2024 Joint 13th International Conference on Soft Computing and Intelligent Systems and 25th International Symposium on Advanced Intelligent Systems (SCIS&ISIS) | 979-8-3503-7333-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/SCISISIS61014.2024.10760167

Pumping Lemmata for Fuzzy Languages Accepted by Jumping and Right One-Way Jumping Fuzzy Finite Automata

Pavel Martinek
Department of Mathematics
Tomas Bata University in Zlín
Nám. T. G. Masaryka 5555, 760 01 Zlín, Czech Republic
e-mail: pmartinek@utb.cz

***Abstract*—Jumping finite automata represent an interesting kind of automata working discontinuously over their input. They can read any input symbol and jump without restrictions on any other (not yet processed) symbol. Right one-way jumping finite automata start their reading from the first symbol of the input word, their 'jumps' can be performed only in the left-to-right direction and after reaching the end of the input word, they can resume from its left hand side. This modified specification (in deterministic and nondeterministic variants) leads to automata with different computational power. The paper deals with fuzzy versions of these automata and with pumping lemmata for fuzzy languages accepted by them.**

***Index Terms*—Fuzzy jumping finite automata, fuzzy right one-way jumping finite automata, fuzzy nondeterministic right one-way jumping finite automata, Pumping lemma.**

I. Introduction

Jumping finite automata were introduced by Meduna and Zemek in [1]. These automata can skip some symbols during processing their input and return to their processing later. They led to enlarged research of computational devices which work over their input discontinuously (cf., e.g., [2], [3], [4] or [5]).

A fuzzification of jumping finite automata was described in [6] where the structure of truth values over the unit interval $[0; 1]$ (with the operations of minimum and maximum) is used. We proceed with fuzzification of further variants of jumping finite automata, specifically deterministic and nondeterministic right one-way jumping finite automata. Some properties of languages accepted by crisp (i.e., non-fuzzy) finite automata can be transformed into corresponding properties of fuzzy languages straightforwardly, while others cannot. Pumping lemmata represent the latter case. For instance, when considering Pumping lemmata formulated for fuzzy regular languages (using various truth value structures), they generally replace the condition of full membership of a word in a crisp language with a positive membership condition for the word in a fuzzy language (cf. [7]). Specifically, the condition $uxy, ux^2y \in L$ is replaced by the conditions of positive membership values of uxy and ux^2y in L without further demands on mutual relationship of the values. In the case of languages accepted by deterministic finite automata, it is easy to prove that the membership values are equal (cf. [8] or [7]). However, if fuzzy languages are accepted by certain nondeterministic fuzzy automata with greater computational power than deterministic fuzzy automata of the same kind, the situation can differ. This holds true for right one-way jumping fuzzy finite automata. In the paper, we will explore Pumping lemmata for the corresponding fuzzy languages.

The presented paper is organized as follows. Section II presents the fundamental concepts of jumping finite automata, deterministic and nondeterministic right one-way jumping finite automata. In Section III, fuzzy versions of these automata are described. Finally, in Section IV, Pumping lemma is adapted to three families of fuzzy languages accepted by the aforementioned fuzzy automata.

II. Preliminaries

Since we assume certain familiarity of the reader with the basic notions from Formal languages theory (cf. [9], [10]), we remind only few of them:

If Σ is a finite nonempty set of symbols we call it an *alphabet*. For any set Σ, we denote by Σ^* the set of all finite strings over Σ (including the empty string ε) provided with the binary operation of concatenation. If $a_i = a \in \Sigma$ for $i \in \{1, \ldots, m\}$, we write a^m instead of $a_1 \cdots a_m$. Further, we use notation $a^0 = \varepsilon$. For a string w over Σ, we denote the number of occurrences of a symbol a in w by $|w|_a$. The number of occurrences of all symbols in w is called the *length* of w and is denoted by $|w|$. If $w = a_1 \cdots a_n$ is a string over Σ (with $|w| = n$) and $\varphi : \{1, \ldots, n\} \to \{1, \ldots, n\}$ is a bijection, then the string $P(w) = a_{\varphi(1)} \cdots a_{\varphi(n)}$ is said to be a *permutation* of w.

The next definition is based on [1].

Definition 1: A *jumping finite automaton* (JFA) is an ordered quintuple $A = (Q, \Sigma, \delta, q_0, F)$ where Q is a nonempty finite set of states, Σ is an input alphabet, $\Sigma \cap Q = \emptyset$, $\delta \subseteq Q \times \Sigma \times Q$ is a transition relation, q_0 is the initial state, and $F \subseteq Q$ is a set of final states.

A *configuration* is a string $uqv \in \Sigma^* Q \Sigma^*$. Here $q \in Q$ is the current state of the automaton A, uv is the not yet processed content of the input string, and the starting symbol of v is the symbol which should be processed in the next computational step.

The binary *jumping relation* (representing a computational step) is a relation $\curvearrowright \subseteq \Sigma^* Q \Sigma^* \times \Sigma^* Q \Sigma^*$ defined by $(uqav, u'rv') \in \curvearrowright$ iff $(q, a, r) \in \delta$ and $uv = u'v'$. As is usual, we denote the reflexive and transitive closure of $\curvearrowright$ by $\curvearrowright^*$.

The *language* $L(A)$ accepted by the jumping finite automaton A is defined by

$$L(A) = \{uv \mid u, v \in \Sigma^*, (uq_0v, q_f) \in \curvearrowright^* \text{ for some } q_f \in F\}.$$

Example 1 ([11]): Let $A = (Q, \Sigma, \delta, q_0, F)$ be a JFA with $Q = \{q_0, q_1, q_2\}$, $\Sigma = \{a, b\}$, $\delta = \{(q_0, a, q_1), (q_1, b, q_2), (q_2, a, q_1)\}$, $F = \{q_2\}$.

The automaton is represented graphically with help of the state diagram in Fig. 1 (i.e. by a labelled directed graph whose nodes represent states of the automaton, the initial state is indicated by the arrow pointing at it from nowhere, each final state is depicted by double circle, and each arc in the graph coincides with a transition — if the arc goes from state q to state r and $(q, a, r) \in \delta$ then the arc is labelled by a.)

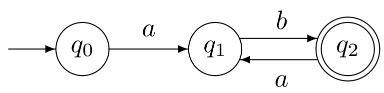

Fig. 1.

For an input string $baab$, we have for example: $(bq_0aab, baq_1b) \in \curvearrowright$, $(baq_1b, bq_2a) \in \curvearrowright$, $(bq_2a, q_1b) \in \curvearrowright$, and $(q_1b, q_2) \in \curvearrowright$. Therefore, $(bq_0aab, q_2) \in \curvearrowright^*$. It is easy to see that $L(A) = \{w \in \Sigma^* \mid |w|_a = |w|_b > 0\}$ which represents the well-known example of non-regular language. □

Jumping finite automaton is defined as a nondeterministic device. Its nondeterministic behaviour follows both from its transition relation and from ability to jump (nondeterministically) to any position in the processed input. The freedom in 'choosing' next position is suppressed in right one-way jumping finite automata which were defined with 'deterministic and nondeterministic transition relations' in [2] and [12], successively.

Definition 2: A *nondeterministic right one-way jumping finite automaton* (NROWJFA) is an ordered quintuple $A = (Q, \Sigma, \delta, q_0, F)$ where its components are defined as in a JFA. A way of how an NROWJFA processes its input is described below. An NROWJFA $A = (Q, \Sigma, \delta, q_0, F)$ is said to be a *right one-way jumping finite automaton*[1] (ROWJFA) if $(q, a, r) \in \delta$ and $(q, a, s) \in \delta$ imply $r = s$ for all $q, r, s \in Q$ and $a \in \Sigma$.

[1]Despite the fact that a right one-way jumping finite automaton represents a deterministic automaton, in agreement with [2] and [13], we will not use adjective 'deterministic' in its name.

A *configuration* of A is a string $qw \in Q\Sigma^*$. Here $q \in Q$ is the current state of the automaton, w is the not yet processed content of the input string, and the starting symbol of w is the symbol which should be either processed or skipped in the next computational step.

The binary *right one-way jumping relation* (representing a computational step) is a relation $\circlearrowright \subseteq Q\Sigma^* \times Q\Sigma^*$ defined by $(quav, rvu) \in \circlearrowright$ iff $(q, a, r) \in \delta$ and $(q, b, r') \notin \delta$ for any symbol b of the string u and any $r' \in Q$. This means that the automaton reads (and deletes) the first possible symbol of the string uav according to its transition relation. The skipped substring u is considered to be moved behind the substring v. As is usual, we denote the reflexive and transitive closure of $\circlearrowright$ by $\circlearrowright^*$.

The *language* $L(A)$ accepted by the (nondeterministic) right one-way jumping finite automaton A is defined by

$$L(A) = \{w \mid w \in \Sigma^*, (q_0w, q_f) \in \circlearrowright^* \text{ for some } q_f \in F\}.$$

Example 2 ([11]): Let $A = (Q, \Sigma, \delta, q_0, F)$ be a ROWJFA with state diagram from Fig. 1.

For an input string $aabb$, we have: $(q_0aabb, q_1abb) \in \circlearrowright$, $(q_1abb, q_2ba) \in \circlearrowright$, $(q_2ba, q_1b) \in \circlearrowright$, and $(q_1b, q_2) \in \circlearrowright$. Hence, $(q_0aabb, q_2) \in \circlearrowright^*$. It is easy to see that $L(A) = \{aw \in a\{a,b\}^* \mid |aw|_a = |w|_b\}$. □

Theorem 1 ([12], [13]):

(i) The family of languages accepted by ROWJFA is the proper subfamily of the family of languages accepted by NROWJFA.

(ii) The family of languages accepted by JFA is the proper subfamily of the family of languages accepted by NROWJFA.

(iii) The families of languages accepted by ROWJFA and JFA are incomparable.

III. Jumping Fuzzy Finite Automata

Similarly to other papers (cf. [14], [6]), we consider fuzzy sets with truth values in the unit interval $[0, 1]$, i.e. a *fuzzy set* in a universe set X is any mapping $M : X \to [0, 1]$, $M(x)$ being interpreted as the truth degree of the fact that "x belongs to M" and being called the *membership value*. A *fuzzy relation* R between sets X and Y is defined as a mapping $R : X \times Y \to [0, 1]$. Analogously, a fuzzy ternary relation R' is defined as a mapping $R' : X \times Y \times Z \to [0, 1]$. For brevity, in the paper we denote the minimum operation over real numbers by the symbol $\wedge$.

Definition 3 ([6]): A *jumping fuzzy finite automaton* (JFFA) is an ordered quintuple $A = (Q, \Sigma, \delta, q_0, F)$ where Q is a nonempty finite set of states, Σ is an input alphabet, $\Sigma \cap Q = \emptyset$, $\delta : Q \times \Sigma \times Q \to [0, 1]$ is a fuzzy transition relation, $q_0 \in Q$ is the initial state, and $F : Q \to [0, 1]$ is a fuzzy set in Q.

The *jumping fuzzy relation* is a mapping $\curvearrowright : \Sigma^* Q \Sigma^* \times \Sigma^* Q \Sigma^* \to [0, 1]$ which is for all $u, v, u', v' \in \Sigma^*$, $a \in \Sigma$, and $q, r \in Q$, defined by

$$\curvearrowright (uqav, u'rv') \overset{\text{def}}{=} \begin{cases} \delta(q,a,r) & \text{if } uv = u'v', \\ 0 & \text{otherwise.} \end{cases}$$

We extend the fuzzy relation $\curvearrowright$ to a fuzzy relation $\curvearrowright^*$: $\Sigma^*Q\Sigma^* \times \Sigma^*Q\Sigma^* \rightarrow [0,1]$ in the following way. Let $u, v, u', v' \in \Sigma^*$ and $q, s \in Q$. Then

$$\curvearrowright^0 (uqv, u'sv') \overset{\text{def}}{=} \begin{cases} 1 & \text{if } uqv = u'sv', \\ 0 & \text{otherwise,} \end{cases}$$

$$\curvearrowright^n (uqv, u'sv') \overset{\text{def}}{=} \max_{\substack{r \in Q \\ u'', v'' \in \Sigma^*}} \{\curvearrowright (uqv, u''rv'') \wedge \curvearrowright^{n-1}(u''rv'', u'sv')\} \text{ for all } n \geq 1,$$

$$\curvearrowright^* (uqv, u'sv') \overset{\text{def}}{=} \max_{i \geq 0} \{\curvearrowright^i (uqv, u'sv')\}.$$

The *fuzzy language* $L(A)$ accepted by the JFFA A is defined by

$$L(A)(w) = \max_{\substack{q \in Q \\ u, v \in \Sigma^* \\ uv = w}} \{\curvearrowright^*(uq_0v, q) \ \wedge \ F(q)\} \ \text{ for all } \ w \in \Sigma^*$$

and is called a JFFA-*language*.

Example 3: Consider JFFA $A = (Q, \Sigma, \delta, q_0, F)$ which is represented graphically with help of the state diagram in Fig. 2.

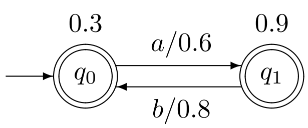

Fig. 2.

It means that

$Q = \{q_0, q_1\}$,

$\Sigma = \{a, b\}$,

$\delta(q_0, a, q_1) = 0.6$,

$\delta(q_1, b, q_0) = 0.8$,

$\delta(q_i, x, q_j) = 0$ otherwise,

$F(q_0) = 0.3, \ F(q_1) = 0.9$.

Then, for example,

$\curvearrowright^* (bq_0a, q_0) = \curvearrowright^2 (bq_0a, q_0) = \max\{\curvearrowright (bq_0a, q_1b) \wedge \curvearrowright (q_1b, q_0), \curvearrowright (bq_0a, bq_1) \wedge \curvearrowright (bq_1, q_0)\} = \max\{0.6 \wedge 0.8, 0.6 \wedge 0\} = 0.6$,

$\curvearrowright^* (baq_0a, q_1) = \curvearrowright^3 (baq_0a, q_1) = \max\{\curvearrowright (baq_0a, q_1ba) \wedge \ \curvearrowright^2 (q_1ba, q_1)\} = \max\{\curvearrowright (baq_0a, q_1ba) \wedge \ \curvearrowright (q_1ba, q_0a) \wedge \ \curvearrowright (q_0a, q_1)\} = \max\{0.6 \wedge 0.8 \wedge 0.6\} = 0.6$.

Obviously,

$L(A)(ba) = \max\{\curvearrowright^* (bq_0a, q_0) \wedge F(q_0), \curvearrowright^* (bq_0a, q_1) \wedge F(q_1)\} = \max\{0.6 \wedge 0.3, 0 \wedge 0.9\} = 0.3$

and

$L(A)(baa) = \max\{\curvearrowright^* (baq_0a, q_1) \wedge F(q_1)\} = 0.6 \wedge 0.9 = 0.6$.

It is easy to see that for all $w \in \{a, b\}^*$,

$$L(A)(w) = \begin{cases} 0.6 & \text{if } |w|_a = |w|_b + 1, \\ 0.3 & \text{if } |w|_a = |w|_b, \\ 0 & \text{otherwise.} \end{cases}$$

□

The next definition represents a modification of the previous definition.

Definition 4: A *nondeterministic right one-way jumping fuzzy finite automaton* (NROWJFFA) is an ordered quintuple $A = (Q, \Sigma, \delta, q_0, F)$ where Q is a nonempty finite set of states, Σ is an input alphabet, $\Sigma \cap Q = \emptyset$, $\delta : Q \times \Sigma \times Q \rightarrow [0,1]$ is a fuzzy transition relation, $q_0 \in Q$ is the initial state, and $F : Q \rightarrow [0,1]$ is a fuzzy set in Q.

The *right one-way jumping fuzzy relation* is a mapping $\circlearrowright$: $Q\Sigma^* \times Q\Sigma^* \rightarrow [0,1]$ which is for all $u, v \in \Sigma^*$, $a \in \Sigma$, and $q, r \in Q$, defined by

$$\circlearrowright (quav, rvu) \overset{\text{def}}{=} \begin{cases} \delta(q,a,r) & \text{if } \delta(q,a,r) > 0 \text{ and} \\ & \delta(q,b,r') = 0 \text{ for any} \\ & \text{symbol } b \text{ of the string } u \\ & \text{and any } r' \in Q, \\ 0 & \text{otherwise.} \end{cases}$$

We extend the fuzzy relation $\circlearrowright$ to a fuzzy relation $\circlearrowright^*$: $Q\Sigma^* \times Q\Sigma^* \rightarrow [0,1]$ in the following way. Let $u, v \in \Sigma^*$ and $q, s \in Q$. Then

$$\circlearrowright^0 (qu, sv) \overset{\text{def}}{=} \begin{cases} 1 & \text{if } qu = sv, \\ 0 & \text{otherwise,} \end{cases}$$

$$\circlearrowright^n (qu, sv) \overset{\text{def}}{=} \max_{\substack{r \in Q \\ u' \in \Sigma^*}} \{\circlearrowright (qu, ru') \wedge \circlearrowright^{n-1}(ru', sv)\} \ \forall n \geq 1,$$

$$\circlearrowright^* (qu, sv) \overset{\text{def}}{=} \max_{i \geq 0} \{\circlearrowright^i (qu, sv)\}.$$

The *fuzzy language* $L(A)$ accepted by the NROWJFFA A is defined by

$$L(A)(w) = \max_{q \in Q} \{\circlearrowright^*(q_0w, q) \ \wedge \ F(q)\} \quad \text{for all} \ \ w \in \Sigma^*$$

and is called an NROWJFFA-*language*.

A ROWJFFA-*language* is defined as the *fuzzy language* $L(A)$ accepted by the ROWJFFA $A = (Q, \Sigma, \delta, q_0, F)$ which differs from NROWJFFA by fulfilling an additional condition: $\delta(q, a, r) > 0$ for some $q, r \in Q, a \in \Sigma$ implies $\delta(q, a, s) = 0$ for all $s \in Q, s \neq r$. The *fuzzy language* $L(A)$ accepted by a ROWJFFA A is called a ROWJFFA-*language*.

The following theorem represents a straightforward generalization of Theorem 1.

Theorem 2:

(i) The family of ROWJFFA-languages is the proper subfamily of the family of NROWJFFA-languages.

(ii) The family of JFFA-languages is the proper subfamily of the family of NROWJFFA-languages.

(iii) The families of ROWJFFA and JFFA-languages are incomparable.

IV. Pumping Lemmata

Pumping lemma for regular languages expresses well-known necessary condition for a language to be regular (cf. [15], [16]):

Theorem 3: If L is a regular language, then there is a number p (the pumping length) where, if w is any string in L of length at least p, then w can be divided into three substrings, $w = xyz$, satisfying the following conditions:

(i) $|y| > 0$,
(ii) $|xy| \leq p$,
(iii) for each $i \geq 0$, $xy^iz \in L$.

There are many papers adapting pumping lemma to fuzzy regular languages with various structures of truth values (see, for example, [8], [17] or [7]). Following Corollary 11 from [2], we adapt the lemma to JFFA, ROWJFFA and NROWJFFA-languages. Since the family of ROWJFFA-languages is the proper subfamily of the family of NROWJFFA-languages by Theorem 2, we obtain two versions of pumping lemma (differing in mutual relationship of membership values of $L(w)$ and $L(xy^iz)$) in the following Theorems 4 and 5 (cf. also Theorems 3 and 4 of [14]).

Theorem 4: Let Σ be an alphabet and $L : \Sigma^* \to [0, 1]$. If L is either a JFFA-language or an NROWJFFA-language, then there is a number p (the pumping length) where, if w is any string of length at least p such that $L(w) > 0$, then there is a permutation $P(w)$ of w which can be divided into three substrings, $P(w) = xyz$, satisfying the following conditions:

(i) $|y| > 0$,
(ii) $|xy| \leq p$,
(iii) for each $i \geq 0$, $L(w) \leq L(xy^iz)$.

Proof: Since L is a JFFA-language (NROWJFFA-language), there is a JFFA (NROWJFFA) $A = (Q, \Sigma, \delta, q_0, F)$ which accepts L. Denote the number of states in Q by p. Let $w = a_1 \cdots a_n$ be a string with $a_1, \ldots, a_n \in \Sigma$, $n \geq p$ and $L(A)(w) > 0$. Further, let $\triangle \in \{\curvearrowright, \circlearrowright\}$. (Clearly, $\triangle = \curvearrowright$ and $\triangle = \circlearrowright$ will cover the cases of JFFA and NROWJFFA, respectively.)

By Definition 3 (or 4), there are $b_1, \ldots, b_n \in \{a_1, \ldots, a_n\}$, $b_1 \cdots b_n = P(w)$, $q_1, \ldots, q_n \in Q$ such that $L(A)(w) = \triangle^n(q_0, w, q_n) \wedge F(q_n) = \delta(q_0, b_1, q_1) \wedge \delta(q_1, b_2, q_2) \wedge \ldots \wedge \delta(q_{n-1}, b_n, q_n) \wedge F(q_n)$.

Since $n \geq p$, by the Pigeonhole principle, a state q' must be repeated in the sequence $q_0, \ldots, q_n$, i.e. $q' = q_j = q_k$ for some $0 \leq j < k \leq n$. Suppose that j and k are the first such indexes, i.e. $q' \neq q_m$ for all $m \in \{0, \ldots, j-1, j+1, \ldots, k-1\}$. We put

$$x = b_1 \cdots b_j,$$
$$y = b_{j+1} \cdots b_k,$$
$$z = b_{k+1} \cdots b_n.$$

Obviously, $|y| > 0$ and $|xy| \leq p$. So, conditions (i) and (ii) are satisfied.

(A) Consider $i \geq 1$. With regard to the inequality $\triangle^*(q_0, xy^iz, q_n) \geq \triangle^*(q_k, y^{i-1}, q_k) \wedge \triangle^*(q_0, xyz, q_n)$, which follows directly from Definition 3 (or 4), we have:

$$\begin{aligned} L(A)(xy^iz) = {} & \max_{q \in Q} \{\triangle^*(q_0, xy^iz, q) \wedge F(q)\} \geq \\ & \triangle^*(q_0, xy^iz, q_n) \wedge F(q_n) \geq \\ & \triangle^*(q_k, y^{i-1}, q_k) \wedge \triangle^*(q_0, xyz, q_n) \wedge F(q_n) = \\ & \triangle^*(q_0, xyz, q_n) \wedge F(q_n) = \\ & L(A)(w). \end{aligned}$$

(B) $$\begin{aligned} L(A)(w) = {} & \delta(q_0, b_1, q_1) \wedge \delta(q_{n-1}, b_n, q_n) \wedge F(q_n) \leq \\ & \delta(q_0, b_1, q_1) \wedge \ldots \wedge \delta(q_{j-1}, b_j, q_j) \wedge \\ & \delta(q_k, b_{k+1}, q_{k+1}) \wedge \ldots \wedge \delta(q_{n-1}, b_n, q_n) \wedge \\ & F(q_n) \leq \triangle^*(q_0, xz, q_n) \wedge F(q_n) \leq \\ & \max_{q \in Q} \{\triangle^*(q_0, xz, q) \wedge F(q)\} = \\ & L(A)(xz). \end{aligned}$$

(A) and (B) imply that condition (iii) of the theorem holds true. □

Remark 1: There is a natural question whether the inequalities in Theorem 4(iii) can be replaced by equalities. The answer is negative both for JFFA and NROWJFFA-languages as follows from the next example.

Example 4: Let $A = (Q, \Sigma, \delta, q_0, F)$ be a JFFA (or NROWJFFA) with graphical representation from Fig. 3.

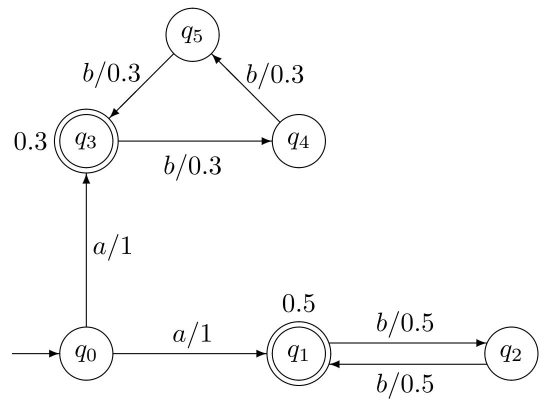

Fig. 3.

Obviously, for any $w \in \{a, b\}^*$, we have

$$L(A)(w) = \begin{cases} 0.5 & \text{if } \; w = ab^{2n} \text{ where } n \geq 0, \\ 0.3 & \text{if } \; w = ab^{3(2n-1)} \text{ where } n \geq 1, \\ 0 & \text{otherwise.} \end{cases}$$

If we follow proof of Theorem 4 then we have p = 6. Consider $w = ab^9$ (whose length is greater than p). Then,

$L(A)(w) = 0.3$. To obtain another string w' by 'pumping' a substring y of a permutation $P(w)$, clearly the substring y must be of the form b^m with $m \in \{1, \ldots, 5\}$ to fulfil the condition $|xy| = |ab^k| \le p = 6$. Hence, $P(w) = xyz = ab^m b^{9-m}$ and $xy^iz = ab^{mi}b^{9-m} = ab^{9+m(i-1)}$. There are the following cases:

- m is odd. Then, $xy^iz = ab^{2n}$ for some $n > 0$ and $L(A)(xy^iz) = 0.5$.
- $m \in \{2, 4\}$. Then, $xy^iz = ab^{9+2(i-1)}$ or $xy^iz = ab^{9+4(i-1)}$ for some $i \ge 1$. However, for example, $i = 2$ implies $L(A)(xy^iz) = L(A)(ab^{11}) = 0$ or $L(A)(xy^iz) = L(A)(ab^{13}) = 0$.

We can conclude that, w' containing y^2 implies $L(A)(w') \in \{0, 0.5\}$ and $L(A)(w) = 0.3 \ne L(A)(w')$. (Similar reasoning can be described for greater numbers p or i, as well.) Therefore, the inequalities in Theorem 4(iii) cannot be replaced by equalities in the case of JFFA-languages or NROWJFFA-languages. $\square$

Replacement of the inequalities in Theorem 4(iii) by equalities is possible only for some narrower family of fuzzy languages. By Theorem 2, the family of ROWJFFA-languages is a proper subfamily of the family of NROWJFFA-languages. This fact leads to the next theorem describing another pumping lemma. For a better understanding of its proof, we present the following example.

Example 5: Let $A = (Q, \Sigma, \delta, q_0, F)$ be a ROWJFFA with graphical representation from Fig. 4.

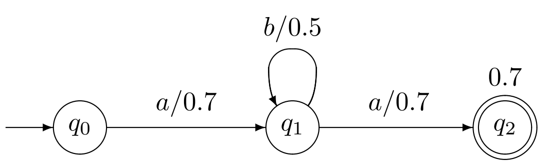

Fig. 4.

Obviously, for any $w \in \{a, b\}^*$, we have

$$L(A)(w) = \begin{cases} 0.7 & \text{if } w = aa, \\ 0.5 & \text{if } w = ab^na \text{ where } n \ge 1, \\ 0 & \text{otherwise.} \end{cases}$$

If we follow proof of Theorem 4 then we have p = 3. Consider $w = aba$ (whose length is equal to p). Then, $L(A)(w) = 0.5$. To obtain another string w' by 'pumping' a substring y of a permutation $P(w)$, clearly there must be $y = b$. However,

$$L(A)(xy^iz) = L(A)(ab^ia) = \begin{cases} 0.7 & \text{if } i = 0, \\ 0.5 & \text{if } i \ge 1. \end{cases}$$

Thus, for all $i \ge 1$, $L(A)(xy^iz) = L(A)(w) = 0.5 \ne L(A)(xz) = 0.7$. This example demonstrates that choice of the value p in proof of Theorem 4(iii) is not great enough to prove the equality $L(A)(xy^iz) = L(A)(xyz)$ for $i = 0$ at ROWJFFA-languages. However, reasonable increasing of the value p in proof of Theorem 5 will solve this obstacle. $\square$

Theorem 5: Let Σ be an alphabet and $L : \Sigma^* \to [0, 1]$. If L is a ROWJFFA-language, then there is a number p (the pumping length) where, if w is any string of length at least p such that $L(w) > 0$, then there is a permutation $P(w)$ of w which can be divided into three substrings, $P(w) = xyz$, satisfying the following conditions:

(i) $|y| > 0$,
(ii) $|xy| \le p$,
(iii) for each $i \ge 0$, $L(w) = L(xy^iz)$.

Proof: The proof can be performed as a slight modification of proof of Theorem 4. To avoid the situation described in Example 5, it suffices to set the pumping length $p = k^{k+1}$ where k denotes the number of states of ROWJFFA accepting the considered language L. (Note that there can be different computational cycles and in a substring of w of length at least k, there is present some cycle.) Clearly, every word of length at least p contains at least 'two runs of a particular cycle'. The equalities in condition (iii) follow from deterministic way of computation of every ROWJFFA, which concerns Part (A) in the end of proof of Theorem 4 (where the determinism changes all inequalities into equalities). Note that Part (A) deals with words containing at least one substring whose reading starts and finishes in the same state q_k and our choice of p ensures words having this property doubled. $\square$

V. Conclusion

In this paper, we introduced deterministic and nondeterministic right one-way jumping fuzzy finite automata and reminded jumping fuzzy finite automata. Pumping lemmata for fuzzy languages accepted by these automata are formulated and proved. Since every pumping lemma expresses a necessary condition for a language to belong to the considered language class, we obtained tools suitable for demonstrating that certain languages cannot be described by the fuzzy automata under consideration.

References

[1] A. Meduna and P. Zemek, "Jumping finite automata," Int. J. Found. Comput. Sci., vol. 23, 2012, pp. 1555–1578.

[2] H. Chigahara, S.Z. Fazekas, and A. Yamamura, "One-way jumping finite automata," Int. J. Found. Comput. Sci., vol. 27, 2016, pp. 391–405.

[3] S.Z. Fazekas, K. Hoshi, and A. Yamamura, "Enhancement of automata with jumping modes," In: Cellular Automata and Discrete Complex Systems, Proceedings of 25th IFIP WG 1.5 International Workshop on Cellular Automata and Discrete Complex Systems, AUTOMATA 2019, A. Castillo-Ramirez, P.P.B. de Oliveira, Eds., Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 11525 LNCS, 2019, pp. 62-76.

[4] K. Mahalingam, U. K. Mishra, and R. Raghavan, "Watson-Crick Jumping Finite Automata," Int. J. Found. Comput. Sci., vol. 31, no. 7, 2020, pp. 891–913.

[5] B. Nagy and F. Otto, "Linear automata with translucent letters and linear context-free trace languages," RAIRO - Theor. Inf. Appl., vol. 54, 2020, article no. 3.

[6] P. Martinek, “Jumping Fuzzy Finite Automata and Their Languages,” In: Fuzzy Systems and Data Mining IV, Proceedings of the 4th International Conference on Fuzzy Systems and Data Mining (FSDM 2018), A. J. Tallón-Ballesteros, K. Li, Eds., Frontiers in Artificial Intelligence and Applications, vol. 309, Amsterdam: IOS Press, 2018, pp. 196–201.

[7] J. R. González de Mendívil, J. R. Garitagoitia, “Fuzzy languages with infinite range accepted by fuzzy automata: Pumping lemma and determinization procedure,” Fuzzy Sets Syst., vol. 249, 2014, pp. 1–26.

[8] D. S. Malik and J. N. Mordeson, “On Fuzzy Regular Languages,” Inf. Sci., vol. 88, 1996, pp. 263–273.

[9] A. Salomaa, Formal Languages, Academic Press, New York, 1973.

[10] D. Wood, Theory of Computation, Addison-Wesley, Boston, 1987.

[11] S. Beier and M. Holzer, “Properties of right one-way jumping finite automata,” In: DCFS 2018, S. Konstantinidis, G. Pighizzini, Eds., LNCS, vol. 10952, Springer, Heidelberg, 2018, pp. 11–23.

[12] S. Beier and M. Holzer, “Nondeterministic right one-way jumping finite automata,” Inf. Comput., vol. 284, 2022, article no. 104687.

[13] S. Beier and M. Holzer, “Properties of right one-way jumping finite automata,” Theor. Comput. Sci., vol. 798, 2019, pp. 78–94.

[14] P. Martinek, “Fuzzy multiset finite automata: Determinism, languages, and pumping lemma,” In: 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2015, article no. 7381915, pp. 60–64.

[15] J. E. Hopcroft, R. Motwani, and J. D. Ullman, Introduction to Automata Theory, Languages, and Computation, 2nd ed., Upper Saddle River: Pearson Addison Wesley, 2003.

[16] M. Sipser, Introduction to the Theory of Computation, 2nd ed., Boston: Thomson Course Technology, 2006.

[17] D. W. Qiu, “Pumping lemma in automata theory based on complete residuated lattice-value logic: A note,” Fuzzy Sets Syst., vol. 157, 2006, pp. 2128–2138.