Filomat 40:1 (2026), 309–320
https://doi.org/10.2298/FIL2601309J

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

Approximate children fuzzy automata over the product structure

Zorana Jančić[a,*], José Ramón González de Mendívil[b], Ivana Micić[a], Miroslav Ćirić[a]

[a]*University of Niš, Faculty of Sciences and Mathematics, Višegradska 33, 18000 Niš, Serbia*
[b]*Departamento de Estadística, Informática y Matemáticas, Universidad Pública de Navarra, Campus de Arrosadia, Pamplona 31006, Spain*

Abstract. The general problem of determinization of fuzzy automata over the product structure is unsolvable, which necessitates the use of approximate methods. This paper introduces a new approach to the approximate determinization of fuzzy finite automata by utilizing a transition to a different structure, known as the truncated product structure. This structure is residuated and has a locally finite semiring reduct, which enables efficient computation. The proposed method constructs a so-called Children automaton and employs approximate weak simulations to achieve more effective determinization. In comparison to existing techniques, this approach significantly enhances the performance and precision of the resulting crisp-deterministic fuzzy automata.

1. Introduction

The determinization of fuzzy finite automata stands as a fundamental and thought-provoking topic in fuzzy automata theory. In the classical framework of Boolean-valued automata, any non-deterministic finite automaton can be replaced by an equivalent deterministic one, albeit with a potentially exponential increase in the number of states. However, this property does not hold for fuzzy automata, as an equivalent deterministic fuzzy finite automaton may not always exist. This complexity makes the determinization process for fuzzy automata considerably more challenging.

There are two primary approaches to the determinization of fuzzy finite automata. The first, known as crisp-determinization, involves converting a fuzzy automaton into an equivalent crisp-deterministic fuzzy automaton [1, 4, 5, 7, 8]. This form can be interpreted as a classical deterministic automaton, where the traditional final state set is replaced by a fuzzy set of final states. The second approach, called fuzzy determinization, produces fuzzy deterministic finite automata, representing a generalized form of crisp-deterministic fuzzy automata, where the initial and transition truth degrees are not restricted to absolute truth [11, 12, 15, 16, 26, 27].

2020 *Mathematics Subject Classification*. Primary 68Q45; Secondary 68Q42, 68Q10.
Keywords. fuzzy automata, determinization, simulation, approximate simulation.
Received: 14 March 2025; Accepted: 03 November 2025
Communicated by Marko Petković
Z. Jančić, I. Micić and M. Ćirić acknowledge the support of the Science Fund of the Republic of Serbia, Grant No. 7750185, Quantitative Automata Models: Fundamental Problems and Applications – QUAM. They are also supported by the Ministry of Science, Technological Development and Innovation, Republic of Serbia, Contract No. 451-03-137/2025-03/200124.
* Corresponding author: Zorana Jančić
Email addresses: zorana.jancic@pmf.edu.rs (Zorana Jančić), mendivil@unavarra.es (José Ramón González de Mendívil), ivana.micic@pmf.edu.rs (Ivana Micić), miroslav.ciric@pmf.edu.rs (Miroslav Ćirić)
ORCID iDs: https://orcid.org/0000-0002-7495-5534 (Zorana Jančić), https://orcid.org/0000-0003-3150-6437 (José Ramón González de Mendívil), https://orcid.org/0000-0002-3816-9464 (Ivana Micić), https://orcid.org/0000-0001-8625-4682 (Miroslav Ćirić)

The concept of crisp-determinization was first explored in [1, 23], with a foundational procedure introduced in [8]. This method, known as the accessible fuzzy subset construction, is the fuzzy analogue of the well-known subset construction used for non-deterministic finite automata. When the semiring reduct of the underlying truth value structure is locally finite, this procedure completes within a finite number of steps. However, this is not guaranteed in the general case. For example, if the underlying truth structure is the product structure, the process may not terminate. Moreover, even when termination occurs, the resulting crisp-deterministic fuzzy automaton can have an excessively large number of states. To mitigate these issues, several improved procedures have been proposed in [4, 5, 7, 14], offering faster solutions and automata with fewer states.

Both crisp- and fuzzy determinization procedures can be further improved using simulations. This enhancement was achieved for crisp-determinization in [7] and for fuzzy determinization in [17]. Simulations significantly enhance determinization procedures, sometimes yielding finite crisp-deterministic or fuzzy deterministic automata in cases where other methods result in infinite ones. Such enhancements play a critical role in Brzozowski-type determinization procedures [4, 11, 12]. However, simulations themselves present challenges. If the underlying truth value structure is not locally finite, the computation of both ordinary and weak simulations may not terminate within a finite number of steps. This issue arises, for instance, with fuzzy automata over the product structure. One way to overcome these challenges is through approximate determinization using approximate simulations.

Approximate determinization of fuzzy finite automata over the product structure was investigated in [19] using a parametric modification of the product t-norm within the pre-determinization framework. This approach covered both crisp- and fuzzy determinization. The resulting deterministic automaton behaves similarly to the original fuzzy automaton, except for words with acceptance degrees below a specified parameter, where the membership value differences also remain within the parameter's bounds. By choosing a sufficiently small parameter, the algorithms from [19] generate deterministic automata whose behavior closely approximates that of the original fuzzy automaton.

The parametric modification of the product t-norm ensures the structure has a locally finite semiring reduct, guaranteeing algorithm termination in a finite number of steps. However, the modified t-norm lacks left continuity, preventing the residuum operation necessary for computing both ordinary and weak simulations. To overcome this limitation, we adopt the approach from [18], which truncates the product structure based on the given parameter. This truncation yields a new residuated structure with a locally finite semiring reduct, which facilitates efficient computation of approximate weak simulations [18]. Building on this structure, we develop new algorithms for the approximate determinization of fuzzy automata, leveraging approximate weak simulations to enhance efficiency. A key contribution of this paper is a new determinization method that always yields a finite deterministic fuzzy automaton, even in cases where general determinization procedures fail to terminate. Unlike traditional methods, our approach guarantees language equivalence with the original fuzzy automaton on all words whose acceptance degrees exceed a predefined parameter ε, thus providing an effective and practical approximation framework.

The paper is organized as follows. Section 2 presents fundamental concepts and notation related to complete residuated lattices and specific structures of fuzzy truth values, with a particular focus on truncated product structures. It also reviews key concepts and results concerning fuzzy sets, relations, languages, automata, deterministic fuzzy automata, and approximate weak simulations and bisimulations. Section 3 introduces the main results for approximate crisp-determinization using approximate weak simulations. Section 4 presents an algorithm for constructing a fuzzy deterministic finite automaton which is equivalent to a given fuzzy automaton for all words which are accepted by the original automaton with the degree greater or equal to ε, where $\varepsilon > 0$ is a very small value.

2. Preliminmaries

2.1. *Complete residuated lattices and particular structures of membership values*

A *resuduated lattice* is an algebra $\mathbb{L} = (\mathbb{L}, \vee, \wedge, \otimes, \to, 0, 1)$ that consists of a non-empty set L, four binary operations $\vee$, $\wedge$, $\otimes$ and $\to$ on L, and two constants 0 and 1 from L, such that the following conditions are satisfied:

(L1) $(\mathbb{L}, \vee, \wedge, 0, 1)$ is a bounded lattice with the least element 0 and the greatest element 1;

(L2) $(\mathbb{L}, \otimes, 1)$ is a commutative semigroup with the identity 1;

(L3) the operations $\otimes$ and $\to$ satisfy the *adjunction property*: for all $a, b, c \in \mathbb{L}$,

$$a \otimes b \leqslant c \quad \Leftrightarrow \quad a \leqslant b \to c,$$

where $\leqslant$ is the ordering in the lattice $(\mathbb{L}, \vee, \wedge, 0, 1)$.

The operation $\otimes$ is called the *multiplication*, and the operation $\to$ is called the *residuum*. The operation $\leftrightarrow$ defined by $(a \leftrightarrow b) = (a \to b) \wedge (b \to a)$, for arbitrary $a, b \in \mathbb{L}$, is called the *biresiduum*. In addition, if $(\mathbb{L}, \vee, \wedge, 0, 1)$ is a complete lattice, then $\mathbb{L}$ is called a *complete residuated lattice*.

In applications of fuzzy sets, the most commonly used complete residuated lattices are the *Gödel*, *Łukasiewicz* and *product structures*, whose carrier set is the real unit interval $\mathbb{I} = [0, 1]$, the operations $\vee$ and $\wedge$ are given by $a \vee b = \max(a, b)$ and $a \wedge b = \min(a, b)$, for all $a, b \in \mathbb{I}$, and the adjoint operations $\otimes$ and $\to$ are defined, for arbitrary $a, b \in \mathbb{I}$, as follows:

$$\textit{Gödel structure}: \qquad a \otimes b = a \wedge b, \quad (a \to b) = \begin{cases} 1 & \text{if } a \leqslant b \\ b & \text{if } a > b \end{cases}; \tag{1}$$

$$\textit{Łukasiewicz structure}: \qquad a \otimes b = \max(a + b - 1, 0), \quad (a \to b) = \min(1 - a + b, 1); \tag{2}$$

$$\textit{product structure}: \qquad a \otimes b = a \cdot b, \quad (a \to b) = \begin{cases} 1 & \text{if } a \leqslant b \\ \frac{b}{a} & \text{if } a > b \end{cases}. \tag{3}$$

The product structure will be denoted by the same letter $\mathbb{I}$ as its carrier, i.e., $\mathbb{I} = (\mathbb{I}, \wedge, \vee, \otimes, \to, 0, 1)$, where $\otimes$ and $\to$ are defied as in (3).

The reduct $(\mathbb{L}, \vee, \otimes, 0, 1)$ of a residuated lattice $\mathbb{L} = (\mathbb{L}, \vee, \wedge, \otimes, \to, 0, 1)$ is a semiring (for a definition we refer the reader to [3]), and it is called the *semiring reduct* of $\mathbb{L}$.

An algebra is called *locally finite* if each of its finitelly generated subalgebras is finite. In particular, a semiring is locally finite if each of its finitelly generated subsemirings is finite, and a monoid is locally finite if each of its finitelly generated submonoids is finite. Let us point out that the semiring reducts of the Gödel and Łukasiewicz structures are locally finite, but the semiring reduct of the product structure is not locally finite.

In order to obtain a locally finite structure that approximate the product structure, in [18] the construction that we present in the sequel was given. Instead of the real unit interval $\mathbb{I} = [0, 1]$, the carrier set of this new structure is assumed to be the interval $\mathbb{I}_\varepsilon = [\varepsilon, 1]$, for a given $\varepsilon \in (0, 1)$.

Theorem 2.1. *For a given $\varepsilon \in (0, 1)$, let $*_\varepsilon$ and $\to_\varepsilon$ be binary operations on $\mathbb{I}_\varepsilon$ defined by*

$$a *_\varepsilon b = \begin{cases} a \cdot b & \text{if } a \cdot b \geqslant \varepsilon, \\ \varepsilon & \text{if } a \cdot b < \varepsilon, \end{cases} \qquad a \to_\varepsilon b = \begin{cases} 1 & \text{if } a \leqslant b, \\ \frac{b}{a} & \text{if } a > b, \end{cases}$$

for arbitrary $a, b \in \mathbb{I}_\varepsilon$. Then $\mathbb{I}_\varepsilon = (\mathbb{I}_\varepsilon, \wedge, \vee, *_\varepsilon, \to_\varepsilon, \varepsilon, 1)$ is a complete residuated lattice and its semiring reduct $(\mathbb{I}_\varepsilon, \vee, *_\varepsilon, \varepsilon, 1)$ is a locally finite semiring.

The proof of this theorem can be seen in [18] (Theorems 3.1 and 3.2).

The complete residuated lattice $\mathbb{I}_\varepsilon$ is called the *ε-truncated product structure*, or just the *truncated product structure*, when it is understood in relation to which ε this truncation is performed.

2.2. Fuzzy subsets and fuzzy relations

Throughout this paper, $\mathbb{L}$ will denote an arbitrary complete residuated lattice.

For a nonempty set A, a *fuzzy subset* of A is defined as any function $\alpha : A \to \mathbb{L}$, and the set of all fuzzy subsets of A is denoted by $\mathbb{L}^A$. For two fuzzy sets $\alpha, \beta \in \mathbb{L}^A$, if $\alpha(a) \leqslant \beta(a)$, for every $a \in A$, then we say that α is *included in* β, and we write $\alpha \leqslant \beta$. For an arbitrary family $\{\alpha_j\}_{j\in J}$ of fuzzy subsets of A, the *union* $\bigvee_{j\in J} \alpha_j$ and the *intersection* $\bigwedge_{j\in J} \alpha_j$ of this family are defined coordinatewise (see [20, 21] for more details).

A *fuzzy relation* between nonempty sets A and B is defined as any fuzzy subset of $A \times B$, and the equality, inclusion, union and intersection of fuzzy relations are defined as for fuzzy sets. Following notation that we use for fuzzy sets, the set of all fuzzy relations between A and B, with membership values in $\mathbb{L}$, will be denoted by $\mathbb{L}^{A\times B}$. The *inverse* of a fuzzy relation $\phi \in \mathbb{L}^{A\times B}$ is defined as the fuzzy relation $\phi^{-1} \in \mathbb{L}^{B\times A}$ given by $\phi^{-1}(b, a) = \phi(a, b)$, for all $a \in A$ and $b \in B$.

Given fuzzy relations $\phi \in \mathbb{L}^{A\times B}$, $\varphi \in \mathbb{L}^{B\times C}$ and fuzzy sets $\alpha \in \mathbb{L}^A$, $\beta \in \mathbb{L}^B$, where A, B and C are non-empty sets, the compositions $\phi \circ \varphi \in \mathbb{L}^{A\times C}$, $\alpha \circ \phi \in \mathbb{L}^B$ and $\phi \circ \beta \in \mathbb{L}^A$ are defined by

$$(\phi \circ \varphi)(a, c) = \bigvee_{b\in B} \phi(a, b) \otimes \varphi(b, c), \tag{4}$$

$$(\alpha \circ \phi)(b) = \bigvee_{a\in A} \alpha(a) \otimes \phi(a, b), \tag{5}$$

$$(\phi \circ \beta)(a) = \bigvee_{b\in B} \phi(a, b) \otimes \beta(b), \tag{6}$$

for all $a \in A$, $b \in B$ and $c \in C$. If $\alpha, \beta \in \mathbb{L}^A$, then we also define $\alpha \circ \beta \in \mathbb{L}$ by

$$\alpha \circ \beta = \bigvee_{a\in A} \alpha(a) \otimes \beta(a). \tag{7}$$

In particular, for fuzzy relations and fuzzy sets with membership values in the ε-truncated product structure $\mathbb{I}_\varepsilon$, for some $\varepsilon \in (0, 1)$, the composition operations defined by (4)–(7) are denoted by $\circ_\varepsilon$.

At the end of this subsection we deal with fuzzy sets and relations with membership values in the product structure $\mathbb{I}$. For a fuzzy set $\alpha \in \mathbb{I}^A$ and $\varepsilon \in (0, 1)$, the *ε-truncation* of α is a fuzzy set $\alpha_\varepsilon \in \mathbb{I}_\varepsilon^{A\times A}$ defined by

$$\alpha_\varepsilon(a) = \begin{cases} \alpha(a) & \text{if } \alpha(a) \geqslant \varepsilon, \\ \varepsilon & \text{if } \alpha(a) < \varepsilon, \end{cases}$$

and, in particular, the *ε-truncation* of a fuzzy relation $\varphi \in \mathbb{L}^{A\times A}$ is a fuzzy relation $\varphi_\varepsilon \in \mathbb{I}_\varepsilon^{A\times A}$ defined by

$$\varphi_\varepsilon(a, b) = \begin{cases} \varphi(a, b) & \text{if } \varphi(a, b) \geqslant \varepsilon, \\ \varepsilon & \text{if } \varphi(a, b) < \varepsilon. \end{cases}$$

2.3. *Fuzzy languages and fuzzy automata*

Throughout this paper, Σ will denote a finite non-empty set called an *alphabet*, whose elements are called *letters*, Σ^* will denote the free monoid over Σ, e will denote the identity of Σ^*, called the *empty word*, and Σ^+ will denote the free semigroup over Σ, that is, $\Sigma^+ = \Sigma^* \setminus \{e\}$. A *fuzzy language* over Σ, with membership values in $\mathbb{L}$, is defined as any fuzzy subset of Σ^*, i.e., as any mapping from Σ^* to $\mathbb{L}$.

A *fuzzy automaton* over Σ and $\mathbb{L}$ (abbrev. FA) is a quadruple $\mathcal{A} = (Q, \Sigma, I, T, F)$, where

- Q is a non-empty set, called the *set of states;*
- $I \in \mathbb{L}^Q$ is the *fuzzy set of initial states;*
- $T \in \mathbb{L}^{Q\times\Sigma\times Q}$ is the *fuzzy transition function;*
- $F \in \mathbb{L}^Q$ is the *fuzzy set of final states.*

If the set of states Q is finite, then $\mathcal{A}$ is called a *fuzzy finite automaton* over Σ and $\mathbb{L}$ (abbrev. FfA).

The fuzzy transition function $T : Q\times X\times Q \to \mathbb{L}$ determines the family $\{T_u\}_{u\in X^*} \subseteq \mathbb{L}^{Q\times Q}$ of fuzzy relations, which are defined inductively as follows:

- T_e is the equality relation on Q, that is, $T_e(a, a) = 1$, for every $a \in Q$, and $T_e(a, b) = 0$, for all $a, b \in Q$ such that $a \neq b$;
- $T_x(a, b) = T(a, x, b)$, for all $a, b \in Q$ and $x \in X$,
- $T_{ux} = T_u \circ T_x$, for all $u \in \Sigma^*$ and $x \in \Sigma$.

Due to the associativity of the composition of relations, we have that $T_{uv} = T_u \circ T_v$, for all $u, v \in X^*$. This means that if $u = x_1x_2 \dots x_n$, for $x_1, x_2, \dots, x_n \in \Sigma$, then $T_u = T_{x_1} \circ T_{x_2} \circ \dots \circ T_{x_n}$. Each T_u we call a *fuzzy transition relation* of $\mathcal{A}$.

The *behavior* of a fuzzy automaton $\mathcal{A} = (Q, \Sigma, I, T, F)$ is a mapping $[\![\mathcal{A}]\!] : \Sigma^* \to \mathbb{L}$, i.e., a fuzzy subset of Σ^*, defined by

$$[\![\mathcal{A}]\!](u) = I \circ T_{x_1} \circ T_{x_2} \circ \cdots \circ T_{x_n} \circ F = I \circ T_u \circ F, \tag{8}$$

for every $u = x_1x_2 \dots x_n \in \Sigma^+$, where $x_1, x_2, \dots, x_n \in \Sigma$, and

$$[\![\mathcal{A}]\!](e) = I \circ F. \tag{9}$$

We say that $[\![\mathcal{A}]\!]$ is the *fuzzy language accepted* (*recognized*) *by* $\mathcal{A}$, and we also say that $\mathcal{A}$ *accepts* (*recognizes*) the fuzzy language $[\![\mathcal{A}]\!]$. In addition, we say that a word $u \in \Sigma^*$ is *accepted with degree* $[\![\mathcal{A}]\!](u)$, and that $[\![\mathcal{A}]\!](u)$ is the *acceptance degree* of u.

Two fuzzy automata $\mathcal{A}$ and $\mathcal{B}$ over Σ and $\mathbb{L}$ are said to be *equivalent* if $[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!]$, i.e., if they accept the same fuzzy language.

Let $\mathcal{A} = (Q, \Sigma, I, T, F)$ be a fuzzy finite automaton over the product structure $\mathbb{I}$ and $\varepsilon \in (0, 1)$. We define a new fuzzy finite automaton $\mathcal{A}_\varepsilon = (Q_\varepsilon, \Sigma, I^\varepsilon, T^\varepsilon, F^\varepsilon)$, called the *ε-copy* of $\mathcal{A}$, which has the same set of states, i.e., $Q_\varepsilon = Q$, whereas the fuzzy set of initial states I^ε, the fuzzy set of final states F^ε, and the fuzzy transition function T^ε are ε-truncations of I, F and T, respectively. The *behavior* of $\mathcal{A}_\varepsilon$ is defined over $\mathbb{I}_\varepsilon$, which means that

$$[\![\mathcal{A}_\varepsilon]\!](u) = I^\varepsilon \circ_\varepsilon T^\varepsilon_{x_1} \circ_\varepsilon T^\varepsilon_{x_2} \circ_\varepsilon \cdots \circ_\varepsilon T^\varepsilon_{x_n} \circ_\varepsilon F^\varepsilon = I^\varepsilon \circ_\varepsilon T^\varepsilon_u \circ_\varepsilon F^\varepsilon. \tag{10}$$

for every $u = x_1x_2 \dots x_n \in \Sigma^+$, where $x_1, x_2, \dots, x_n \in \Sigma$, and

$$[\![\mathcal{A}_\varepsilon]\!](e) = I^\varepsilon \circ_\varepsilon F^\varepsilon. \tag{11}$$

The following was proven in [18]:

Proposition 2.2. *Let $\mathcal{A}$ be a fuzzy finite automaton over the product structure $\mathbb{I}$, let $\varepsilon \in (0, 1)$, and let $\mathcal{A}_\varepsilon$ be the ε-copy of $\mathcal{A}$. Then the fuzzy language accepted by $\mathcal{A}_\varepsilon$ is the ε-truncation of the fuzzy language accepted by $\mathcal{A}$, that is*

$$[\![\mathcal{A}_\varepsilon]\!] = [\![\mathcal{A}]\!]_\varepsilon. \tag{12}$$

For a given $\varepsilon \in (0, 1)$, two fuzzy finite automata $\mathcal{A}$ and $\mathcal{B}$ over the product structure $\mathbb{I}$ and the same alphabet Σ are said to be *ε-equivalent* if their ε-copies are equivalent, that is, if $[\![\mathcal{A}]\!]_\varepsilon = [\![\mathcal{B}]\!]_\varepsilon$. This means that the fuzzy languages accepted by ε-equivalent automata $\mathcal{A}$ and $\mathcal{B}$ must coincide on all words whose acceptance degree in $\mathcal{A}$ or $\mathcal{B}$ is greater than or equal to ε, and they can differ only on those words whose acceptance degree in $\mathcal{A}$ and $\mathcal{B}$ is less than ε.

2.4. Deterministic fuzzy automata

Let $\mathcal{A} = (Q, \Sigma, I, T, F)$ be a fuzzy automaton over a complete residuated lattice $\mathbb{L}$. If $I = \{p/1\}$ is a singleton fuzzy set with the membership degree 1 (i.e., a singleton crisp set), and $T : Q \times \Sigma \to Q$ is a crisp mapping, then $\mathcal{A}$ is called a *crisp-deterministic fuzzy automaton* (abbrev. cDFA). If the set of states Q is finite, then we call $\mathcal{A}$ a *crisp-deterministic fuzzy finite automaton* (abbrev. cDFfA). This model of deterministic fuzzy automata and the corresponding determinization procedures were investigated in [1, 4, 5, 7, 8, 14].

As usual when working with classical deterministic automata, for a crisp-deterministic fuzzy automaton $\mathcal{A} = (Q, \Sigma, I, T, F)$ over $\mathbb{L}$, the transition function $T : Q\times\Sigma \to Q$ is extended to the function $T^* : Q\times\Sigma^* \to Q$ by $T^*(a, e) = a$ and $T^*(a, ux) = T(T^*(a, u), x)$, for all $a \in Q$, $u \in \Sigma^*$ and $x \in \Sigma$. For the sake of simplicity, we will write T instead of T^*. The behavior of the crisp-deterministic fuzzy automaton $\mathcal{A}$ is now represented by

$$[\![\mathcal{A}]\!](u) = F(T(p, u)), \tag{13}$$

for every $u \in \Sigma^*$.

Further, let $\mathcal{A} = (Q, \Sigma, I, T, F)$ be a fuzzy finite automaton over $\mathbb{L}$. The families $\{I_u \,|\, u \in \Sigma^*\}$ and $\{F_u \,|\, u \in \Sigma^*\}$ of fuzzy subsets of Q are defined inductively as follows:

$$I_e = I, \qquad I_{ux} = I_u \circ T_x, \tag{14}$$

$$F_e = F, \qquad F_{xu} = T_x \circ F_u, \tag{15}$$

for all $u \in \Sigma^*$ and $x \in \Sigma$. The *Nerode automaton* of $\mathcal{A}$ is defined as a crisp-deterministic fuzzy automaton $\mathcal{N}(\mathcal{A}) = (Q_N, \Sigma, \{I_e/1\}, T_N, F_N)$, whose set of states is $Q_N = \{I_u \mid u \in \Sigma^*\}$, and the transition function $T_N : Q_N \times \Sigma \to Q_N$ and the fuzzy set of final states $F_N : Q_N \to \mathbb{L}$ are given by:

$$T_N(I_u, x) = I_{ux}, \qquad F_N(I_u) = I_u \circ F, \tag{16}$$

for all $u \in \Sigma^*$ and $x \in \Sigma$. It is easy to verify that these functions are well-defined. In addition, for any $u \in \Sigma^*$ we have that

$$[\![\mathcal{N}(\mathcal{A})]\!](u) = F_N(T_N(I_e, u)) = F_N(I_u) = I_u \circ F = I \circ T_u \circ F = [\![\mathcal{A}]\!](u),$$

which means that $\mathcal{N}(\mathcal{A})$ s equivalent to $\mathcal{A}$.

The *reverse Nerode automaton* of a fuzzy finite automaton $\mathcal{A}$, denoted by $\mathcal{R}(\mathcal{A})$, is the Nerode automaton of the reverse fuzzy finite automaton $\overline{\mathcal{A}}$ of $\mathcal{A}$, i.e., $\mathcal{R}(\mathcal{A}) = \mathcal{N}(\overline{\mathcal{A}})$. It can be represented by $\mathcal{R}(\mathcal{A}) = (Q_R, \Sigma, \{F_e/1\}, T_R, F_R)$, where the set of states is $Q_R = \{F_u \,|\, u \in X^*\}$, and the transition function $T_R : Q_R \times \Sigma \to Q_R$ and the fuzzy set of final states $F_R : Q_R \to \mathbb{L}$ are given by

$$T_R(F_u, x) = F_{xu}, \qquad F_R(F_u) = I \circ F_u, \tag{17}$$

for all $u \in \Sigma^*$ and $x \in \Sigma$.

2.5. Approximate weak simulations and bisimulations

Let $\mathcal{A} = (Q, \Sigma, I, T, F)$ be a fuzzy finite automaton over the product structure $\mathbb{I}$, and let $\varepsilon \in (0, 1)$. A fuzzy relation $\varphi \in \mathbb{I}^{Q\times Q}$ is called an *ε-weak forward simulation* on $\mathcal{A}$ if its ε-truncation φ_ε is an ordinary weak forward simulation on $\mathcal{A}_\varepsilon$, i.e., if the following inequalities are satisfied:

$$I^\varepsilon \leqslant I^\varepsilon \circ_\varepsilon \varphi_\varepsilon^{-1}, \tag{18}$$

$$\varphi_\varepsilon^{-1} \circ_\varepsilon F_u^\varepsilon \leqslant F_u^\varepsilon, \quad \text{for all } u \in X^*. \tag{19}$$

In addition, φ is called an *ε-weak forward bisimulation* on $\mathcal{A}$ if its ε-truncation is an ordinary weak forward bisimulation on $\mathcal{A}_\varepsilon$, i.e., if φ_ε satisfies (18) and (19) together with:

$$I^\varepsilon \leqslant I^\varepsilon \circ_\varepsilon \varphi_\varepsilon, \tag{20}$$

$$\varphi_\varepsilon \circ_\varepsilon F_u^\varepsilon \leqslant F_u^\varepsilon, \quad \text{for all } u \in X^*. \tag{21}$$

If a fuzzy relation φ is reflexive, then inequality (19) is equivalent to

$$\varphi_\varepsilon^{-1} \circ_\varepsilon F_u^\varepsilon = F_u^\varepsilon, \quad \text{for all } u \in X^*. \tag{22}$$

Theorem 2.3. *Let $\mathcal{A} = (Q, \Sigma, I, T, F)$ be a fuzzy finite automaton over the product structure $\mathbb{I}$, let $\varepsilon \in (0, 1)$ and let φ be an ε-weak forward bisimulation on $\mathcal{A}$. The greatest element of the set:*

$$H = \{\psi \in \mathbb{I}^{Q\times Q} \,|\, \psi_\varepsilon = \varphi_\varepsilon\},$$

is relation φ_ε. Let ψ be an element of the set H, then $\psi_\varepsilon = \varphi_\varepsilon$.

Proof. For a given fuzzy relation φ, its ε-truncation φ_ε is an element of the set H.

According to Proposition 3.4. a) [18], for every relation ψ, holds $\psi \leqslant \psi_\varepsilon$. Therefore, $\psi \leqslant \varphi_\varepsilon$, which means that φ_ε is the greatest element of set H. □

This fact completes Theorem 4.6 and Corollary 4.7 [18].

3. Approximate crisp-determinization by means of ε-weak simulations

Let $\mathcal{A} = (Q, \Sigma, I, T, F)$ be a fuzzy finite automaton over the product structure $\mathbb{I}$. Let $\varepsilon \in (0, 1)$, and let $\varphi \in \mathbb{I}^{Q\times Q}$ be a fuzzy relation. We define a family of fuzzy relations $\{\varphi_v\}_{v\in\Sigma^*}$, inductively as follows. For empty word e:

$$\varphi_e = I^\varepsilon \circ_\varepsilon \varphi_\varepsilon^{-1}, \tag{23}$$

for all $u \in \Sigma^*$ and $x \in \Sigma$:

$$\varphi_{ux} = \varphi_u \circ_\varepsilon T_x^\varepsilon \circ_\varepsilon \varphi_\varepsilon^{-1}. \tag{24}$$

According to the fact that the semiring reduct of $\mathbb{I}_\varepsilon$ is locally finite, the family of fuzzy relations $\{\varphi_v\}_{v\in\Sigma^*}$ is finite. The automaton $\mathcal{A}_\varphi = (Q_\varphi, \Sigma, \varphi_e, T_\varphi, F_\varphi)$, where $Q_\varphi = \{\varphi_u \,|\, u \in \Sigma^*\}$, and let $T_\varphi : Q_\varphi \times \Sigma \to Q_\varphi$ and $F_\varphi : Q_\varphi \to \mathbb{I}_\varepsilon$ are given by:

$$T_\varphi(\varphi_u, x) = \varphi_{ux}, \quad F_\varphi(\varphi_u) = \varphi_u \circ_\varepsilon F^\varepsilon,$$

for each word $u \in \Sigma^*$ and every letter $x \in \Sigma$ is crisp-deterministic fuzzy finite automaton [6]. Furthermore, it was proven in [6] that if φ were to be a reflexive ε-weak forward simulation on $\mathcal{A}$, then $\mathcal{A}_\varphi$ is ε-equivalent to $\mathcal{A}$.

Theorem 3.1. *Let $\mathcal{A} = (Q, \Sigma, I, T, F)$ be a fuzzy automaton and $\varepsilon \in (0, 1)$, such that ε. Let φ be the greatest ε_1-weak forward bisimulation on $\mathcal{A}$. Then the automaton $\mathcal{A}_{\varphi_2}$ is isomorphic to the Nerode automaton of the after-set fuzzy automaton $\mathcal{A}/\varphi$ over the ε-truncated product structure $\mathbb{I}_\varepsilon$.*

Proof. First we will describe the Nerode automaton of the after-set fuzzy automaton $\mathcal{A}/\varphi$ over the ε-truncated product structure $\mathbb{I}_\varepsilon$. Therefore, instead of considering the automaton $\mathcal{A}$ we will observe the ε-copy of the automaton $\mathcal{A}$, and instead of φ we will consider its ε-truncation φ_ε. According to Corollary 4.7, φ_ε is a fuzzy equivalence, which also means that $\varphi_\varepsilon = \varphi_\varepsilon^{-1}$.

Now, denote by $\mathcal{C} = (C, \Sigma, I_C, T_C, F_C)$ the after-set automaton $\mathcal{A}_\varepsilon/\varphi_\varepsilon$. The set of states is $C = \{a\varphi_\varepsilon \,|\, a \in Q\}$. The fuzzy set of initial states $I_C : C \to \mathbb{I}_\varepsilon$ and fuzzy set of final states $F_C : C \to \mathbb{I}_\varepsilon$ are defined by:

$$I_C(a\varphi_\varepsilon) = I^\varepsilon \circ_\varepsilon \varphi_\varepsilon(a), \quad F_C(a\varphi_\varepsilon) = \varphi_\varepsilon \circ_\varepsilon F^\varepsilon(a), \quad \text{for all } a\varphi_\varepsilon \in C.$$

The fuzzy transition function: $T_C : C \times \Sigma \times C \to \mathbb{I}_\varepsilon$ is defined by:

$$T_C(a\varphi_\varepsilon, x, b\varphi_\varepsilon) = \varphi_\varepsilon \circ_\varepsilon T_x^\varepsilon \circ_\varepsilon \varphi_\varepsilon(a, b), \quad \text{for all } a, b \in Q.$$

Now, using the fact that $\varphi_\varepsilon = \varphi_\varepsilon^{-1}$, when we apply the Nerode construction to this automaton, we obtain exactly the automaton $\mathcal{A}_\varphi$. □

In the sequel, we propose the new construction of the crisp-deterministic fuzzy automaton, which is an adaptation of the Children automaton[7].

Let $\mathcal{A} = (Q, \Sigma, I, T, F)$ be a fuzzy finite automaton over the product structure $\mathbb{I}$. The size of the alphabet Σ will be denoted by k (with $x_1, ..., x_k$ we will denote all the letters of Σ). Let $\varepsilon \in (0, 1)$ and let $\varphi \in \mathbb{I}^{Q\times Q}$ be a fuzzy relation. The family of fuzzy relations $\{\varphi_v\}_{v\in\Sigma^*}$ is given by formulas (23)-(24). Let $k+1$-tuple $\varphi_u^{\varepsilon c}$ be given by the following formula:

$$\varphi_u^{\varepsilon c} = (\varphi_{ux_1}, \varphi_{ux_2}, ..., \varphi_{ux_k}, \varphi_u \circ_\varepsilon F^\varepsilon),$$

for any $u \in X^*$. Since the family $\{\varphi_v\}_{v\in\Sigma^*}$ is finite, the set $Q_\varphi^{\varepsilon c} = \{\varphi_u^{\varepsilon c} \mid u \in \Sigma^*\}$ is also finite.

Finally, let $T_\varphi^{\varepsilon c} : Q_\varphi^{\varepsilon c} \times \Sigma \to Q_\varphi^{\varepsilon c}$ and $F_\varphi^{\varepsilon c} : Q_\varphi^{\varepsilon c} \to \mathbb{I}_\varepsilon$ be defined as:

$$T_\varphi^{\varepsilon c}(\varphi_u^{\varepsilon c}, x) = \varphi_{ux}^{\varepsilon c}, \quad F_\varphi^{\varepsilon c}(\varphi_u^{\varepsilon c}) = \varphi_u \circ_\varepsilon F^\varepsilon,$$

for each word $u \in \Sigma^*$ and every letter $x \in \Sigma$. To prove that $T_\varphi^{\varepsilon c}$ is well defined, consider $\varphi_u^{\varepsilon c}$ and $\varphi_v^{\varepsilon c}$ such that $\varphi_u^{\varepsilon c} = \varphi_v^{\varepsilon c}$. From the definition of $\varphi^{\varepsilon c}$, it follows that for every $x \in \Sigma$, $\varphi_{ux} = \varphi_{vx}$ holds. Which implies that,

$$\begin{aligned} T_\varphi^{\varepsilon c}(\varphi_u^{\varepsilon c}, x) &= (\varphi_{uxx_1}, \varphi_{uxx_2}, ..., \varphi_{uxx_k}, \varphi_{ux} \circ_\varepsilon F^\varepsilon) = \\ &= (\varphi_{ux} \circ_\varepsilon T_{x_1}^\varepsilon \circ_\varepsilon \varphi_\varepsilon^{-1}, ..., \varphi_{ux} \circ_\varepsilon T_{x_k}^\varepsilon \circ_\varepsilon \varphi_\varepsilon^{-1}, \varphi_{ux} \circ_\varepsilon F^\varepsilon) \\ &= (\varphi_{vx} \circ_\varepsilon T_{x_1}^\varepsilon \circ_\varepsilon \varphi_\varepsilon^{-1}, ..., \varphi_{vx} \circ_\varepsilon T_{x_k}^\varepsilon \circ_\varepsilon \varphi_\varepsilon^{-1}, \varphi_{vx} \circ_\varepsilon F^\varepsilon) \\ &= T_\varphi^{\varepsilon c}(\varphi_v^{\varepsilon c}, x). \end{aligned}$$

Accordingly, T_φ is well-defined and $\mathcal{A}_\varphi^{\varepsilon c} = (Q_\varphi^{\varepsilon c}, \Sigma, \varphi_e^{\varepsilon c}, T_\varphi^{\varepsilon c}, F_\varphi^{\varepsilon c})$ is a crisp-deterministic fuzzy finite automaton.

Theorem 3.2. *Let $\mathcal{A} = (Q, \Sigma, I, T, F)$ be a fuzzy finite automaton over the product structure $\mathbb{I}$, let $\varepsilon \in (0, 1)$, and let φ be a reflexive ε-weak forward simulation on $\mathcal{A}$.*

The automaton $\mathcal{A}_\varphi^{\varepsilon c} = (Q_\varphi^{\varepsilon c}, \Sigma, \varphi_e^{\varepsilon c}, T_\varphi^{\varepsilon c}, F_\varphi^{\varepsilon c})$ is an accessible crisp-deterministic fuzzy finite automaton which satisfies:

$$[\![\mathcal{A}_\varphi^{\varepsilon c}]\!] = [\![\mathcal{A}_\varepsilon]\!]. \tag{25}$$

In other words, $\mathcal{A}_\varphi^{\varepsilon c}$ is an accessible cdffa ε-equivalent to $\mathcal{A}$.

Proof. According to the definition of automaton $\mathcal{A}_\varphi^{\varepsilon c}$ it directly follows that it is accessible and we already proved that it is a finite crisp-deterministic automaton. Therefore, it remains to be shown that the language is ε-equivalent to $\mathcal{A}$.

Since φ is an ε- weak forward simulation, system of equations (22) holds and thus for every word $u = x_{i_1} \dots x_{i_n} \in \Sigma^*$, where $x_{i_1}, ..., x_{i_n} \in \Sigma$ and $n \in \mathbb{N}$ holds:

$$\begin{aligned} [\![\mathcal{A}_\varphi^{\varepsilon c}]\!](u) &= F_\varphi^{\varepsilon c}(T_\varphi^{\varepsilon c}(\varphi_e^{\varepsilon c}, u)) = F_\varphi^{\varepsilon c}(\varphi_u^{\varepsilon c}) = \varphi_u \circ_\varepsilon F^{\varepsilon c} = \\ &= (I^\varepsilon \circ_\varepsilon \varphi_\varepsilon^{-1} \circ_\varepsilon T_{x_1}^\varepsilon \circ_\varepsilon \varphi_\varepsilon^{-1} \circ_\varepsilon T_{x_2}^\varepsilon \circ_\varepsilon \cdots \circ_\varepsilon \varphi_\varepsilon^{-1} \circ_\varepsilon T_{x_n}^\varepsilon \circ_\varepsilon \varphi_\varepsilon^{-1}) \circ_\varepsilon F^\varepsilon = \\ &= I^\varepsilon \circ_\varepsilon F_u^\varepsilon = I^\varepsilon \circ_\varepsilon T_u^\varepsilon \circ_\varepsilon F^\varepsilon = [\![\mathcal{A}_\varepsilon]\!](u). \end{aligned}$$

Similarly, it can be shown for an empty word equation $[\![\mathcal{A}_\varphi]\!](e) = [\![\mathcal{A}_\varepsilon]\!](e)$ also holds. Hence, equation (25) holds. □

Moreover, in the case where the fuzzy relation φ is a fuzzy equality, the automaton $\mathcal{A}_\varphi^{\varepsilon c}$ is the classical Children automaton of the Nerode automaton of ε-copy of automaton $\mathcal{A}$, and we will denote it by $\mathcal{A}_N^{\varepsilon c}$.

Theorem 3.3. *Let $\mathcal{A} = (Q, \Sigma, I, T, F)$ be a fuzzy finite automaton on the product structure, and let $\varepsilon \in (0, 1)$. Then the Children automaton $\mathcal{A}_N^{\varepsilon c} = (Q^{\varepsilon c}, \Sigma, I_e^{\varepsilon c}, T^{\varepsilon c}, F^{\varepsilon c})$ is a homomorphic image of the Nerode automaton $\mathcal{A}_N^\varepsilon = (Q_N, \Sigma, I_e^\varepsilon, T_N^\varepsilon, F_N^\varepsilon)$ of the ε-copy of the automaton $\mathcal{A}$, and $|\mathcal{A}_N^{\varepsilon c}| \leqslant |\mathcal{A}_N^\varepsilon|$.*

Proof. Let $f : Q_N \to Q^{\varepsilon c}$ be defined as:

$$f(I_u^{\varepsilon}) = I_u^{\varepsilon c}, \quad u \in \Sigma^*.$$

Let $u, v \in X^*$ be words, such that $I_u^{\varepsilon} = I_v^{\varepsilon}$. Then:

$$f(I_u^{\varepsilon}) = I_u^{\varepsilon c} = (I_{ux_1}^{\varepsilon}, I_{ux_2}^{\varepsilon}, ..., I_{ux_k}^{\varepsilon}, I_u^{\varepsilon} \circ_{\varepsilon} F^{\varepsilon}) = (I_u^{\varepsilon} \circ_{\varepsilon} T_{x_1}, I_u^{\varepsilon} \circ_{\varepsilon} T_{x_2}, ..., I_u^{\varepsilon} \circ_{\varepsilon} T_{x_k}, I_u^{\varepsilon} \circ_{\varepsilon} F^{\varepsilon}) =$$
$$(I_v^{\varepsilon} \circ_{\varepsilon} T_{x_1}, I_v^{\varepsilon} \circ_{\varepsilon} T_{x_2}, ..., I_v^{\varepsilon} \circ_{\varepsilon} T_{x_k}, I_v^{\varepsilon} \circ_{\varepsilon} F^{\varepsilon}) = (I_{vx_1}^{\varepsilon}, I_{vx_2}^{\varepsilon}, ..., I_{vx_k}^{\varepsilon}, I_v^{\varepsilon} \circ_{\varepsilon} F^{\varepsilon}) = I_v^{\varepsilon c} = f(I_c^{\varepsilon}).$$

So, the mapping f is well defined. Clearly, it is a surjective mapping. Also for any $u \in \Sigma^*$ and $x \in \Sigma$:

$$f(T_N^{\varepsilon}(I_u^{\varepsilon}, x)) = T^{\varepsilon c}(f(I_u^{\varepsilon}), x).$$

Furthermore, $f(I_e^{\varepsilon}) = I_u^{\varepsilon c}$ and for all $u \in \Sigma^*$ holds $f(F_N^{\varepsilon}(I_u^{\varepsilon}) = F^{\varepsilon c}(f(I_u^{\varepsilon c})))$. So, the mapping is an surjective homomorphism, which implies $|\mathcal{A}_N^{\varepsilon c}| \leqslant |\mathcal{A}_N^{\varepsilon}|$. □

Theorem 3.4. *Let $\mathcal{A} = (Q, \Sigma, I, T, F)$ be a fuzzy finite automaton on the product structure, let $\varepsilon_1, \varepsilon_2 \in (0, 1)$, such that $\varepsilon_1 < \varepsilon_2$ and let φ_1 and φ_2 be reflexive ε_1 and ε_2-weak forward simulations on $\mathcal{A}$.*
Then, the automaton $\mathcal{A}_{\varphi_1}^{\varepsilon_1 c}$ is ε_2-equivalent to $\mathcal{A}_{\varphi_2}^{\varepsilon_2 c}$, that is, $[\![(\mathcal{A}_{\varphi_1}^{\varepsilon_1 c})]\!]_{\varepsilon_2} = [\![\mathcal{A}_{\varphi_2}^{\varepsilon_2 c}]\!]_{\varepsilon_2}$.

Proof. As has been shown in the previous theorem, for $\varepsilon_1, \varepsilon_2 \in (0, 1)$ holds $[\![\mathcal{A}_{\varphi_1}^{\varepsilon_1 c}]\!] = [\![\mathcal{A}_{\varepsilon_1}]\!]$ and $[\![\mathcal{A}_{\varphi_2}^{\varepsilon_2 c}]\!] = [\![\mathcal{A}_{\varepsilon_2}]\!]$. Next, according to (12) it follows that:

$$[\![\mathcal{A}_{\varphi_1}^{\varepsilon_1 c}]\!] = [\![\mathcal{A}]\!]_{\varepsilon_1}, \quad [\![\mathcal{A}_{\varphi_2}^{\varepsilon_2 c}]\!] = [\![\mathcal{A}]\!]_{\varepsilon_2}.$$

Now, according to Proposition 3.4 [18] for values $\varepsilon_1, \varepsilon_2 \in (0, 1)$, such that $\varepsilon_1 \leqslant \varepsilon_2$ holds:

$$[\![\mathcal{A}]\!]_{\varepsilon_2} = ([\![\mathcal{A}]\!]_{\varepsilon_1})_{\varepsilon_2}.$$

Hence, we conclude that $\mathcal{A}_{\varphi_1}^{\varepsilon_1 c}$ is ε_2-equivalent to $\mathcal{A}_{\varphi_2}^{\varepsilon_2 c}$.
□

Theorem 3.5. *Let $\mathcal{A} = (Q, \Sigma, I, T, F)$ be a fuzzy finite automaton on the product structure $\mathbb{I}$, let $\varepsilon \in (0, 1)$ and let φ be the greatest ε-weak forward simulation on $\mathcal{A}$. The automaton $\mathcal{A}_{\varphi}^{\varepsilon c}$ is isomorphic to the automaton $C_N^{\varepsilon c}$, where C_N is the Nerode automaton of the after-set automaton $\mathcal{A}/\varphi$.*

Proof. Follows directly from the Theorem 3.1. □

Theorem 3.6. *Let $\mathcal{A} = (Q, \Sigma, I, T, F)$ be a fuzzy finite automaton on the product structure $\mathbb{I}$, let $\varepsilon \in (0, 1)$ and let φ be the greatest ε-weak forward simulation on $\mathcal{A}$. Then, $\mathcal{A}_{\varphi}^{\varepsilon c}$ is homomorphic image of the automaton $\mathcal{A}_{\varphi}$, consequently $|\mathcal{A}_{\varphi}^{\varepsilon c}| \leqslant |\mathcal{A}_{\varphi}|$.*

Proof. Define the mapping $f : Q_{\varphi} \to Q_{\varphi}^{\varepsilon c}$ as $f(\varphi_u) = \varphi_u^{\varepsilon c}$, for every $u \in \Sigma^*$. It can easily be shown that f is an surjective homomorphism. □

4. Algorithm for constructing cDFfA $\mathcal{A}_{\varphi}^{\varepsilon c}$

This section includes an algorithm for constructing the children automaton $\mathcal{A}_{\varphi}^{\varepsilon c}$ of a given fuzzy finite automaton $\mathcal{A}$ over the product structure.

Algorithm 1: Construction of a children ε-equivalent cDFfA

Input: Fuzzy finite automaton $\mathcal{A} = (Q, \Sigma, I, T, F)$ over the product structure, $\Sigma = \{x_1, ..., x_n\}$ finite alphabet , $\varepsilon \in (0, 1)$ and the greatest ε-weak forward simulation φ on $\mathcal{A}$.

Output: $\mathcal{A}^{\varepsilon c}_{\varphi} = (Q^{\varepsilon c}_{\varphi}, \Sigma, \varphi^{\varepsilon c}_{e}, T^{\varepsilon c}_{\varphi}, F^{\varepsilon c}_{\varphi})$ – an accessible crisp-deterministic fuzzy finite automaton ε-equivalent to $\mathcal{A}$.

1 initialize $Q^{\varepsilon c}_{\varphi}$ with an empty set and *unprocessed* with an empty queue;
2 **foreach** $i \in \{1, ..., n\}$ **do**
3   compute $\varphi^0_i = I^\varepsilon \circ_\varepsilon \varphi^{-1}_\varepsilon \circ_\varepsilon T^\varepsilon_{x_i} \circ_\varepsilon \varphi^{-1}_\varepsilon$
4 compute $\varphi^0_{n+1} = I^\varepsilon \circ_\varepsilon \varphi^{-1}_\varepsilon \circ_\varepsilon F^\varepsilon$;
5 insert tuple $\varphi_0 = (\varphi^0_1, \varphi^0_2, ..., \varphi^0_n, \varphi^0_{n+1})$ (as an element) to both $Q^{\varepsilon c}_{\varphi}$ and *unprocessed*;
6 set $\varphi^{\varepsilon c}_{e} := \varphi_0$;
7 **while** *unprocessed is not empty* **do**
8   extract a $n + 1$-tuple $\varphi_0 = (\varphi^0_1, \varphi^0_2, ..., \varphi^0_n, \varphi^0_{n+1})$ from *unprocessed*;
9   **foreach** $j \in \{1, ..., n\}$ **do**
10   **foreach** $i \in \{1, ..., n\}$ **do**
11   compute $\varphi^1_i = \varphi^0_j \circ_\varepsilon T^\varepsilon_{x_i} \circ_\varepsilon \varphi^{-1}_\varepsilon$
12   compute $\varphi^1_{n+1} = \varphi^0_j \circ_\varepsilon F^\varepsilon$;
13         $\varphi_1 := (\varphi^1_1, \varphi^1_2, ..., \varphi^1_n, \varphi^1_{n+1})$;
14   **if** $\varphi_1 \notin Q^{\varepsilon c}_{\varphi}$ **then**
15   insert the state φ_1 (as an element) to both $Q^{\varepsilon c}_{\varphi}$ and *unprocessed*;
16             $F^{\varepsilon c}_{\varphi}(\varphi_1) = \varphi^1_{n+1}$
17         $T^{\varepsilon c}_{\varphi}(\varphi_0, x_j) = \varphi_1$

Here, we will give a brief explanation of the Algorithm 1. In the first six steps of the algorithm, we compute a tuple $\varphi^{\varepsilon c}_{e}$ which is the initial element of automaton $\mathcal{A}^{\varepsilon c}_{\varphi}$, and we add this element to the empty set $Q^{\varepsilon c}_{\varphi}$. Then, for each element φ_0 from $Q^{\varepsilon c}_{\varphi}$ and each letter x_j of the alphabet Σ we repeat the following: computing of $n + 1$-tuple φ_1 and checking if this tuple already exists in the set $Q^{\varepsilon c}_{\varphi}$, if not we add it there.

The following example shows the case of an automaton where it was not possible to calculate a finite crisp-deterministic fuzzy automaton equivalent to the given automaton using the algorithms proposed in [4, 5, 8], while using the Algorithm 1 proposed in this paper it is possible to calculate a crisp-deterministic fuzzy automaton ε-equivalent to the given automaton.

Example 4.1. Let $\mathcal{A}$ be a fuzzy automaton over the product structure and an alphabet $X = \{x\}$, where fuzzy sets of initial and final states, as well as the fuzzy transition function are given by the following vectors and matrices:

$$I = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}, \quad F = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \quad T_x = \begin{bmatrix} 0.1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix}.$$

In case when $\varepsilon = 0.000001$, using the procedure given in [18] we can compute the greatest ε-weak forward simulation φ on $\mathcal{A}$:

$$\varphi = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ \varepsilon & 0 & 1 & \varepsilon \\ 1 & 0 & 1 & 1 \end{bmatrix}.$$

Then, using Algorithm 1, we compute the states of an accessible crisp-deterministic fuzzy finite automaton

ε-equivalent to $\mathcal{A}$, i.e. the states of the automaton $\mathcal{A}_{\varphi}^{\varepsilon c} = (Q_{\varphi}^{\varepsilon c}, \Sigma, \varphi_e^{\varepsilon c}, T_{\varphi}^{\varepsilon c}, F_{\varphi}^{\varepsilon c})$. For every $n \in \{0, ..., 5\}$ we have:

$$\varphi_{x^n}^{\varepsilon c} = (\begin{bmatrix}(0.1)^{n+1} & (0.1)^n & \varepsilon & (0.1)^{n+1}\end{bmatrix} \quad , \quad (0.1)^n),$$

while, for every $n \in N$ such that $n > 5$ we have:

$$\varphi_{x^n}^{\varepsilon c} = (\begin{bmatrix}\varepsilon & \varepsilon & \varepsilon & \varepsilon\end{bmatrix} \quad , \quad \varepsilon).$$

Therefore, we can conclude that the automaton $\mathcal{A}_{\varphi}^{\varepsilon c}$ has exactly seven different states.

5. Conclusion and future work

This paper introduced a new approach to the determinization of fuzzy automata, with a key advantage: unlike traditional methods that may result in an infinite-state system, the proposed method always generates a finite crisp-deterministic fuzzy automaton. It is important to emphasize, however, that the resulting crisp-deterministic fuzzy automaton is not fully language-equivalent to the original fuzzy automaton. Instead, the automata agree on the words whose acceptance degrees exceed a predefined threshold parameter ε. As part of our future work, we plan to adapt the proposed determinization procedure so that it can be applied to weighted max-plus automata as well.

References

[1] R. Bělohlávek, *Determinism and fuzzy automata*, Inform. Sci. **143** (2002), 205–209.

[2] J. A. Brzozowski, *Canonical regular expressions and minimal state graphs for definite events*, in: Mathematical Theory of Automata, Polytechnic Press, Polytechnic Institute of Brooklyn, New York, 1962, 529–561.

[3] M. Droste, W. Kuich, H. Vogler (eds.), *Handbook of Weighted Automata*, Springer, Berlin, 2009.

[4] Z. Jančić, M. Ćirić, *Brzozowski type determinization for fuzzy automata*, Fuzzy Sets Syst. **249** (2014), 73–82.

[5] Z. Jančić, J. Ignjatović, M. Ćirić, *An improved algorithm for determinization of weighted and fuzzy automata*, Inform. Sci. **181** (2011), 1358–1368.

[6] Z. Jančić, I. Micić, M. Ćirić, J. R. G. de Mendívil, S. Stanimirović, *Approximate determinization of fuzzy automata over the product structure by means of approximate weak simulations*, Fuzzy Sets Syst., to appear (2025).

[7] Z. Jančić, I. Micić, J. Ignjatović, M. Ćirić, *Further improvement of determinization methods for fuzzy finite automata*, Fuzzy Sets Syst. **301** (2016), 79–102.

[8] J. Ignjatović, M. Ćirić, S. Bogdanović, *Determinization of fuzzy automata with membership values in complete residuated lattices*, Inform. Sci. **178** (2008), 164–180.

[9] J. Ignjatović, M. Ćirić, S. Bogdanović, T. Petković, *Myhill–Nerode type theory for fuzzy languages and automata*, Fuzzy Sets Syst. **161** (2010), 1288–1324.

[10] J. R. G. de Mendívil, J. R. Garitagoitia, *Determinization of fuzzy automata via factorization of fuzzy states*, Inform. Sci. **283** (2014), 165–179.

[11] J. R. Gonzalez de Mendívil, *Conditions for minimal fuzzy deterministic finite automata via Brzozowski's procedure*, IEEE Trans. Fuzzy Syst. **26** (2018), 2409–2420.

[12] J. R. Gonzalez de Mendívil, F. Fariña, *Canonization of max–min fuzzy automata*, Fuzzy Sets Syst. **376** (2019), 152–168.

[13] J. R. Gonzalez de Mendívil, *A generalization of Myhill–Nerode theorem for fuzzy languages*, Fuzzy Sets Syst. **301** (2016), 103–115.

[14] I. Micić, Z. Jančić, J. Ignjatović, M. Ćirić, *Determinization of fuzzy automata by means of the degrees of language inclusion*, IEEE Trans. Fuzzy Syst. **23** (2015), 2144–2153.

[15] A. G. de Mendívil Grau, S. Stanimirović, F. Fariña, *Minimal determinization algorithm for fuzzy automata*, IEEE Trans. Fuzzy Syst. **31** (2023), 3812–3822.

[16] A. G. de Mendívil Grau, S. Stanimirović, F. Fariña, *Reduction graph for minimal determinization of fuzzy automata*, in: Fuzzy Logic and Technology, and Aggregation Operators, Springer, Cham, 2023, 543–554.

[17] S. Stanimirović, M. Ćirić, J. Ignjatović, *Determinization of fuzzy automata by factorizations of fuzzy states and right invariant fuzzy quasi-orders*, Inform. Sci. **469** (2018), 79–100.

[18] I. Micić, M. Ćirić, J. Matejić, S. Stanimirović, L. A. Nguyen, *Approximate weak simulations and bisimulations for fuzzy automata over the product structure*, Fuzzy Sets Syst. **485** (2024), 108959.

[19] I. Micić, S. Stanimirović, J. R. Gonzalez de Mendívil, M. Ćirić, Z. Jančić, *Finite determinization of fuzzy automata using a parametric product-based t-norm*, Fuzzy Sets Syst. **488** (2024), 108990.

[20] R. Bělohlávek, *Fuzzy Relational Systems: Foundations and Principles*, Kluwer Academic/Plenum Publishers, New York, 2002.

[21] R. Bělohlávek, V. Vychodil, *Fuzzy Equational Logic*, Springer, Berlin, 2005.

[22] Y. Li, K. Qin, X. He, *Some new approaches to constructing similarity measures*, Fuzzy Sets Syst. **234** (2014), 46–60.

[23] Y. M. Li, W. Pedrycz, *Fuzzy finite automata and fuzzy regular expressions with membership values in lattice ordered monoids*, Fuzzy Sets Syst. **156** (2005), 68–92.
[24] D. Kirsten, L. Mäurer, *On the determinization of weighted automata*, J. Autom. Lang. Comb. **10** (2005), 287–312.
[25] C. Allauzen, M. Mohri, *Efficient algorithms for testing twins property*, J. Autom. Lang. Comb. **8** (2003), 117–144.
[26] M. Mohri, *Finite-state transducers in language and speech processing*, Comput. Linguist. **23** (1997), 269–311.
[27] G. Rahonis, *Fuzzy languages*, in: Handbook of Weighted Automata, Springer, Berlin, 2009, 481–517.