of this model. Chapter 4 describes the deterministic sensing $5' \rightarrow 3'$ Watson-Crick automata without sensing parameter and the classes of the accepted languages by various types of this model. Chapter 5 is dedicated to conclusion and future work.

Chapter 2

BASIC DEFINITIONS AND NOTATIONS

In this chapter, we describe the structure of DNA molecule briefly. After we recall some concepts of formal language theory. In the last section, we determine some basic definitions and notations for our model.

2.1 DNA's structure

Strung monomers which are called deoxyribonucleotides construct a polymer named a DNA. For simplicity, we use "nucleotide" term rather than deoxyribonucleotides. Each nucleotide includes three elements: a sugar, a phosphate group and a nitrogenous base. The sugar consists of five carbon atoms which are numbered from $1'$ through $5'$. The phosphate group and the base are attached to the $5'$ and $1'$ carbon, respectively. There exists a hydroxyl group (OH) connected to the $3'$ carbon atom in the sugar structure. The diversity of nucleotides refers to their base. There exist four possible bases: Adenine (A), Cytosine (C), Guanine (G) and Thymine (T). The figure 2.1 depicts the structure of a nucleotide (in a very simplified model) where B shows one the four bases (C, G, T, A), P illustrates the phosphate group and five carbon atoms of sugar base which are numbered from $1'$ to $5'$.

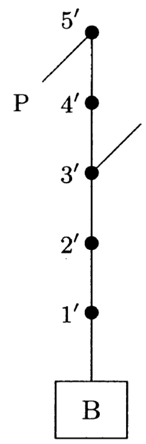


Figure 2.1: The structure of the nucleotide.

Two nucleotides can be connected in two different ways:

1. The 5′-phosphate group of a nucleotide is connected to the 3′ hydroxyl group of another by covalent bond (one of the strongest chemical bonds). It is shown in figure 2.2. Note that the connection of the 3′-OH group of one nucleotide to the 5′-phosphate group of the next one gives the directionality to the molecule; it can be the $3' - 5'$ direction or the $5' - 3'$ direction. The directionality is decisive to figure out the processing and the functionality of DNA.

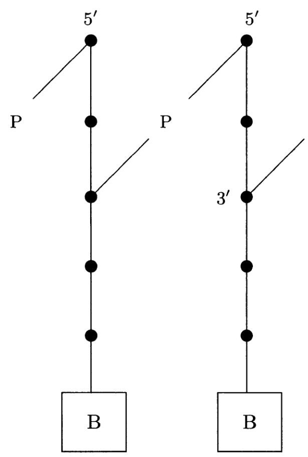


Figure 2.2: Phosphodiester bond.

2. The bases of two nucleotides can be interacted together by hydrogen bond which

is weaker than covalent bond. The bonding leads to the following constraint on the base pairing: A and T can be connected through two hydrogen bonds, while C and G can be connected by three hydrogen bonds. Other pairings are not possible. Therefore, the pairing between C and G is stronger than the pairing between A and T. The figure 2.3 shows the pairing principle.

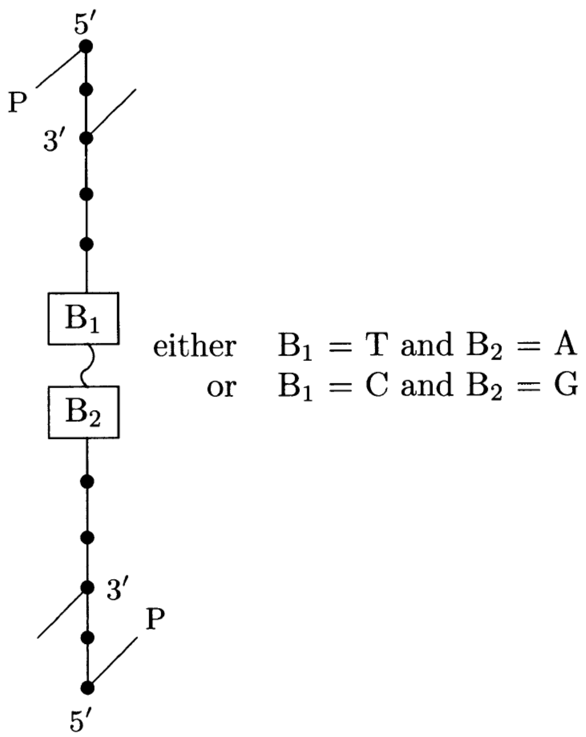


Figure 2.3: Hydrogen bond.

The rule of pairing called as the Watson-Crick complementarity (James D. Watson and Francis H. C. Crick who obtained the Nobel Prize for showing the DNA's double helix structure). This rule expresses that one nucleotide can be joint to another one which is its complement. The figure 2.4 represents the Watson–Crick complementarity rule for bonding two single stranded molecules by hydrogen bonds. This fact means that in the double stranded molecule two single strands must be in the opposite direction: the $5'$ direction of one strand in one nucleotide is connected to the $3'$ end of another strand of the next nucleotide. It is standard convention that the upper strand moves from $5'$ to $3'$ direction (left to right) and the lower strand moves from $3'$ to $5'$ direction (right to left).

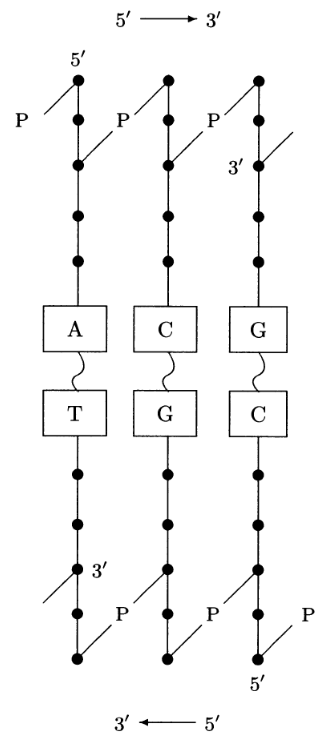


Figure 2.4: Forming double strands.

2.2 Formal languages and automata theory

Now some language families related to Chomsky hierarchy are recalled; for full definitions, readers refer to [8, 27]. A grammar is a quadruple $G = (N, T, S, P)$, where N, T are called the non-terminal and terminal alphabet, $S \in N$ is the initial letter and P is called a finite set of production rules of $(N \cup T)^* N (N \cup T)^* \times (N \cup T)^*$. A rule (v, w) of P is written in the form $v \to w$. For $v, w \in (N \cup T)^*$ we write:

$v \Rightarrow w$ if and only if there exist $v_1 v_2 v' w' \in (N \cup T)^*$ such that $v = v_1 v' v_2, w = v_1 w' v_2$ and $v' \to w' \in P$. A grammar G generates the language L consists of set of terminal words which are derived from the initial letter $L(G) = \{w | S \Rightarrow^* w \wedge w \in T^*\}$. We denote the empty word by λ. Two grammars are equivalent if they can generate the same language modulo λ. On the other hand $G_1 = G_2$ if $L(G_1) - \{\lambda\} = L(G_2) - \{\lambda\}$. We can classify the Chomsky grammars according to the form of their rules [26].

- **Monotonous:** if for all $v \to w \in P$ we have $|v| \leq |w|$.
- **Context-Sensitive grammars (CS):** if each rule holds the form $uAv \to uwv$ with $A \in N, u, v, w \in (N \cup T)^*, w \neq \lambda$.
- **Context-Free grammars (CF):** if each rule holds the form $A \to v$ with $A \in N$ and $v \in (N \cup T)^*$.
  - Linear grammars (LIN): every rule holds the form $A \to v, A \to vBw; A, B \in N; v, w \in T^*$.
    * Right linear grammars: if each rule $v \to w \in P$ has $v \in N$ and $w \in T^* \cup T^*N$.
    * Left linear grammars: if every rule $v \to w \in P$ has $v \in N$ and $w \in T^* \cup NT^*$.
- **Regular grammars (REG):** if each rule has the following forms $A \to w, A \to wB$; where $A, B \in N$ and $w \in T^*$.

The monotonous, CS, CF, and REG are also known as type 0, type 1, type 2, and type 3, respectively.

Note that, the class of linear context-free grammars generates the family LIN of linear context-free languages. In general, for a generating or accepting device A, we assign its generated or accepted language $L(A)$.

2.3 Definitions, preliminaries

In this section, we follow the basic definitions and notations of a Watson-Crick automaton, after which we recall some restricted variants of Watson-Crick automata definition of sensing $5' \to 3'$ Watson-Crick automaton with sensing parameter and the concept of determinism.

We assume that the reader is familiar with basic concepts of formal languages and automata, otherwise she or he is referred, e.g., to [8, 27]. The set of non-negative integers is denoted by $\mathbb{N}$.

Now we describe the automata model we are interested in. The two strands of the DNA molecule have opposite $5' \to 3'$ orientations. This proposes taking into account a variant of Watson-Crick finite automata that parse the two strands of the Watson-Crick tape in opposite directions. Figure 2.5 indicates the initial configuration of such an automaton on the left. The $5' \to 3'$ WK automaton is sensing, if the heads sense that they are meeting. We are working with models that finish the computing process at that phase. In Figure 2.5, this moment can be seen on the right. We note that there are also models which are continuing the process and they can also accept some non-context-free languages.

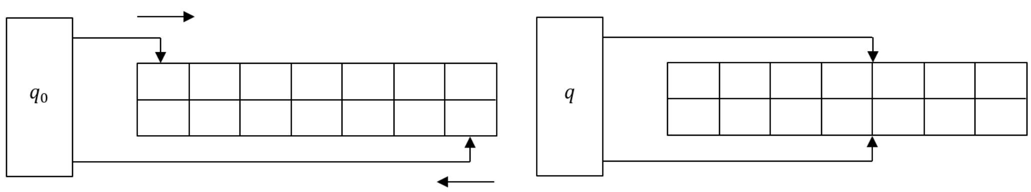


Figure 2.5: A sensing $5' \to 3'$ WK automaton in the initial configuration and in an accepting configuration (with a final state q).

Here, we follow the definition and description from [22]. (Later on we will also recall the earlier concept using sensing parameter from, e.g., [18] to show some results connecting the two models.)

Formally, a Watson-Crick automaton is a 6-tuple $M = (V, \rho, Q, q_0, F, \delta)$, where: V is the (input) alphabet, $\rho \subseteq V \times V$ denotes a complementarity relation, Q represents a finite set of states, $q_0 \in Q$ is the initial state, $F \subseteq Q$ is the set of final (also called accepting) states and δ is called transition mapping, it is of the form $\delta : Q \times \begin{pmatrix} V^* \\ V^* \end{pmatrix} \rightarrow 2^Q$, such that it is non empty only for finitely many triplets $(q, u, v), q \in Q, u, v \in V^*$. In sensing $5' \rightarrow 3'$ WK automata every pair of positions in the Watson-Crick tape is read by exactly one of the heads in an accepting computation, and therefore the complementarity relation cannot play importance, instead, for simplicity, we assume that it is the identity relation. Thus, it is more convenient to consider the input as a normal word instead the double stranded form. Note here that complementarity can be excluded from the traditional models as well, see [10] for details.

Since δ is not empty only for a finite set of triplets, there is/are a/some word(s) with maximal length that can be read in a transition by a given automaton. Consequently, let us define the radius r of an automaton by the maximum of the sum of the lengths of the substrings of the input that can be read by the automaton in a transition.

Further, a configuration of a Watson-Crick automaton is a pair (q, w) where q is the current state of the automaton and w is the part of the input word which has not been processed (read) yet. For $w', x, y \in V^*, q, q' \in Q$, we write a transition between two configurations as: $(q, xw'y) \Rightarrow (q', w')$ if and only if $q' \in \delta(q, x, y)$. We denote the reflexive and transitive closure of the relation $\Rightarrow$ (one step of a computation, that is, a transition) by $\Rightarrow^*$ (computation). Therefore, for a given $w \in V^*$, an accepting computation is a sequence of transitions $(q_0, w) \Rightarrow^* (q_F, \lambda)$, starting from the initial state and ending in a final state with no input left.

The language accepted by a WK automaton M is:

$$L(M) = \{w \in V^* | (q_0, w) \Rightarrow^* (q_F, \lambda), q_F \in F\}.$$

It was shown in [22] that the class of sensing $5' \rightarrow 3'$ WK automata that we have recalled accept exactly the class of linear context-free languages, LIN.

It is well known that over the unary alphabet the class of regular, linear and context-free languages coincide. Thus, in this thesis, we assume that the alphabet consists of at least two symbols.

The shortest nonempty word accepted by M is denoted by w_s, if it is uniquely determined. Otherwise we may use the notation w_s for any of them (in case there are more than one word with this condition).

There are some restricted variants of WK automata which are widely known:

F: all-final, i.e., with only final states: if $Q = F$;

N: stateless, i.e., with only one state: if $Q = F = \{q_0\}$;

S: simple (at most one head moves in a step) $\delta : (Q \times ((\lambda, V^*) \cup (V^*, \lambda))) \rightarrow 2^Q$;

1: 1-limited (exactly one letter is being read in each step) $\delta : (Q \times ((\lambda, V) \cup (V, \lambda))) \rightarrow 2^Q$.

Clearly, all **N** WK automata are **F** WK automata at the same time. Also, by definition, all **1** WK automata are, in fact, **S** WK automata also. However, since the restrictions **N** and **F** are about the states, and the restrictions **S** and **1** are about the length of the words that can be read in a transition (step of a computation), additional variants are also understood by using mixed constrains such as **F1**, **N1**, **FS**, **NS** WK automata.

Now, as an example, we show the language $L = \{a^n b^m \mid n, m \geq 0\}$ that can be accepted by an **N1** sensing $5' \to 3'$ WK automaton (Figure 2.6).

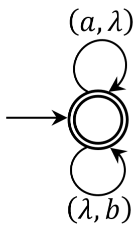


Figure 2.6: A sensing $5' \to 3'$ WK automaton of type **N1** accepting the language $\{a^n b^m \mid n, m \geq 0\}$.

So far, we have not given anything about determinism. We are using the following definition. If at each possible configuration at most one transition step is possible, then a WK automaton is deterministic. It means, a WK automaton is deterministic if and only if $\forall w \in V^*$ and $\forall q \in Q$ there exists at most one $w' \in V^*$ and $q' \in Q$ such that $(q, w) \Rightarrow (q', w')$.

We note that for the traditional WK automata reading both strands completely, there are various definitions of determinism (allowing also to play with the complementarity relation), but for our automata there is only one type of determinism.

Determinism is a feature that is orthogonal to the earlier special restrictions, thus we will study here, deterministic sensing $5' \to 3'$ WK automata (without any further restriction), deterministic **F** sensing $5' \to 3'$ WK automata, deterministic **N** sensing $5' \to 3'$ WK automata, deterministic **S** sensing $5' \to 3'$ WK automata, $\ldots$, deterministic **F1** sensing $5' \to 3'$ WK automata, etc.

Now the concept of sensing $5' \to 3'$ WK automaton with sensing parameter is recalled from [15, 18]. A 6-tuple $M = (V, \rho, Q, q_0, F, \delta_s)$ is a sensing $5' \to 3'$ WK automaton with sensing parameter where, V, ρ, Q, q_0 and F are exactly the same as in our model and δ_s is the transition mapping. It is defined using the sensing condition in the following way:

$\delta_s : \left(Q \times \begin{pmatrix} V^* \\ V^* \end{pmatrix} \times D \right) \to 2^Q$, where the sensing distance set is defined by $D = \{0, 1, \ldots, r, +\infty\}$ where r is the radius of the automaton. The set D gives the distance between two heads from 0 to r, and gives $+\infty$, when the distance of the two heads is more than r. On the other hand, by the set D, the automaton controls the appropriate meeting of the heads. Some transitions are allowed or denied depending on the actual distance of the positions of the heads (if it is not more than r) taking care of reading only string(s) having their length not more than the distance of the heads. In a sensing $5' \to 3'$ WK automaton with sensing parameter a configuration $\begin{pmatrix} w_1 \\ w_2 \end{pmatrix} (q, s) \begin{pmatrix} w_1' \\ w_2' \end{pmatrix}$ contains the state $q \in Q$, the sensing distance $s \in D$, where the input is $\begin{pmatrix} w_1 w_1' \\ w_2 w_2' \end{pmatrix}$ with the condition $w_1 w_1' = w_2 w_2'$. The part w_1 has been already processed by the left head (upper strand) and the part w_2' has been processed by the right head (lower strand). A transition between two configurations can be written as:

$$\begin{pmatrix} w_1 \\ w_2y \end{pmatrix} (q,+\infty) \begin{pmatrix} xw'_1 \\ w'_2 \end{pmatrix} \Rightarrow \begin{pmatrix} w_1x \\ w_2 \end{pmatrix} (q',s) \begin{pmatrix} w'_1 \\ yw'_2 \end{pmatrix} \text{ for } w_1,w_2,w'_1,w'_2,x,y \in V^*$$

with $|w_2y| - |w_1| > r$, $q,q' \in Q$ if and only if $w_1xw'_1 = w_2yw'_2$ and $q' \in \delta_s\left(q, \begin{pmatrix} x \\ y \end{pmatrix}, +\infty\right)$, further $s = \begin{cases} |w_2| - |w_1x|, & \text{if } |w_2| - |w_1x| \le r, \\ +\infty, & \text{otherwise,} \end{cases}$ and

$$\begin{pmatrix} w_1 \\ w_2y \end{pmatrix} (q,s_1) \begin{pmatrix} xw'_1 \\ w'_2 \end{pmatrix} \Rightarrow \begin{pmatrix} w_1x \\ w_2 \end{pmatrix} (q',s) \begin{pmatrix} w'_1 \\ yw'_2 \end{pmatrix} \text{ for } w_1,w_2,w'_1,w'_2,x,y \in V^*$$

with $0 \le |w_2y| - |w_1| = s_1 \le r$, and $q,q' \in Q$ if and only if $w_1xw'_1 = w_2yw'_2$ and $q' \in \delta_s\left(q, \begin{pmatrix} x \\ y \end{pmatrix}, s_1\right)$, further $s = |w_2| - |w_1x|$.

A sensing $5' \to 3'$ WK automaton M with sensing parameter accepts a string w if and only if $\begin{pmatrix} \lambda \\ w \end{pmatrix} (q_0,s_0) \begin{pmatrix} w \\ \lambda \end{pmatrix} \Rightarrow^* \begin{pmatrix} w'_1 \\ w'_1 \end{pmatrix} (q_f,0) \begin{pmatrix} w''_2 \\ w''_2 \end{pmatrix}$ for $q_f \in F$ where s_0 is $+\infty$ if $|w| > r$, otherwise it is $|w|$. The automaton M accepts the language $L(M)$ consisting of all such strings. Note that we have recalled and used here the original two-strand description of the sensing $5' \to 3'$ WK automata. In fact both this old model and the new model without the sensing parameter can be defined and used both on string languages and on double strands (where the complementarity is a bijection). (We are using the string language form for the new model only for brevity.) The deterministic sensing $5' \to 3'$ WK automata with sensing parameter was also defined. If there is at most one transition step that can occur in each configuration, then the automaton is deterministic. The language class that can be accepted by deterministic $5' \to 3'$ WK automata with sensing parameter is denoted by 2detLIN, as these are exactly those languages that are accepted by the deterministic counterpart of a 2-head machine model capable to accept the linear context-free languages. It is known that

2detLIN is incomparable with the class of deterministic linear languages accepted by deterministic one-turn pushdown automata [18].

Chapter 3

A NEW SENSING $5' \to 3'$ WATSON-CRICK AUTOMATA CONCEPT

3.1 Hierarchy by sensing $5' \to 3'$ WK automata

Theorem 3.1.1. The following classes of languages coincide:

- the class of linear context-free languages defined by linear context-free grammars,
- the language class accepted by sensing $5' \to 3'$ WK finite automata,
- the class of languages accepted by **S** sensing $5' \to 3'$ WK automata,
- the class of languages accepted by **1** sensing $5' \to 3'$ WK automata.

Proof. For DNA computing reasons (and for simplicity) we work with λ-free languages. The proof is constructive, first we show that the first class is included in the last one. Let $G = (N, T, S, P)$ be a linear context-free grammar having productions only in the forms $A \to aB, A \to Ba, A \to a$ with $A, B \in N$, $a \in T$. Then the **1** sensing $5' \to 3'$ WK automaton $M = (T, id, N \cup \{q_f\}, S, \{q_f\}, \delta)$ is defined with $B \in \delta(A, u, v)$ if $A \to uBv \in P$ and $q_f \in \delta(A, u, \lambda)$ if $A \to u \in P$ $(u, v \in T \cup \{\lambda\})$. Clearly, each (terminated) derivation in G coincides to a(n accepting) computation of M, and vice versa. Thus the first class is included in the last one.

The inclusions between the fourth, third and second classes are obvious by definition. To close the circle, we need to show that the second class is in the first one. Let the sensing $5' \to 3'$ WK automaton $M = (V, id, Q, q_0, F, \delta)$ be given. Let us construct the linear context-free grammar $G = (Q, V, q_0, P)$ with productions: $p \to uqv$ if $q \in \delta(q, u, v)$ and $p \to uv \in P$ if $q \in \delta(q, u, v)$ and $q \in F$ $(p, q \in Q,\ u, v \in V^*)$. Again, the (accepting) computations of M are in a bijective correspondence to the (terminated) derivations in G. Thus, the proof is finished. □

Based on the previous theorem we may assume that the considered sensing $5' \to 3'$ WK automata have no λ-movements, i.e., at least one of the heads is moving in each transition.

Lemma 3.1.2. Let M be an **F1** sensing $5' \to 3'$ WK automaton and let the word $w \in V^+$ that is in $L(M)$. Let $|w| = k$, then for each l, where $0 \leq l \leq k$, there is at least one word $w_l \in L(M)$ such that $|w_l| = l$.

Proof. According to the definition of **F1** sensing $5' \to 3'$ WK automaton, w can be accepted in k steps such that in each step, the automaton can read exactly one letter. Moreover, each state is final, therefore by considering the first l steps of the k steps, the word $w_l = w'_l w''_l$ is accepted by M, where w'_l is read by the left head and w''_l is read by the right head during these l steps, respectively. □

Remark 3.1.3. Since, by definition, every **N1** sensing $5' \to 3'$ WK automaton is **F1** WK automaton at the same time, Lemma 3.1.2 applies for all **N1** sensing $5' \to 3'$ WK automata also.

Theorem 3.1.4. The class of languages that can be accepted by **N1** sensing $5' \to 3'$ WK automata is properly included in the language class accepted by **NS** sensing $5' \to 3'$ WK automata.

Proof. Obviously, these automata have exactly one state. In **NS** machines, the reading head may read some letters in a transition, while the input should be read letter by letter by **N1** machines. The language $L = \{a^{3n}b^{2m} \mid n,m \geq 0\}$ proves the proper inclusion. In this language w_s is bb and in an **NS** automaton it can be accepted by any of the following transitions: (bb,λ), (λ,bb). Although by Lemma 3.1.2, w_s cannot be the shortest nonempty accepted word in a language accepted by an **N1** sensing $5' \to 3'$ WK automaton. Figure 3.1 shows that language L can be accepted by an **NS** sensing $5' \to 3'$ WK automaton. Therefore, the proper inclusion stated in the theorem is proven. □

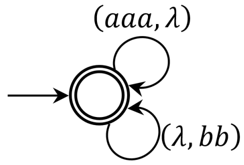


Figure 3.1: A sensing $5' \to 3'$ WK automaton of type **NS** accepting the language $\{a^{3n}b^{2m} \mid n,m \geq 0\}$.

Theorem 3.1.5. The class of languages that can be accepted by **NS** sensing $5' \to 3'$ WK automata is properly included in the language class accepted by **N** sensing $5' \to 3'$ WK automata.

Proof. The language $L = \{a^{(2n+m)}b^{(2m+n)} \mid n,m \geq 0\}$ proves the proper inclusion. Suppose that there is an **NS** sensing $5' \to 3'$ WK automaton that accepts L. The **NS** sensing $5' \to 3'$ WK automaton has exactly one state and one of the heads can move at a time. The w_s of L is aab (or abb). It can be accepted by one of the following loop transitions: (aab,λ), (λ,aab), (abb,λ) or (λ,abb) by an **NS** sensing $5' \to 3'$ WK automaton. Each of the mentioned transitions can lead to accept different language from the language $\{a^{(2n+m)}b^{(2m+n)} \mid n,m \geq 0\}$. For instance, using several times the transition (aab, λ), the language $\{(aab)^n \mid n \geq 0\}$ is accepted which is not a subset of the language L. Therefore, the language $\{a^{(2n+m)}b^{(2m+n)} \mid n,m \geq 0\}$ cannot be accepted by **NS** sensing $5' \rightarrow 3'$ WK automata. Figure 3.2 shows that this language can be accepted by an **N** sensing $5' \rightarrow 3'$ WK automaton. Hence, the theorem holds. □

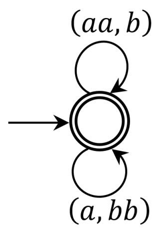


Figure 3.2: An **N** sensing $5' \rightarrow 3'$ WK automaton of type **N** accepting the language $\{a^{2n+m}b^{2m+n} \mid n,m \geq 0\}$.

The next three theorems highlight the difference between the new model and the model with sensing parameter.

Theorem 3.1.6. The class of languages that can be accepted by **F1** sensing $5' \rightarrow 3'$ WK automata is properly included in the language class of **FS** sensing $5' \rightarrow 3'$ WK automata.

Proof. Obviously, all states of these automata are final and **F1** sensing $5' \rightarrow 3'$ WK automata should read the input letter by letter, while **FS** sensing $5' \rightarrow 3'$ WK automata may read some letters in a transition. To show proper inclusion, consider the language $L = \{(aa)^n(bb)^m \mid m \leq n \leq m+1, m \geq 0\}$. The word w_s can be aa and by Lemma 3.1.2, w_s cannot be the shortest nonempty accepted word for an **F1** sensing $5' \rightarrow 3'$ WK automaton. However, L can be accepted by an **FS** sensing $5' \rightarrow 3'$ WK automaton as it is shown in Figure 3.3. The theorem is proven. □

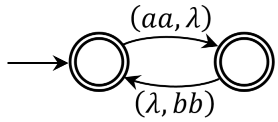


Figure 3.3: A sensing $5' \to 3'$ WK automaton of type **FS** accepting the language $\{(aa)^n(bb)^m \mid m \leq n \leq m+1, m \geq 0\}$.

Theorem 3.1.7. The language class accepted by **FS** sensing $5' \to 3'$ WK automata is properly included in the language class of **F** sensing $5' \to 3'$ WK automata.

Proof. The language $L = \{a^{2n+q}c^{4m}b^{2q+n} \mid n,q \geq 0, m \in \{0,1\}\}$ proves the proper inclusion. Let us assume, contrary that L is accepted by an **FS** sensing $5' \to 3'$ WK automaton. Let the radius of this automaton be r. Let $w = a^{2n+q}b^{2q+n} \in L$ with $n,q \geq r$ such that $|w| = 3n+3q > r$. Then the word w cannot be accepted by using only one of the transitions (from the initial state q_0), i.e., $\delta(q_0, a^{2n+q}b^{2q+n}, \lambda)$ or $\delta(q_0, \lambda, a^{2n+q}b^{2q+n})$ is not possible. Therefore, by considering the position of the heads after using any of the transitions from the initial state q_0 in **FS** sensing $5' \to 3'$ WK automaton (all states are final and one of the heads can move), it is clear that either a prefix or a suffix of w with length at most r is accepted by the automaton. But neither a word from a^+, nor from b^+ is in L. This fact contradicts to our assumption, hence L cannot be accepted by any **FS** sensing $5' \to 3'$ WK automata. However, it can be accepted by **F** $5' \to 3'$ WK automata, since the two heads can move at the same time and they can read both blocks of a's and b's simultaneously. In Figure 3.4, an all-final $5' \to 3'$ WK automaton can be seen which accepts L. $\square$

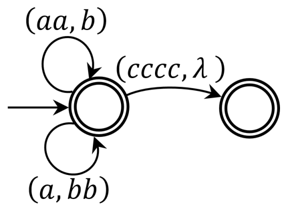


Figure 3.4: A sensing $5' \to 3'$ WK automaton of type **F** accepting the language $\{a^{2n+q}c^{4m}b^{2q+n} \mid n, q \geq 0, m \in \{0,1\}\}$.

The following result also shows that the new model differs from the one that is using the sensing parameter in its transitions.

Theorem 3.1.8. The language class accepted by **F** sensing $5' \to 3'$ WK automata is properly included in the language class of sensing $5' \to 3'$ WK automata.

Proof. The language $L = \{a^n cb^n c \mid n \geq 1\}$ can be accepted by a sensing $5' \to 3'$ WK automaton (without restrictions) (see Figure 3.5). Now we show that there is no **F** sensing $5' \to 3'$ WK automaton which accepts L. Assume the contrary that the language L is accepted by an **F** sensing $5' \to 3'$ WK automaton. Let the radius of the automaton be r. Let $w = a^m cb^m c \in L$ with $m \geq r$. Thus the word w cannot be accepted by applying exactly one transition from the initial state q_0. Now, suppose that there exists $q \in \delta(q_0, w_1, w_2)$ such that w can be accepted by using transition(s) from q. Since in **F** sensing $5' \to 3'$ WK automaton all states are final, then the concatenation of w_1 and w_2 is accepted, thus, it must be in L (i.e. $w_1w_2 \in L$). Therefore $w_1w_2 = a^{m'}cb^{m'}c$ where $2m' + 2 \leq r \leq m$. To expand both blocks a^+ and b^+ to continue the accepting path of w, the left head must be before/in/right after the subword $a^{m'}$, and the right head must be right before/in/right after the subword $b^{m'}$. However, this is contradicting the fact that the two heads together already read

$a^{m'}cb^{m'}c$. Hence, it is not possible to accept w by an **F** sensing $5' \to 3'$ WK automaton and the language L cannot be accepted by an **F** sensing $5' \to 3'$ WK automaton. □

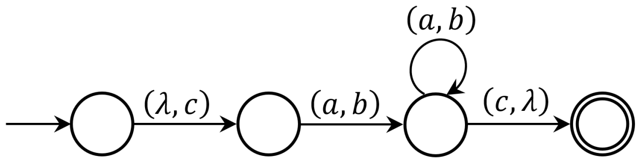


Figure 3.5: A sensing $5' \to 3'$ WK automaton accepts the language $\{a^n cb^n c \mid n \geq 1\}$.

Proposition 3.1.9. The language $L = \{a^n b^m \mid n = m \text{ or } n = m+1\}$ can be accepted by **F1** sensing $5' \to 3'$ WK automata, but cannot be accepted by **N1**, **NS** and **N** sensing $5' \to 3'$ WK automata.

Proof. As it is shown in Figure 3.6, L can be accepted by an **F1** sensing $5' \to 3'$ WK automaton. Suppose that L can be accepted by an **N** sensing $5' \to 3'$ WK automaton. The w_s of L is a, therefore at least one of the loop-transitions (a, λ) and (λ, a) is possible from the only state. Since this automaton has only one state, using any of these transitions leads to accept a^n for any $n \geq 2$ which are not in L. Thus this language cannot be accepted by an **N**, **N1**, **NS** sensing $5' \to 3'$ WK automaton. □

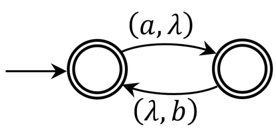


Figure 3.6: An **F1** sensing $5' \to 3'$ WK automaton accepts the language $L = \{a^n b^m \mid n = m \text{ or } n = m+1\}$.

Remark 3.1.10. The following statements follow from Proposition 3.1.9:

(a) The class of languages that can be accepted by **N1** sensing $5' \to 3'$ WK automata is properly included in the language class accepted by **F1** sensing $5' \to 3'$ WK automata.

(b) The class of languages that can be accepted by **NS** sensing $5' \to 3'$ WK automata is properly included in the language class accepted by **FS** sensing $5' \to 3'$ WK automata.

(c) The class of languages that can be accepted by **N** sensing $5' \to 3'$ WK automata is properly included in the language class accepted by **F** sensing $5' \to 3'$ WK automata.

3.1.1 Incomparability results

Theorem 3.1.11. The class of languages that can be accepted by **N** sensing $5' \to 3'$ WK automata is incomparable with the classes of languages that can be accepted by **FS** and **F1** sensing $5' \to 3'$ WK automata under set theoretic inclusion.

Proof. The language $L = \{ww^R \mid w \in \{a,b\}^*\}$ can be accepted by an **N** sensing $5' \to 3'$ WK automaton (Figure 3.7). Suppose that an **FS** sensing $5' \to 3'$ WK automaton accepts L. Let the radius of this automaton be r. Let $w_2 = w_1 w_1^R \in L$ with $w_1 = (bbbaaa)^m$ and $m > r$. The word w_2 cannot be accepted by using only one of the transitions from the initial state q_0, i.e., $\delta(q_0, w_1 w_1^R, \lambda)$ or $\delta(q_0, \lambda, w_1 w_1^R)$ is not possible (because the length of w_2). Therefore there exists either $q \in \delta(q_0, w_3 w_3^R, \lambda)$, $w_3 \in V^*$ or $q \in \delta(q_0, \lambda, w_3 w_3^R)$, $w_3 \in V^*$ such that w_2 can be accepted by using transition(s) from q. Since the word $w_3 w_3^R$ should be in the language L (i.e., it is an even palindrome) and the length of bbb and aaa patterns in w_2 is odd, the only even palindrome proper prefix (suffix) of w_2 is bb. Thus $w_3 w_3^R = bb$ must hold.

Without loss of generality, assume that there exists $q \in \delta(q_0, bb, \lambda)$ in the automaton. By continuing the process, we must have at least one of $q' \in \delta(q, w_4, \lambda)$ or $q' \in \delta(q, \lambda, w_4)$ such that $bbw_4 \in L$ and w_4 is either the prefix or the suffix of the remaining unread part of word w_2, i.e., $ba^3(b^3a^3)^{m-1}(a^3b^3)^m$, with length less than m. Clearly, w_4 cannot be a prefix, and it can be only the suffix bb. Thus, in q' the unprocessed part of the input is $ba^3(b^3a^3)^{m-1}(a^3b^3)^{m-1}a^3b$. Now the automaton must read a prefix or a suffix of this word, let us say w_5 such that $bbw_5bb \in L$, that is w_5 itself is an even palindrome, and its length is at most $r < m$. But such a word does not exist, the length of bbb and aaa patterns in the unread part is odd and their length is more than r. We have arrived to a contradiction, thus L cannot be accepted by any **FS** sensing $5' \to 3'$ WK automaton.

To prove the other direction, let us consider the language $L = \{a^n b^m \mid n = m \text{ or } n = m+1\}$. This language can be accepted by an **F1** sensing $5' \to 3'$ WK automaton as it is shown in Figure 3.6. Moreover, by Proposition 3.1.9 this language cannot be accepted by any **N** sensing $5' \to 3'$ WK automata. □

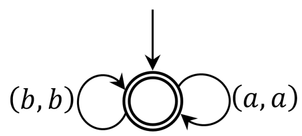


Figure 3.7: A sensing $5' \to 3'$ WK automaton of type **N** accepting the language of even palindromes $\{ww^R \mid w \in \{a,b\}^*\}$.

Theorem 3.1.12. The language class accepted by **NS** sensing $5' \to 3'$ WK automata is incomparable with the language class accepted by **F1** sensing $5' \to 3'$ WK automata.

Proof. Consider the language $L = \{a^{3n}b^{2m} \mid n, m \geq 0\}$. An **NS** sensing $5' \to 3'$ WK automaton can move one of its heads at a time. Therefore it can read three a's by the left head or two b's by the right head (see Figure 3.1). Although, according to Lemma 3.1.2, w_s is bb and it cannot be the shortest nonempty accepted word for an **F1** sensing $5' \to 3'$ WK automaton. Therefore, this language cannot be accepted by an **F1** sensing $5' \to 3'$ WK automaton.

Now let us consider the language $L = \{a^n b^m \mid n = m \text{ or } n = m+1\}$. This language can be accepted by an **F1** sensing $5' \to 3'$ WK automaton as it is shown in Figure 3.6. By Proposition 3.1.9, it is already shown that L cannot be accepted by any **N** sensing $5' \to 3'$ WK automata and obviously it cannot be accepted by any **NS** sensing $5' \to 3'$ WK automata, neither. □

Table 3.1: Some specific languages belonging to language classes accepted by various classes of WK automata. Reference to figures indicate a specific automaton that accept the given language. ✗ indicates that the language cannot be accepted by the automata type of the specific column. Trivial inclusions are also shown, e.g., in the first line **N1** in, e.g., column **F** means that every **N1** automaton is, in fact, also an **F** automaton.

Language	**N1**	**NS**	**N**	**F1**	**FS**	**F**	**WK**
$\{a^n b^m \mid n,m \geq 0\}$	Fig. 2.6	**N1**	**N1**	**N1**	**N1**	**N1**	**N1**
$\{a^{3n} b^{2m} \mid n,m \geq 0\}$	✗	Fig. 3.1	**NS**	✗	**NS**	**NS**	**NS**
$\{ww^R \mid w \in \{a,b\}^*\}$	✗	✗	Fig. 3.7	✗	✗	**N**	**N**
$\{a^n b^m \mid n = m \text{ or } n = m+1\}$	✗	✗	✗	Fig. 3.6	**F1**	**F1**	**F1**
$\{(aa)^n (bb)^m \mid m \leq n \leq m+1, m \geq 0\}$	✗	✗	✗	✗	Fig. 3.3	**FS**	**FS**
$\{a^{2n+q} c^{4m} b^{2q+n} \mid n,q \geq 0, m \in \{0,1\}\}$	✗	✗	✗	✗	✗	Fig. 3.4	**F**
$\{a^n c b^n c \mid n \geq 1\}$	✗	✗	✗	✗	✗	✗	Fig. 3.5

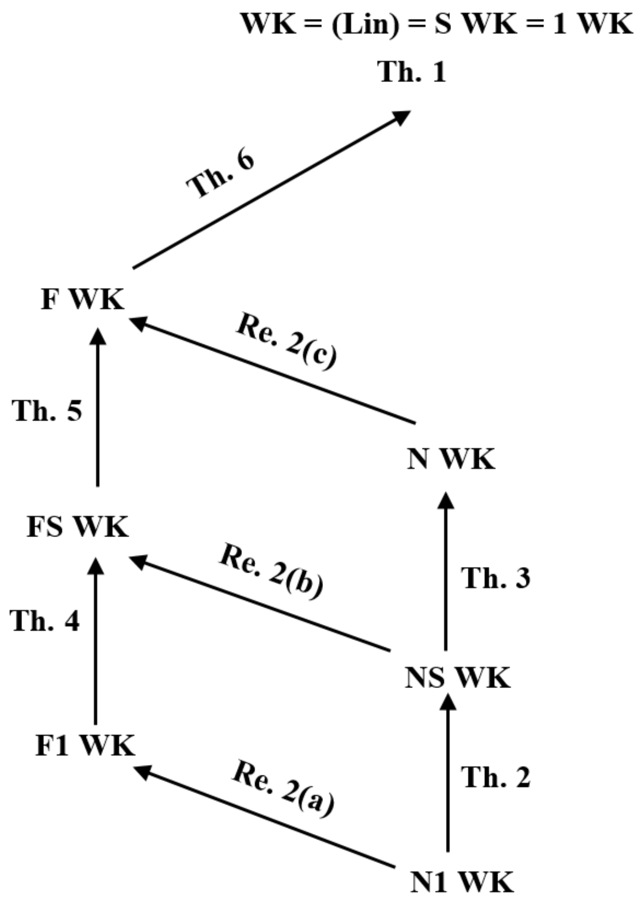


Figure 3.8: Hierarchy of sensing $5' \to 3'$ WK finite automata languages in a Hasse diagram (the language classes accepted by various types of sensing $5' \to 3'$ WK finite automata, the types of the automata are displayed in the figure with the abbreviations: **N**: stateless, **F**: all-final, **S**: simple, **1**: 1-limited; Lin stands for the class of linear context-free languages). Labels on the arrows indicate where the proof of the proper containment was presented (Th stands for Theorems, Re stands for Remark). The language classes for which the containment is not shown are incomparable.

Chapter 4

ON DETERMINISTIC SENSING $5' \to 3'$ WATSON-CRICK FINITE AUTOMATA

4.1 Characterizing 2detLIN with the new model

In this section, we show that some of the restrictions are not real restrictions (regarding the class of accepted languages), but more like normal forms. We also prove that our deterministic WK automata accepts exactly the class 2detLIN defined by the deterministic counterpart of the model working with sensing parameter.

We start the results with a general result on deterministic sensing $5' \to 3'$ WK automata. Allowing long strings to read with both heads may confuse the users to immediately see whether an automaton, in fact, is deterministic. Therefore, we start with a characterisation of the deterministic WK automata.

Proposition 4.1.1. The $5' \to 3'$ WK automaton $(V, id, Q, q_0, F, \delta)$ is not deterministic $5' \to 3'$ WK automaton if and only if there exist $q, q_1, q_2 \in Q$ and $w_L, w_R, u_1^L, u_2^L, u_1^R, u_2^R \in V^*$ such that $q_1 \in \delta(q, w_L u_1^L, u_1^R w_R)$ and $q_2 \in \delta(q, w_L u_2^L, u_2^R w_R)$ where either

- for each $i \in \{L, R\}$ there is a $j_i \in \{1, 2\}$ such that $u_1^i u_2^i = u_{j_i}^i$, moreover, $u_1^L u_2^L u_1^R u_2^R \neq \lambda$,

  or

- $u_1^L u_1^R u_2^L u_2^R = \lambda$ and $q_1 \neq q_2$.

Proof. Let us assume that $\exists q, q_1, q_2 \in Q$ such that $q_1 \in \delta(q, w_L u_1^L, u_1^R w_R)$ and $q_2 \in \delta(q, w_L u_2^L, u_2^R w_R)$ where either $|u_1^i| + |u_2^i| = \max\{|u_1^i|, |u_2^i|\}$ when $\exists u_j^i \neq \lambda$ or $q_1 \neq q_2$ when $u_1^L u_1^R u_2^L u_2^R = \lambda$. Also, let $u_L = u_1^L u_2^L$ and $u_R = u_1^R u_2^R$. By using configuration $(q, w_L u_L u_R w_R)$, it is clear that by reading two possible transitions $\delta(q, w_L u_1^L, u_1^R w_R)$ and $\delta(q, w_L u_2^L, u_2^R w_R)$, two different configurations $(q_1, u_2^L u_2^R)$ and $(q_2, u_1^L u_1^R)$ are reached, respectively. Therefore, the automaton is not deterministic $5' \rightarrow 3'$ WK.

Now, let us assume that the automaton is not deterministic $5' \rightarrow 3'$ WK automaton. Thus, $\exists q, q_1, q_2 \in Q$, $w \in V^*$ such that $(q, w) \Rightarrow (q_1, w_1)$ and $(q, w) \Rightarrow (q_2, w_2)$ where $w_1, w_2 \in V^*$. Let $w = w_{L_1} w_1 w_{R_1}, w = w_{L_2} w_2 w_{R_2}$ where $w_{L_i}, w_{R_i} \in V^*, i = 1, 2$. Moreover, let $w_L = \begin{cases} w_{L_1} & \text{if } |w_{L_1}| \leq |w_{L_2}| \\ w_{L_2} & \text{if } |w_{L_2}| < |w_{L_1}| \end{cases}$ and $w_R = \begin{cases} w_{R_1} & \text{if } |w_{R_1}| \leq |w_{R_2}| \\ w_{R_2} & \text{if } |w_{R_2}| < |w_{R_1}| \end{cases}$. Thus, $\exists u_1^L, u_1^R, u_2^L, u_2^R \in V^*$ such that $w_{L_i} = w_L u_i^L$ and $w_{R_i} = u_i^R w_R, i = 1, 2$. Therefore, there exist $q_1, q_2 \in Q$ such that $q_1 \in \delta(q, w_L u_1^L, u_1^R w_R)$ and $q_2 \in \delta(q, w_L u_2^L, u_2^R w_R)$ where either $|u_1^i| + |u_2^i| = \max\{|u_1^i|, |u_2^i|\}$ when $\exists u_j^i \neq \lambda$ or $q_1 \neq q_2$ when $w_1 = w_2$ (i.e. $u_1^L u_1^R u_2^L u_2^R = \lambda$). $\square$

We note here, that in a not deterministic $5' \rightarrow 3'$ WK automaton, to have two different transitions at a configuration does not necessarily imply to have two different successor configurations. Both can map a given configuration to a unique successor. For example, both transitions $q' \in \delta(q, ab, b)$ and $q' \in \delta(q, a, bb)$ map the configuration (q, abb) to (q', λ). However, in all these cases (assuming an alphabet with at least two letters) there are also configurations such that the given transitions map them to different configurations. In the mentioned example, consider, e.g., the configuration $(q, ababb)$.

The first case of the condition of the proposition allows the cases when exactly one of the strings $u_1^L, u_2^L, u_1^R, u_2^R$ is not empty, and cases, when exactly two of them are nonempty, especially, when $u_1^L \neq \lambda$, $u_2^R \neq \lambda$ and when $u_2^L \neq \lambda$, $u_1^R \neq \lambda$. In the second case of the condition all four of these words are empty. It never happens that exactly three of these words are nonempty. Assuming that both $u_1^L \neq \lambda$ and $u_2^L \neq \lambda$, there are two cases. If one of them (let us say u_1^L) is a prefix of the other (i.e. $u_2^L = u_1^L u_m$ where u_m maybe empty), then by choosing $w_L' = w_L u_1^L$ implies the existence of $u_1'^L = \lambda$ and $u_2'^L = u_m$. In the other case, when none of the words u_1^L and u_2^L is a prefix of the other, then there is no configuration where both of the transitions are allowed. A similar argument works if both $u_1^R \neq \lambda$ and $u_2^R \neq \lambda$.

As we have already seen in Proposition 4.1.1 that deterministic WK-automata may have some not obvious transitions. Consider, e.g., transitions reading pairs of strings $(aa, babab)$ and $(aaac, bbab)$ in a state. It is easy to see that to divide each of those transitions to two transitions allowing only to read the same strings head by head one after the other, and then, letter by letter, we receive an automaton that is not deterministic any more. Thus, considering the simple and 1-limited WK-automata, we cannot use the technique which can easily be used in the nondeterministic case to show their equivalence to the unrestricted case regarding the accepted language class. We need to work out a more careful technique.

Theorem 4.1.2. The accepted language classes of deterministic **S** and **1** sensing $5' \to 3'$ WK finite automata are equal to the language class that can be accepted by deterministic sensing $5' \to 3'$ WK finite automata (without restrictions).

Proof. The proof is constructive. Let $A = (V, id, Q, q_0, F, \delta)$ be a deterministic sensing $5' \to 3'$ WK automaton. Let us start with a simple modification of A if it has a transition with (λ, λ). Having a transition $|\delta(q, \lambda, \lambda)| > 0$ for a state q is possible in a deterministic sensing $5' \to 3'$ WK automaton if and only if there is no other transition from state q, but the only one, let say, $p = \delta(q, \lambda, \lambda)$. In this case the transitions of states p should be copied also to q, instead its original only one transition. In this way all (λ, λ) transitions can be eliminated. Further, we assume that A does not have a state with that type of transition.

Now, we construct a deterministic **1** sensing $5' \to 3'$ WK automaton $B = (V, id, Q \times V^{\leq r} \times V^{\leq r}, (q_0, \lambda, \lambda), F \times \{\lambda\} \times \{\lambda\}, \delta')$ to accept $L(A)$ where r is the radius of A.

We will do the construction by obtaining a finite sequence of automata of the form $B_i = (V, id, Q \times V^{\leq r} \times V^{\leq r}, (q_0, \lambda, \lambda), F \times \{\lambda\} \times \{\lambda\}, \delta_i')$, such that in the end of the sequence we obtain the 1-limited automaton B.

We start the process with B_0 defined as follows. In each automaton B_i the state (q, λ, λ) corresponds to $q \in Q$ in A. Let us copy, first, all transitions of A to B_0 by adding $(p, \lambda, \lambda) = \delta_0'((q, \lambda, \lambda), w_1, w_2)$ if and only if $\delta(q, w_1, w_2) = p$. Clearly, B_0 is equivalent to A, they accept the same language.

Now, we simulate the original transitions by shortening them step by step.

We have a loop that should be executed for each $q \in Q$, one after the other. Let us consider a state $q \in Q$ that has not been considered yet and the automaton B_i with the highest index that is obtained so far.

Since the automaton A is deterministic, it is impossible to have both transitions (w_1, λ) and (λ, w_2) with $w_1, w_2 \in V^+$ from state q.

If each transition from q has a nonempty string read by the left head, then let the transitions δ'_{i+1} in B_{i+1} from (q, λ, λ) be defined as follows: for each $a \in V$,

- let $\delta'_{i+1}((q, \lambda, \lambda), a, \lambda) = (p, \lambda, \lambda)$ if there is state p such that $\delta(q, a, \lambda) = p$ (actually, we copy that transition from δ'_i), otherwise
- let $\delta'_{i+1}((q, \lambda, \lambda), a, \lambda) = (q, a, \lambda)$ if a is the first letter of at least one of the words that can be read by the left head in a transition from q.

Moreover, we also add transitions from the states (q, a, λ) (for each appropriate $a \in V$):

- let $\delta'_{i+1}((q, a, \lambda), w_1, w_2) = (p, \lambda, \lambda)$ if $p = \delta(q, aw_1, w_2)$.

If there is a transition from q such that the left head reads the empty word, then (symmetrically), for each $a \in V$,

- let $\delta'_{i+1}((q, \lambda, \lambda), \lambda, a) = (p, \lambda, \lambda)$ if there is state p such that $\delta(q, \lambda, a) = p$, otherwise
- let $\delta'_{i+1}((q, \lambda, \lambda), \lambda, a) = (q, \lambda, a)$ if a is the last letter of at least one of the words that can be read by the right head in a transition from q.

Moreover, if the latter case occurs, we also add transitions from the states (q, λ, a) $(a \in V)$:

- let $\delta'_{i+1}((q, \lambda, a), w_1, w_2) = (p, \lambda, \lambda)$ if $p = \delta(q, w_1, w_2 a)$.

In this step, only the transitions from states of the form (q,u,v) (with $u,v \in V \cup \{\lambda\}$) are changed, all other transitions will be copied form δ'_i to δ'_{i+1}. In this way, the automaton B_{i+1} is equivalent to B_i, accepting exactly the same language. Moreover, based on Proposition 4.1.1, it can be seen, that we have preserved the deterministic property. However, the lengths of the read words from state (q,λ,λ) have been reduced to have 1-letter transitions and then, the rest read in a subsequent transition (from state (q,a,λ) or (q,λ,a) for $a \in V$).

Further, we iterate this process. Let i be the index of the last automaton created. Till there is a transition from a state (q,u,v) $(u,v \in V^*)$ in which more than one letter is being read (or both heads read), then the process should be done for state (q,u,v) as well:

Since the automaton B_i is deterministic, it is impossible to have both transitions (w_1,λ) and (λ,w_2) with $w_1,w_2 \in V^+$ from state (q,u,v).

If each transition from (q,u,v) has a nonempty string read by the left head, then let the transitions δ'_{i+1} in B_{i+1} from (q,u,v) be defined as follows: for each $a \in V$,

- let $\delta'_{i+1}((q,u,v),a,\lambda) = (p,\lambda,\lambda)$ if there is state p such that $\delta(q,ua,v) = p$ (actually, we copy that transition from δ'_i), otherwise
- let $\delta'_{i+1}((q,u,v),a,\lambda) = (q,ua,v)$ if a is the first letter of at least one of the words that can be read by the left head in a transition from (q,u,v) in B_i.

Moreover, we also add transitions from the states (q,ua,v) (for each appropriate $a \in V$):

- let $\delta'_{i+1}((q,ua,v),w_1,w_2) = (p,\lambda,\lambda)$ if $p = \delta(q,uaw_1,w_2v)$.

If there is a transition from (q,u,v) such that the left head reads the empty word, then for each $a \in V$,

- let $\delta'_{i+1}((q,u,v),\lambda,a) = (p,\lambda,\lambda)$ if there is state p such that $\delta(q,u,av) = p$, otherwise
- let $\delta'_{i+1}((q,u,v),\lambda,a) = (q,u,av)$ if a is the last letter of at least one of the words that can be read by the right head in a transition from (q,u,v) in B_i.

Moreover, if the latter case occurs, we also add transitions from the states (q,u,av) ($a \in V$):

- let $\delta'_{i+1}((q,u,av),w_1,w_2) = (p,\lambda,\lambda)$ if $p = \delta(q,uw_1,w_2av)$.

Furthermore, all transitions from other states than (q,u,v) will be copied form δ'_i to δ'_{i+1}.

Finally, a deterministic automaton B_j is constructed such that in each transition exactly one letter is being read by either head while the other head reads nothing. Let, then $B = B_j$ which is also equivalent to A.

Based on the construction, we have shown that the language classes accepted by deterministic sensing $5' \to 3'$ WK automaton is included in the language classes accepted by deterministic **1** sensing $5' \to 3'$ WK automata. Since this latter is a special case of the former one, this can happen only if these two classes are the same. Since, in fact all deterministic **1** sensing $5' \to 3'$ WK automata are deterministic **S** sensing $5' \to 3'$ WK automata, the theorem is completely proven. □

Remark 4.1.3. Notice that in deterministic **S**, and so **1** sensing $5' \to 3'$ WK automata, in each state at most one of the heads is allowed to read. Consequently, the states can be partitioned to two subsets depending on whether the first (left) head is allowed to move or not.

Although the deterministic sensing $5' \to 3'$ WK automata with sensing parameter (the old model) is deterministic also, the possible transitions depend on the distance of the heads. It is an important difference between the models that Remark 4.1.3 does not hold for deterministic **S**, and so **1** sensing $5' \to 3'$ WK automata with sensing parameter. In any state, the transitions should be uniquely defined only for every fixed sensing parameter. Thus, it may happen that a transition reading an a with the left head is allowed with sensing parameter $+\infty$, but the right head may read an a when the sensing distance is 1 such that the automaton still deterministic. Thus, it is not evident at all if our weaker model (without the additional tool, the parameter) is able to accept the same language class.

However, we can establish the following important result.

Theorem 4.1.4. The language class accepted by deterministic sensing $5' \to 3'$ WK automata without sensing parameter equals to the class of languages that can be accepted by deterministic sensing $5' \to 3'$ WK automata with sensing parameter.

Proof. By Theorem 4 in [18] and our Theorem 4.1.2, it is enough to show that the language class accepted by deterministic **1** sensing $5' \to 3'$ WK automaton without sensing parameter equals to the class of languages that can be accepted by a deterministic **1** sensing $5' \to 3'$ WK automaton with sensing parameter. It is obvious that in these latter automata the sensing parameter set D could include only values ∞ and 1. The proof is constructive in both directions.

Let us consider, first, the direction to show that the language class accepted by deterministic **1** sensing $5' \to 3'$ WK automata without sensing parameter is included in the language class accepted by deterministic **1** sensing $5' \to 3'$ WK automaton with sensing parameter.

Let $A' = (V, id, Q, q_0, F, \delta')$ be a sensing $5' \to 3'$ WK automaton without sensing parameter. Let the sensing $5' \to 3'$ WK automaton with sensing parameter $A = (V, id, Q, q_0, F, \delta_s)$ be defined as follows. For each transition $q' \in \delta'(q, a, \lambda)$ where $a \in V$, let $q' \in \delta_s\left(q, \begin{pmatrix} a \\ \lambda \end{pmatrix}, +\infty\right)$ be the transition in sensing $5' \to 3'$ WK automaton with sensing parameter when $q' \notin F$. Otherwise, if $q' \in F$ then both $q' \in \delta_s\left(q, \begin{pmatrix} a \\ \lambda \end{pmatrix}, 1\right)$ and $q' \in \delta_s\left(q, \begin{pmatrix} a \\ \lambda \end{pmatrix}, +\infty\right)$. Similarly this can be done for transition $q' \in \delta'(q, \lambda, a)$. It is clear that the automaton A accepts exactly $L(A')$ having essentially the same accepting computation.

Now, considering the other direction, we show that the language class accepted by deterministic **1** sensing $5' \to 3'$ WK automata with sensing parameter is included in the language class accepted by deterministic **1** sensing $5' \to 3'$ WK automata without sensing parameter. Notice that at transitions with sensing parameter 1 we can reach the accepting state by using either head to read the last letter of the input, these steps, in fact are equivalent.

Let $A = (V, id, Q, q_0, F, \delta_s)$ be a sensing $5' \to 3'$ WK automaton with sensing parameter. Now, let $A' = (V, id, Q', q'_0, F', \delta')$ be the sensing $5' \to 3'$ WK automaton without sensing parameter where $Q' = Q \cup Q \times F \cup \{q_f\}$ represents the finite set of states where $q_f \notin Q$ and $F' = Q \times F \cup \{q_f\}$. If $q_0 \in F$, then let $q'_0 = (q_0, q_0)$, otherwise let $q'_0 = q_0$. Further, the transition function δ' is defined in the following way. For each state of A we are considering the transitions with sensing parameter $+\infty$ to put them in the partitions. If in a state q transitions with the left (first) head are allowed with this parameter then we use these states for left head transitions. (More precisely, the first partition of states includes the ones satisfying the following: there is a transition with the left head with parameter $+\infty$ or there is no transition at all with parameter $+\infty$. This condition is equivalent to the following one: there is no transition with the right head with this parameter in A from state q.) Formally, for each $q \in Q$ and $a \in V$ do the following:

- If $q_1 \in \delta_s(q, \begin{pmatrix} a \\ \lambda \end{pmatrix}, +\infty)$, $q_2 \in \delta_s(q, \begin{pmatrix} \lambda \\ a \end{pmatrix}, 1)$ where $q_2 \in F$ then let $(q_1, q_2) \in \delta'(q, a, \lambda)$ (notice that $(q_1, q_2) \in F'$).
- If $q_1 \in \delta_s(q, \begin{pmatrix} \lambda \\ a \end{pmatrix}, +\infty)$, $q_2 \in \delta_s(q, \begin{pmatrix} \lambda \\ a \end{pmatrix}, 1)$ where $q_2 \in F$ then let $(q_1, q_2) \in \delta'(q, \lambda, a)$.
- If $q_1 \in \delta_s(q, \begin{pmatrix} a \\ \lambda \end{pmatrix}, +\infty)$, $q_2 \in \delta_s(q, \begin{pmatrix} a \\ \lambda \end{pmatrix}, 1)$ where $q_2 \in F$ then let $(q_1, q_2) \in \delta'(q, a, \lambda)$.
- If $q_1 \in \delta_s(q, \begin{pmatrix} \lambda \\ a \end{pmatrix}, +\infty)$, $q_2 \in \delta_s(q, \begin{pmatrix} a \\ \lambda \end{pmatrix}, 1)$ where $q_2 \in F$ then let $(q_1, q_2) \in \delta'(q, \lambda, a)$.

Further, continue as follows:

- If $q_1 \in \delta_s(q, \begin{pmatrix} a \\ \lambda \end{pmatrix}, +\infty)$, but there is no transition with either head reading an a at state q with parameter 1 then let $q_1 \in \delta'(q, a, \lambda)$ (notice that $q_1 \notin F'$).

- If $q_1 \in \delta_s(q, \begin{pmatrix} \lambda \\ a \end{pmatrix}, +\infty)$, but there is no transition with either head reading an a at state q with parameter 1 then let $q_1 \in \delta'(q, \lambda, a)$.
- If $q_1 \in \delta_s(q, \begin{pmatrix} a \\ \lambda \end{pmatrix}, 1)$ with $q_1 \in F$, and there is no transition with either head reading an a at state q with parameter $+\infty$, but there is a letter $b \in V$ such that $|\delta_s(q, \begin{pmatrix} b \\ \lambda \end{pmatrix}, +\infty)| = 1$ then let $q_f \in \delta'(q, a, \lambda)$.
- If $q_1 \in \delta_s(q, \begin{pmatrix} \lambda \\ a \end{pmatrix}, 1)$ with $q_1 \in F$, and there is no transition with either head reading an a at state q with parameter $+\infty$, but there is a letter $b \in V$ such that $|\delta_s(q, \begin{pmatrix} b \\ \lambda \end{pmatrix}, +\infty)| = 1$ then let $q_f \in \delta'(q, a, \lambda)$.
- If $q_1 \in \delta_s(q, \begin{pmatrix} a \\ \lambda \end{pmatrix}, 1)$ with $q_1 \in F$, and there is no transition with either head reading an a at state q with parameter $+\infty$, but there is a letter $b \in V$ such that $|\delta_s(q, \begin{pmatrix} \lambda \\ b \end{pmatrix}, +\infty)| = 1$ then let $q_f \in \delta'(q, \lambda, a)$.
- If $q_1 \in \delta_s(q, \begin{pmatrix} \lambda \\ a \end{pmatrix}, 1)$ with $q_1 \in F$, and there is no transition with either head reading an a at state q with parameter $+\infty$, but there is a letter $b \in V$ such that $|\delta_s(q, \begin{pmatrix} \lambda \\ b \end{pmatrix}, +\infty)| = 1$ then let $q_f \in \delta'(q, \lambda, a)$.
- If $q_1 \in \delta_s(q, \begin{pmatrix} a \\ \lambda \end{pmatrix}, 1)$ with $q_1 \in F$, and there is no transition with either head reading any letter at state q with parameter $+\infty$, then let $q_f \in \delta'(q, a, \lambda)$.
- If $q_1 \in \delta_s(q, \begin{pmatrix} \lambda \\ a \end{pmatrix}, 1)$ with $q_1 \in F$, and there is no transition with either head reading any letter at state q with parameter $+\infty$, then let $q_f \in \delta'(q, a, \lambda)$.

The extension of the transition mapping to the pairs of states is based on the first element of the pair: for each $q \in Q, a \in V$, consider each state $(q,f) \in Q \times F$ with the following transitions:

- If $q_1 \in \delta_s(q, \begin{pmatrix} a \\ \lambda \end{pmatrix}, +\infty)$, $q_2 \in \delta_s(q, \begin{pmatrix} \lambda \\ a \end{pmatrix}, 1)$ where $q_2 \in F$ then let $(q_1, q_2) \in \delta'((q,f), a, \lambda)$.
- If $q_1 \in \delta_s(q, \begin{pmatrix} \lambda \\ a \end{pmatrix}, +\infty)$, $q_2 \in \delta_s(q, \begin{pmatrix} \lambda \\ a \end{pmatrix}, 1)$ where $q_2 \in F$ then let $(q_1, q_2) \in \delta'((q,f), \lambda, a)$.
- If $q_1 \in \delta_s(q, \begin{pmatrix} a \\ \lambda \end{pmatrix}, +\infty)$, $q_2 \in \delta_s(q, \begin{pmatrix} a \\ \lambda \end{pmatrix}, 1)$ where $q_2 \in F$ then let $(q_1, q_2) \in \delta'((q,f), a, \lambda)$.
- If $q_1 \in \delta_s(q, \begin{pmatrix} \lambda \\ a \end{pmatrix}, +\infty)$, $q_2 \in \delta_s(q, \begin{pmatrix} a \\ \lambda \end{pmatrix}, 1)$ where $q_2 \in F$ then let $(q_1, q_2) \in \delta'((q,f), \lambda, a)$.

Further, consider the next steps.

- If $q_1 \in \delta_s(q, \begin{pmatrix} a \\ \lambda \end{pmatrix}, +\infty)$, but there is no transition with either head reading an a at state q with parameter 1 then let $q_1 \in \delta'((q,f), a, \lambda)$.
- If $q_1 \in \delta_s(q, \begin{pmatrix} \lambda \\ a \end{pmatrix}, +\infty)$, but there is no transition with either head reading an a at state q with parameter 1 then let $q_1 \in \delta'((q,f), \lambda, a)$.
- If $q_1 \in \delta_s(q, \begin{pmatrix} a \\ \lambda \end{pmatrix}, 1)$ with $q_1 \in F$, and there is no transition with either head reading an a at state q with parameter $+\infty$, but there is a letter $b \in V$ such that $|\delta_s(q, \begin{pmatrix} b \\ \lambda \end{pmatrix}, +\infty)| = 1$ then let $q_f \in \delta'((q,f), a, \lambda)$.
- If $q_1 \in \delta_s(q, \begin{pmatrix} \lambda \\ a \end{pmatrix}, 1)$ with $q_1 \in F$, and there is no transition with either head

reading an a at state q with parameter $+\infty$, but there is a letter $b \in V$ such that $|\delta_s(q, \begin{pmatrix} b \\ \lambda \end{pmatrix}, +\infty)| = 1$ then let $q_f \in \delta'((q,f),a,\lambda)$.

- If $q_1 \in \delta_s(q, \begin{pmatrix} a \\ \lambda \end{pmatrix}, 1)$ with $q_1 \in F$, and there is no transition with either head reading an a at state q with parameter $+\infty$, but there is a letter $b \in V$ such that $|\delta_s(q, \begin{pmatrix} \lambda \\ b \end{pmatrix}, +\infty)| = 1$ then let $q_f \in \delta'((q,f),\lambda,a)$.
- If $q_1 \in \delta_s(q, \begin{pmatrix} \lambda \\ a \end{pmatrix}, 1)$ with $q_1 \in F$, and there is no transition with either head reading an a at state q with parameter $+\infty$, but there is a letter $b \in V$ such that $|\delta_s(q, \begin{pmatrix} \lambda \\ b \end{pmatrix}, +\infty)| = 1$ then let $q_f \in \delta'((q,f),\lambda,a)$.
- If $q_1 \in \delta_s(q, \begin{pmatrix} a \\ \lambda \end{pmatrix}, 1)$ with $q_1 \in F$, and there is no transition with either head reading any letter at state q with parameter $+\infty$, then let $q_f \in \delta'((q,f),a,\lambda)$.
- If $q_1 \in \delta_s(q, \begin{pmatrix} \lambda \\ a \end{pmatrix}, 1)$ with $q_1 \in F$, and there is no transition with either head reading any letter at state q with parameter $+\infty$, then let $q_f \in \delta'((q,f),a,\lambda)$.

Finally, there is no transition from the state q_f.

One may easily check that the constructed automaton A' is also deterministic. Further it simulates the computation of the original automaton with the same states or by the first element of the pair of states, but the very last step of the accepting computations, where it accepts based on the second element of a pair of states. Consequently, it accept $L(A)$.

Therefore, the two models of deterministic WK automata are proven to be equivalent.

□

One can summarize the main results of this section as:

Corollary 4.1.5. The class of languages accepted by arbitrary deterministic sensing $5' \to 3'$ WK automata without sensing parameter is exactly the language class 2detLIN. Moreover, the same class of languages is accepted by the class of deterministic **1** sensing $5' \to 3'$ WK automata without sensing parameter.

We close the results of this section by showing how the new model is applicable to know more about the language class 2detLIN. Note that some closure properties of 2detLIN were already established in [18]. It was shown that this family is not closed under the regular operations, i.e., it is not closed under any of the operations union, concatenation and Kleene-closure. We complement those results by showing the closure properties under other set theoretic operations, i.e., under complementation and intersection.

Proposition 4.1.6. The language class 2detLIN is closed under the operation set theoretic complement.

Proof. Let M be a deterministic **1** sensing $5' \to 3'$ WK automaton without sensing parameter that accepts the 2detLIN language L. Should some transitions be not defined at M, that is, it may get stuck on some input reaching a configuration. Then, we should add a new state which will be the sink state. By Remark 4.1.3, for every state we may add the missing transitions (i.e., the missing letters will be read by the given head) to the sink state. Now, the automaton can fully read and process any input word and make the decision of the acceptance at that stage. Finally, by complementing the set of accepting states, we obtain a new automaton that accepts exactly those words that were not accepted by the original automaton. □

Applying the previous result, knowing that the class is not closed under union, by the De Morgan's law, we can infer also the following property.

Corollary 4.1.7. The language class 2detLIN is not closed under intersection.

4.2 Hierarchy results

In this section, our focus is to establish hierarchy results among the classes of accepted languages. We are focusing on the restricted classes deterministic **N1**, **NS**, **N**, **F1**, **FS** and **F** sensing $5' \rightarrow 3'$ WK automata.

We start from the most restricted class of WK automata to show that there are languages that are accepted by them:

Example 4.2.1. The language $L_1 = \{a^n | n \in \mathbb{N}\}$ is accepted by a deterministic **N1** sensing $5' \rightarrow 3'$ WK automaton as Fig. 4.1 shows. In fact, L_1 is accepted by any of the restricted and unrestricted classes of deterministic sensing $5' \rightarrow 3'$ WK automata, since by definition an **N1** automaton satisfies any of the constraints we have defined.

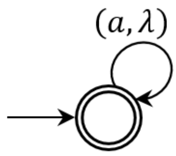


Figure 4.1: A deterministic **N1** sensing $5' \rightarrow 3'$ WK automaton accepting the language $L_1 = \{a^n | n \in \mathbb{N}\}$.

In fact, since only one of the heads can read in a deterministic **N1** sensing $5' \rightarrow 3'$ WK automaton, exactly the same class of languages are accepted by them, as the class accepted by stateless finite automata. This implies that, if we have the strong restriction on the states (no state), the strong restriction on the transitions (letter by letter, exactly 1 letter in a transition), moreover, our automaton is deterministic, we do not gain anything by adding the second head.

We continue our studies by the following observation. Any input with at most length r can be processed in one step; for every longer input the automaton must make more steps of computation before accepting them. Formally, we can state and prove a related statement about some classes of languages accepted by restricted variants.

Lemma 4.2.2. Let M be an **F1** sensing $5' \to 3'$ WK automaton and let the word $w \in V^+$be in $L(M)$. Let $|w| = n$, then for each m, where $0 \leq m \leq n$, there is at least one word $uv \in L(M)$ such that $|uv| = m$, $w = uxv$ and $u,x,v \in V^*$.

Proof. By considering the definition of **F1** sensing $5' \to 3'$ WK automaton, w can be accepted in n steps such that in each step, the automaton can read exactly one letter. Moreover, each state is final, therefore by considering the first m steps of the n steps, the word uv is accepted by M, where u is read by the left head and v is read by the right head during these m steps, respectively. □

Although this lemma is more general and works also for nondeterministic WK automata, it will also be very helpful studying the deterministic variants. Remember, that all **N1** WK automata are also **F1** WK automata.

The next series of results provides us separating languages for the various classes defined by restricted variants of WK automata.

Lemma 4.2.3. The language $L_2 = \{(ab+b)^*\}$ is accepted by

- deterministic **NS** sensing $5' \to 3'$ WK automata,
- deterministic **N** sensing $5' \to 3'$ WK automata,
- deterministic **F1** sensing $5' \to 3'$ WK automata,
- deterministic **FS** sensing $5' \to 3'$ WK automata,
- deterministic **F** sensing $5' \to 3'$ WK automata.

However, L_2 is not accepted by any

- deterministic **N1** sensing $5' \to 3'$ WK automata.

Proof. We show a deterministic **NS** sensing $5' \to 3'$ WK automaton in Fig. 4.2 (a) that accepts exactly L_2. That is clear that the automaton is deterministic, and the two transitions are corresponding to the words ab and b. This automaton is also a deterministic **N** (and **F**) sensing $5' \to 3'$ WK automaton.

L_2 is also accepted by a deterministic **F1** sensing $5' \to 3'$ WK automaton, as Fig. 4.2 (b) displays a solution. This automaton is also a deterministic **FS** and **F** sensing $5' \to 3'$ WK automaton.

Finally, we show that L_2 is not accepted by any **N1** sensing $5' \to 3'$ WK automata. Let us suppose, contrary, that an **N1** sensing $5' \to 3'$ WK automaton accepts L_2. Since L_2 contains words which contain the letter a, e.g., ab, it is possible only if the automaton has a loop transition by reading exactly one letter a. However, this loop transition leads to accept words a, aa, etc., which are not in L_2. This contradiction proves the last part of the theorem. $\square$

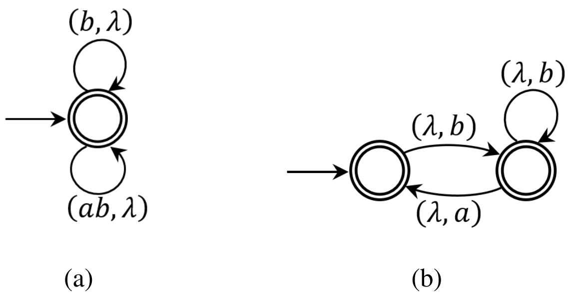


Figure 4.2: (a) A deterministic **NS** sensing $5' \to 3'$ WK automaton and (b) a deterministic **F1** sensing $5' \to 3'$ WK automaton accepting the language L_2.

Lemma 4.2.4. The language $L_3 = \{(ab)^n | n \in \mathbb{N}\}$ is accepted by

- deterministic **NS** sensing $5' \to 3'$ WK automata,
- deterministic **N** sensing $5' \to 3'$ WK automata,
- deterministic **FS** sensing $5' \to 3'$ WK automata,
- deterministic **F** sensing $5' \to 3'$ WK automata.

However, L_3 is not accepted by any

- deterministic **F1** sensing $5' \to 3'$ WK automata,
- deterministic **N1** sensing $5' \to 3'$ WK automata.

Proof. Considering the language $L_3 = \{(ab)^n | n \in \mathbb{N}\}$, the word w_s is ab and in an **NS** sensing $5' \to 3'$ WK automaton it can be accepted by one of the transitions: (λ, ab) or (ab, λ). Indeed, L_3, as shown in Figure 4.3, is accepted by an **NS** sensing $5' \to 3'$ WK automaton. In fact this automaton is also **N**, **FS** and **F** sensing $5' \to 3'$ WK automaton.

On the other hand, by Lemma 4.2.2, w_s cannot be the shortest nonempty word accepted by an **F1** or an **N1** sensing $5' \to 3'$ WK automaton. □

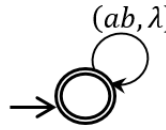


Figure 4.3: A deterministic **NS** sensing $5' \to 3'$ WK automaton accepting the language $\{(ab)^n | n \in \mathbb{N}\}$.

Lemma 4.2.5. Let us consider the linear context-free grammar $(\{S\}, \{a,b\}, S, \{S \to aSb, S \to Sa, S \to \lambda\})$. Let L_4 denote the language generated by this grammar. Then, L_4 is accepted by

- deterministic **N** sensing $5' \to 3'$ WK automata,
- deterministic **F1** sensing $5' \to 3'$ WK automata,
- deterministic **FS** sensing $5' \to 3'$ WK automata,
- deterministic **F** sensing $5' \to 3'$ WK automata.

However, L_4 is not accepted by any

- deterministic **NS** sensing $5' \to 3'$ WK automata,
- deterministic **N1** sensing $5' \to 3'$ WK automata.

Proof. L_4 is an infinite language with the following properties: Each nonempty word of L_4 starts with a letter a, this can be seen easily seeing the productions of the given grammar. There are words in L_4 that contain some b's, however, the number of b's in each word of L_4 is less than or equal to the number of a's in that word. Actually, L_4 contains each word, which starts with at least as many a's as the number of b's in it (whenever a b is derived, by the first production, an additional a is put in the beginning of the word).

Now, we present a deterministic **N** sensing $5' \to 3'$ WK automaton and a deterministic **F1** sensing $5' \to 3'$ WK automaton (that also **FS** and **F** WK automaton) accepting the language L_4, in Figure 4.4 (a) and (b), respectively. By the usual method (see, e.g., [22]) how a sensing $5' \to 3'$ WK automaton can be transformed to an equivalent grammar, it is clear that the deterministic **N** sensing $5' \to 3'$ WK automaton is equivalent to the grammar of the language L_4.

Our deterministic **F1** sensing $5' \to 3'$ WK automaton can be transformed to the linear context-free grammar $(\{S,A\},\{a,b\},S,\{S \to aA, A \to Sb, A \to Aa, S \to \lambda, A \to \lambda\})$. As one may check, this grammar generates also exactly L_4.

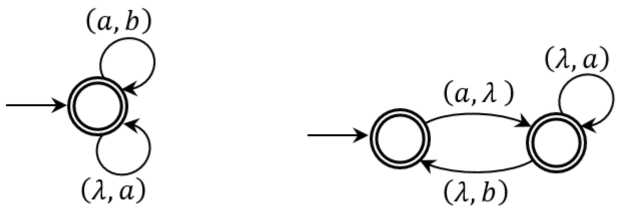
Now we turn to the last two statements of the theorem. The word w_s of L_4 is a and a deterministic **NS** sensing $5' \to 3'$ WK automaton can accept w_s by one of the transitions (a,λ) or (λ,a).

Let us consider the case when the automaton has the loop transition (a,λ). In this case, since the automaton is deterministic, all transitions must be in the form of (w_i,λ) with $w_i \in V^*$. Since there are words that contain b's, there must be at least one transition (w_j,λ) such that w_j contains the letter b. However, each w_i must be in L_4, thus each of them start with a. This, however, leads to the contradiction: a deterministic **NS** automaton cannot have both transitions (a,λ) and (w_j,λ) since a is a prefix of the word w_j.

Now let us consider the other case, i.e., if the **NS** automaton has the transition (λ,a). Since the automaton is deterministic, now, all the transitions are loops with (λ,w_i)

with some $w_i \in V^*$. Moreover, since the automaton is deterministic, each word w_i ends with a letter b. The word ab is in L_4. One option is if ab is accepted by one transition (λ, ab). However, then, by iterative use of this transition, the word $abab$ would also be accepted, but clearly $abab \notin L_4$. The other option is if ab is accepted by applying transition (λ, b) followed by the application of (λ, a). In this case, however, not only the latter one, but also (λ, b) is in the automaton. But, in this case, the word b would also be accepted, which is not in L_4.

Therefore this language is not accepted by any deterministic **NS** and so, **N1** sensing $5' \to 3'$ WK automata. $\square$



(a) (b)

Figure 4.4: The language L_4 is accepted by (a) a deterministic **N** sensing $5' \to 3'$ WK automaton and (b) a deterministic **F1** sensing $5' \to 3'$ WK automaton.

Lemma 4.2.6. Let us consider the linear context-free grammar $(\{S\}, \{a,b\}, S, \{S \to aaSb, S \to Saa, S \to \lambda\})$. Let L_5 denote the language generated by this grammar. Then, L_5 is accepted by

- deterministic **N** sensing $5' \to 3'$ WK automata,
- deterministic **FS** sensing $5' \to 3'$ WK automata,
- deterministic **F** sensing $5' \to 3'$ WK automata.

However, L_5 is not accepted by any

- deterministic **F1** sensing $5' \to 3'$ WK automata,
- deterministic **NS** sensing $5' \to 3'$ WK automata,
- deterministic **N1** sensing $5' \to 3'$ WK automata.

Proof. L_5 is obtained from L_4 by replacing each occurrence of the letter a by the word aa. Consequently, L_5 is accepted by a deterministic **N** and by an **FS** sensing $5' \to 3'$ WK automata shown in Figure 4.5.

Now, we prove that there is no deterministic **F1** sensing $5' \to 3'$ WK automata that accept L_5. The word w_s of L_5 is aa but by Lemma 4.2.2, aa cannot be the shortest nonempty accepted word for any deterministic **F1** (and **N1**) sensing $5' \to 3'$ WK automata.

To show that there is no deterministic **NS** sensing $5' \to 3'$ WK automata that accept L_5 is going by a similar argument as for the language L_4 replacing all a's in the proof by the word aa. □

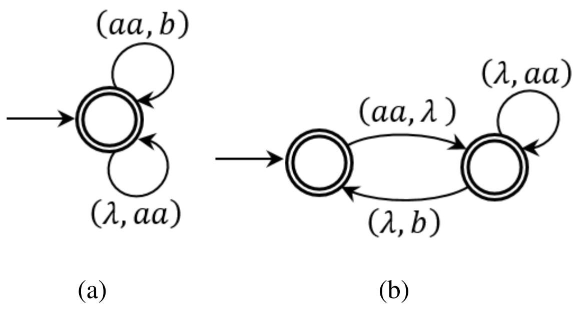


Figure 4.5: (a) A deterministic **N** sensing $5' \to 3'$ WK automaton and (b) a deterministic **FS** sensing $5' \to 3'$ WK automaton accepting the language L_5.

Lemma 4.2.7. The language $L_6 = \{a^{2n}b^{2n} | n \in \mathbb{N}\}$ is accepted by

- deterministic **N** sensing $5' \to 3'$ WK automata,

- deterministic **F** sensing $5' \to 3'$ WK automata.

However, L_6 is not accepted by any

- deterministic **FS** sensing $5' \to 3'$ WK automata,
- deterministic **F1** sensing $5' \to 3'$ WK automata,
- deterministic **NS** sensing $5' \to 3'$ WK automata,
- deterministic **N1** sensing $5' \to 3'$ WK automata.

Proof. First, Figure 4.6 shows a deterministic **N** sensing $5' \to 3'$ WK automaton that clearly accepts L_6. It is in fact also an **F** WK automaton.

Second, we prove that L_6 is not accepted by any deterministic **FS** sensing $5' \to 3'$ WK automata (which proves also the last three statements of the lemma). Contrary, let us assume that L_6 is accepted by a deterministic **FS** sensing $5' \to 3'$ WK automaton. Let the radius of this automaton be r. Let $w = a^{2m}b^{2m} \in L$ with $m > \frac{r}{4}$. Then the word w cannot be accepted by using only one of the transitions from initial state q_0, i.e., $\delta(q_0, a^{2m}b^{2m}, \lambda)$ or $\delta(q_0, \lambda, a^{2m}b^{2m})$ is not possible. Since, all states are final and every word of L has the same number of a's and b's then neither a prefix nor a suffix of w can be accepted by a transition from q_0. This fact contradicts to our assumption, hence this language cannot be accepted by any deterministic **FS** sensing $5' \to 3'$ WK automata and obviously by any of **F1**, **NS** and **N1** sensing $5' \to 3'$ WK automata. □

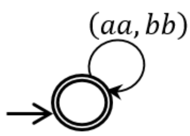


Figure 4.6: A deterministic **N** sensing $5' \to 3'$ WK automaton accepting the language $\{a^{2n}b^{2n} | n \in \mathbb{N}\}$.

Lemma 4.2.8. The language $L_7 = \{a^n b^m | n \in \mathbb{N}, m \in \{0,1\}\}$ is accepted by

- deterministic **F1** sensing $5' \to 3'$ WK automata,
- deterministic **FS** sensing $5' \to 3'$ WK automata,
- deterministic **F** sensing $5' \to 3'$ WK automata.

On the other hand, L_7 is not accepted by any

- deterministic **N** sensing $5' \to 3'$ WK automata,
- deterministic **NS** sensing $5' \to 3'$ WK automata,
- deterministic **N1** sensing $5' \to 3'$ WK automata.

Proof. We prove, first, that the language L_7 is accepted by a deterministic **F1** sensing $5' \to 3'$ WK automaton: Figure 4.7 shows the solution. In fact, this automaton is also deterministic **FS**, and so, **F** sensing $5' \to 3'$ WK automaton.

On the other hand, the transitions in a deterministic **N** sensing $5' \to 3'$ WK automaton are loop transitions. The word b is in L_7, the automaton must have a transition to accept it. But any loop transition containing b causes that the automaton can accept the words having more than one b which results words not in L. Since all **NS** and **N1** sensing $5' \to 3'$ WK automata are **N** sensing $5' \to 3'$ WK automata, the argument also applies for them. □

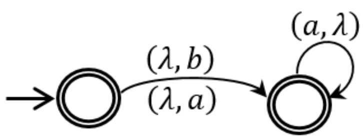


Figure 4.7: Language $L_7 = \{a^n b^m | n \in \mathbb{N}, m \in \{0,1\}\}$ is accepted by a deterministic **F1** sensing $5' \to 3'$ WK automaton.

Lemma 4.2.9. The language $L_8 = \{(ab)^n(cc)^m | n \in \mathbb{N}, m \in \{0,1\}\}$ is accepted by

- deterministic **FS** sensing $5' \to 3'$ WK automata,
- deterministic **F** sensing $5' \to 3'$ WK automata.

However, L_8 is not accepted by any

- deterministic **F1** sensing $5' \to 3'$ WK automata,
- deterministic **N** sensing $5' \to 3'$ WK automata,
- deterministic **NS** sensing $5' \to 3'$ WK automata,
- deterministic **N1** sensing $5' \to 3'$ WK automata.

Proof. On one hand, L_8 is accepted by an **FS** sensing $5' \to 3'$ WK automaton (see Figure 4.8). Obviously this is also an **F** sensing $5' \to 3'$ WK automaton.

On the other hand, first we show that no **F1** sensing $5' \to 3'$ WK automaton accepts L_8. The word w_s of this language is ab or cc which can be accepted by transitions (λ, ab), (ab, λ), (cc, λ), or (λ, cc) in an **FS** sensing $5' \to 3'$ WK automaton. However, according to Lemma 4.2.2, w_s cannot be the shortest nonempty accepted word in **F1** sensing $5' \to 3'$ WK automaton.

Finally, we show that no **N** sensing $5' \to 3'$ WK automaton accepts L_8. Since c is not in L_8, but $cc \in L_8$, the stateless automaton must have a transition to accept cc in one step: at least one of the loop transitions (λ, cc), (c, c), or (cc, λ) must be in the automaton. However, the iterated use of this transition leads to accept words, e.g., $cccc$ which are not in L_8. This contradiction shows that no **N**, and thus, no **NS** and **N1** sensing $5' \to 3'$ WK automaton accepts L_8. $\square$

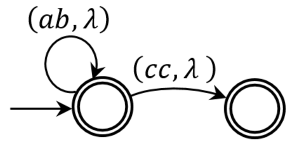


Figure 4.8: A deterministic **FS** sensing $5' \to 3'$ WK automaton accepting the language $L_8 = \{(ab)^n(cc)^m | n \in \mathbb{N}, m \in \{0,1\}\}$.

Lemma 4.2.10. The language $L_9 = \{a^{2n}(ccccc)^q b^{2n} | n \in \mathbb{N}, q \in \{0,1\}\}$ is accepted by

- deterministic **F** sensing $5' \to 3'$ WK automata.

However, L_9 is not accepted by any

- deterministic **FS** sensing $5' \to 3'$ WK automata,
- deterministic **F1** sensing $5' \to 3'$ WK automata,
- deterministic **N** sensing $5' \to 3'$ WK automata,
- deterministic **NS** sensing $5' \to 3'$ WK automata,
- deterministic **N1** sensing $5' \to 3'$ WK automata.

Proof. First we present a deterministic **F** sensing $5' \to 3'$ WK automaton that accepts L_9: see Figure 4.9.

Now, let us assume, contrary, that L_9 is accepted by a deterministic **FS** sensing $5' \to 3'$ WK automaton. Let the radius of this automaton be r. Let $w = a^{2m}b^{2m} \in L$ with $m > \frac{r}{4}$. Then the word w cannot be accepted by using only one of the transitions from initial state q_0, i.e., $\delta(q_0, a^{2m}b^{2m}, \lambda)$ or $\delta(q_0, \lambda, a^{2m}b^{2m})$ is not possible. Since, all states are final and every word of L has the same number of a's and b's then neither a prefix nor a suffix of w can be accepted by a transition from q_0. This fact contradicts to our assumption, hence L_9 is not accepted by any deterministic **FS** sensing $5' \to 3'$

WK automata. This implies also that there is no deterministic **F1**, **NS** and **N1** sensing $5' \to 3'$ WK automata that accept L_9.

Finally, we need also to prove that L_9 is not accepted by any deterministic **N** sensing $5' \to 3'$ WK automaton. Assume that L_9 is accepted by a stateless automaton. Since $ccccc$ is in L_9, but not any of the words $c, cc, ccc, cccc$ are in L_9, there must be a transition in which exactly 5 c's are read. But then, applying this loop transition iteratively, words like c^{10} are also accepted. Since they are not in L_9, we got a contradiction, which proves our statement. □

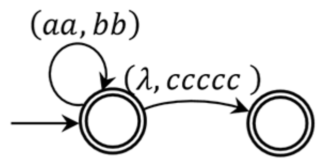


Figure 4.9: A deterministic **F** sensing $5' \to 3'$ WK automaton accepting the language $\{a^{2n}c^{5q}b^{2n} | n \in \mathbb{N}, q \in \{0,1\}\}$.

Lemma 4.2.11. The language $L_{10} = \{a^n d b^n c | n \in \mathbb{N}\}$ is accepted by

- deterministic sensing $5' \to 3'$ WK automata.

However, L_{10} is not accepted by any

- deterministic **F** sensing $5' \to 3'$ WK automata,
- deterministic **FS** sensing $5' \to 3'$ WK automata,
- deterministic **F1** sensing $5' \to 3'$ WK automata,
- deterministic **N** sensing $5' \to 3'$ WK automata,
- deterministic **NS** sensing $5' \to 3'$ WK automata,
- deterministic **N1** sensing $5' \to 3'$ WK automata.

Proof. The language $L_{10} = \{a^n db^n c | n \geq 1\}$ is accepted by a deterministic sensing $5' \to 3'$ WK automaton (without restrictions) as it is shown in Figure 4.10. In the proof of Theorem 6 in [22], we showed that the language L_{10} cannot be accepted by any nondeterministic **F** sensing $5' \to 3'$ WK automaton. Therefore, there is no deterministic **F**, **FS**, **F1**, **N**, **NS** and **N1** sensing $5' \to 3'$ WK automata that accept L_{10}. □

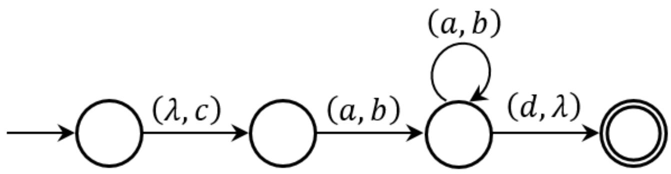


Figure 4.10: A deterministic sensing $5' \to 3'$ WK automaton accepts the language $\{a^n db^n c | n \geq 1\}$.

Finally, we recall that 2detLIN is a proper subset of the family of linear context-free languages, e.g., the language $L_{11} = \{a^n ba^n \mid n \in \mathbb{N}\} \cup \{a^n ca^{2n} \mid n \in \mathbb{N}\}$ is a linear language and it is not in 2detLIN.

Now, we are ready to state the hierarchy results for the classes accepted by the various restricted variants of WK automata.

Theorem 4.2.12. The following proper inclusions hold for the language classes accepted by the variants of deterministic sensing $5' \to 3'$ WK automata without sensing parameter:

$$\mathbf{N1} \subsetneq \mathbf{NS} \subsetneq \mathbf{N} \subsetneq \mathbf{F} \subsetneq \text{2detLIN} \subsetneq \text{LIN}$$

$$\mathbf{N1} \subsetneq \mathbf{F1} \subsetneq \mathbf{FS} \subsetneq \mathbf{F} \quad \text{and} \quad \mathbf{NS} \subsetneq \mathbf{FS}$$

(The abbreviations of the type of the automata above denote the language class defined by the given restricted class, while LIN denotes the class of linear context-free languages.)

Proof. First, we notice that all the subset relations included in the theorem are trivial by considering the constraints of the variants and by knowing that, by Corollary 4.1.5, the class 2detLIN is the class accepted by the class of arbitrary deterministic sensing $5' \to 3'$ WK automata with sensing parameter.

To show the properness of the inclusions, we provide the separating languages L_2, L_4, L_7, L_{10} and L_{11} for the first line (applying Lemmas 4.2.3, 4.2.5, 4.2.8 and 4.2.11, respectively).

For the second line, the separating languages L_2, L_3, L_6 and L_8 can be used, based on Lemmas 4.2.3, 4.2.4, 4.2.7 and 4.2.9, respectively. □

Among these hierarchy results some highlight the difference between the new model (without sensing parameter) and the old model [18].

Remark 4.2.13. Deterministic **F1** sensing $5' \to 3'$ WK automata with sensing parameter are as powerful as the deterministic sensing $5' \to 3'$ WK automata with sensing parameter without any additional restrictions. In the new model, opposite to this, we have a finer hierarchy: The language class accepted by deterministic **F1** sensing $5' \to 3'$ WK automata is a proper subset of the language class accepted by deterministic **FS** sensing $5' \to 3'$ WK automata. Further, the language class accepted by deterministic **FS** sensing $5' \to 3'$ WK automata is a proper subset of the language class accepted by deterministic **F** sensing $5' \to 3'$ WK automata. Finally, the class accepted by deterministic **F** sensing $5' \to 3'$ WK automata is a proper subset of 2detLIN.

We can complement the previous results by some incomparability results.

Theorem 4.2.14. The pairs of language classes accepted by the following pairs of classes of deterministic sensing $5' \to 3'$ WK automata are incomparable under set theoretical inclusion:

1. classes of deterministic **F1** and **NS** sensing $5' \to 3'$ WK automata,
2. classes of deterministic **F1** and **N** sensing $5' \to 3'$ WK automata,
3. classes of deterministic **FS** and **N** sensing $5' \to 3'$ WK automata.

Proof. The first two statement follows from Lemmas 4.2.4 and 4.2.8, by the properties of languages L_3 and L_7. The last statement follows from Lemmas 4.2.7 and 4.2.8, by the properties of languages L_6 and L_7. □

To complete the picture we show further relations among the language classes defined by variants of deterministic sensing $5' \to 3'$ WK automata.

Theorem 4.2.15. The intersections of the language classes accepted by the following pairs of classes of deterministic sensing $5' \to 3'$ WK automata contain languages that are not in the third class:

- $(\mathbf{F1} \cap \mathbf{NS}) \setminus \mathbf{N1} \neq \emptyset$,
- $(\mathbf{F1} \cap \mathbf{N}) \setminus \mathbf{NS} \neq \emptyset$,
- $(\mathbf{FS} \cap \mathbf{N}) \setminus \mathbf{NS} \neq \emptyset$.

(The abbreviations denotes the classes of accepted languages, respectively.)

Proof. Consider the languages L_2, L_4, L_5 from Lemma 4.2.3, 4.2.5, 4.2.6, respectively.

□

The complete picture is shown in Figure 4.11, where the abbreviations of the specific automata models represent the class of accepted languages, respectively.

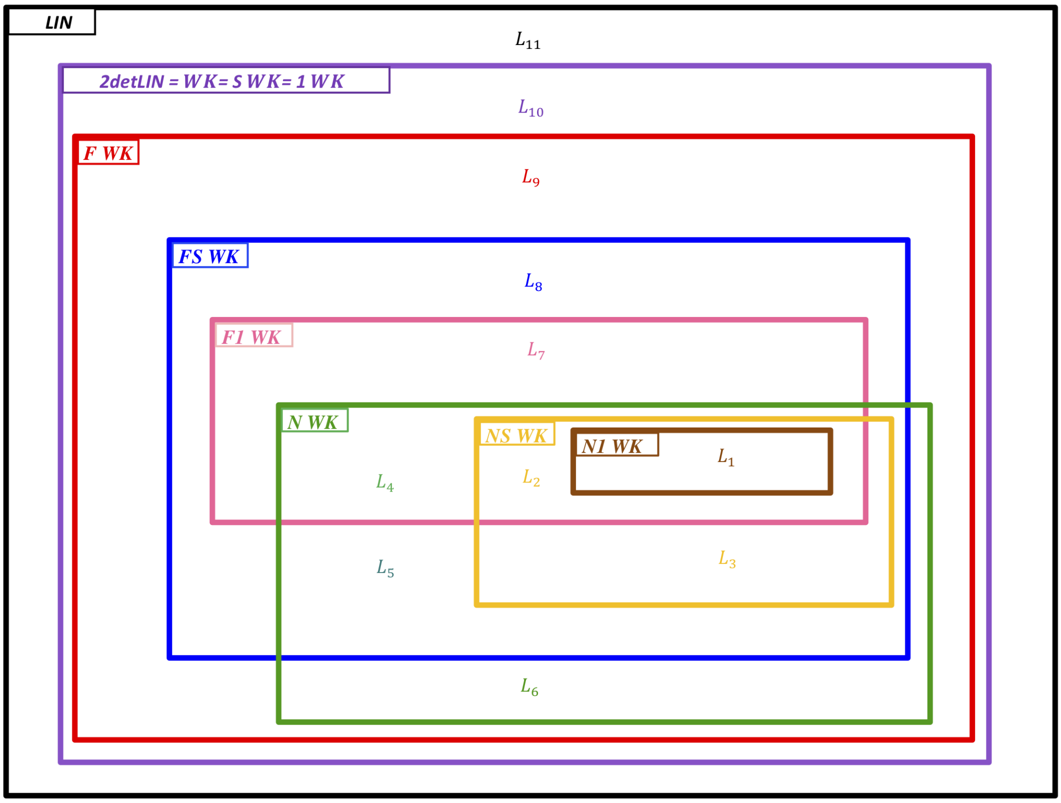


Figure 4.11: Hierarchy and incomparability results of deterministic sensing $5' \to 3'$ WK finite automata languages (where the abbreviations of various automata classes stand for the defined language classes) in a Venn-Euler diagram: 2detLIN and its special subclasses are displayed inside LIN, the class of linear context-free languages.

Chapter 5

CONCLUSION

The general nondeterministic variants (the automata without using any restrictions) of the new model without sensing parameter and the old model with sensing parameter have the same accepting power, i.e., exactly the linear context-free languages [14, 18, 22]. However, by our proofs, the new model gives a more finer hierarchy, as it is displayed in Figure 3.8. Table 3.1 gives some specific languages that separate some of the language classes. Further comparisons of related language classes and properties of the language classes defined by the new model are left to the future.

In this thesis we have shown that this is also true for their deterministic counterparts, both of them characterise the class 2detLIN. This result was not straightforward, since the sensing parameter gave more freedom in the old model allowing different set of transitions when the heads are close to the meeting point and when they are not. However, we have efficiently simulated the original automata with the new model keeping it deterministic. In this way, by our results, the class 2detLIN can be further analysed using these newer and simpler automata without the very technical sensing parameter. We have proven here that this class is closed under complementation, but not under intersection. A summary of our hierarchy results is shown in Figure 4.11. We note that the deterministic hierarchy of languages investigated in Chapter 4 is very similar to the hierarchy shown for the nondeterministic model which is described in Chapter 3, although the classes are different. Already nondeterministic **N1** automata are more powerful than their deterministic variants. We should also recall that, in the nondeterministic case, it was trivial to simulate the string reading feature of the automata by having the restriction to read exactly 1 letter in each transition. This was more technical in the deterministic case (that we have managed here in Theorem 4.1.2). On the other side, the hierarchy presented here is finer (containing 7 classes) than the hierarchy obtained by the model with sensing parameter (containing only 4 classes including 2detLIN, [18]).

REFERENCES

[1] Adleman, L. M. (1994). Molecular computation of solutions to combinatorial problems, *Science*, 226, 1021-1024.

[2] Amar, V. & Putzolu, G. R. (1964). On a Family of Linear Grammars, *Information and Control* 7/3, 283-291.

[3] Czeizler, E., Czeizler E. (2006). A Short Survey on Watson-Crick Automata, *Bulletin of the EATCS*, 88, 104-119.

[4] Fernau, F. & Sempere, J. M. (2000). Permutations and Control Sets for Learning Non-regular Language Families, *ICGI 2000, LNCS/LNAI 1891*, 75-88.

[5] Freund, R. & Păun, G. & Rozenberg, G. & Salomaa, A. (1997). Watson-Crick finite automata, *In: 3rd DIMACS Sympozium On DNA Based Computers*, Philadelphia, 305–317.

[6] Hegedüs, L. & Nagy, B. & Egecioglu, Ö (2012). Stateless multicounter $5' \to 3'$ Watson-Crick automata: the deterministic case, *Natural Computing* 11/3, 361-368.

[7] Holzer, M. & Kutrib, M. & Otto, F. (2017). Two-sided strictly locally testable languages, *NCMA 2017*, 135-150.

[8] Hopcroft, J. E. & Ullman, J. D. (1979). Introduction to Automata Theory, Languages and Computation, *Addison-Wesley Publishing Company, Reading*

MA.

[9] Kocman, R. & Nagy, B. & Krivka, Z. & Meduna, A. (2018). *A jumping* $5' \rightarrow 3'$ *Watson-Crick finite automata model*, NCMA 2018, 117-132.

[10] Kuske, D. & Weigel, P. (2004). The role of the complementarity relation in Watson-Crick automata and sticker systems, *In: Developments in Language Theory, DLT 2004, Lecture Notes in Computer Science, LNCS 3340*, Springer, Berlin, Heidelberg, 272–283.

[11] Kutrib, M. & Otto, F. (2018). Two-sided locally testable languages, *NCMA 2018*, 133-148.

[12] Leupold, P. & Nagy, B. (2010). $5' \rightarrow 3'$ Watson-Crick automata with several runs, *Fundamenta Informaticae 104*, 71-91.

[13] Nagy, B. (2007). On $5' \rightarrow 3'$ sensing Watson-Crick finite automata, *Preproceedings of DNA07*, The 13th International Meeting on DNA Computing, Memphis, Tennessee, USA, 327-336.

[14] Nagy, B. (2008). On $5' \rightarrow 3'$ sensing Watson-Crick finite automata, *In: Garzon M.H., Yan H. (eds): DNA Computing. DNA 2007: Selected revised papers, Lecture Notes in Computer Science, LNCS 4848*, Springer, Berlin, Heidelberg, 256-262.

[15] Nagy, B. (2009). On a hierarchy of $5' \rightarrow 3'$ sensing WK finite automata languages, *In: Computaility in Europe, CiE 2009: Mathematical Theory and Computational*

Practice, Abstract Booklet, Heidelberg, 266–275.

[16] Nagy, B. (2010). $5' \to 3'$ Sensing Watson-Crick Finite Automata, *In: Gabriel Fung (ed.): Sequence and Genome Analysis II - Methods and Applications , iConcept Press*, 39–56.

[17] Nagy, B. (2012). A class of 2-head finite automata for linear languages, *Triangle 8: Languages. Mathematical Approaches*, 89-99.

[18] Nagy, B. (2013). On a hierarchy of $5' \to 3'$ sensing Watson-Crick finite automata languages, *Journal of Logic and Computation 23*(4), 855–872.

[19] Nagy, B. (2015). A family of two-head pushdown automata, *NCMA 2015*, 177-191.

[20] Nagy, B. & Kovács, Z. (2019). On simple $5' \to 3'$ sensing Watson-Crick finite-state transducers, *NCMA 2019*, 155-170.

[21] Nagy, B. & Otto, F. (2019). Two-Head Finite-State Acceptors with Translucent Letters, *SOFSEM 2019, LNCS 11376*, 406-418.

[22] Nagy, B. & Parchami, Sh. & Mir-Mohammad-Sadeghi, H. (2017). A New Sensing $5' \to 3'$ Watson-Crick Automata Concept, *AFL 2017, EPTCS 252*, 195-204. doi: 10.4204/EPTCS.252.19

[23] Parchami, Sh. & Nagy B. (2018). Deterministic Sensing $5' \to 3'$ Watson-Crick Automata Without Sensing Parameter, *UCNC 2018, LNCS 10867*, 173-187.

[24] Nagy B. & Parchami, Sh. (2020). On deterministic sensing $5' \to 3'$ Watson-Crick finite automata: a full hierarchy in 2detLIN, *Acta Informatica*. doi=10.1007/s00236-019-00362-6

[25] Păun, G. (1997). Marcus Contextual Grammars, *Studies in Linguistics and Philosophy Volume 67*, Kluwer, Dordrecht.

[26] Păun, G. & Rozenberg, G. & Salomaa, A. (2002). DNA Computing: New Computing Paradigms, *Springer-Verlag*.

[27] Rozenberg, G. & Salomaa, A. (1997). editors: Handbook of Formal Languages, *Springer*.

[28] Sempere, J. M. (2018). On the application of Watson-Crick finite automata for the resolution of bioinformatic problems, *NCMA 2018*, 29-30.

[29] Sempere, J. M. & García, P. (1994). A Characterization of Even Linear Languages and its Application to the Learning Problem, *ICGI 1994, LNCS/LNAI 862*, 38-44.

[30] Sempere, J. M. & García, P. (2002). Learning Locally Testable Even Linear Languages from Positive Data, *ICGI 2002*, 225-236.