ARTICLE IN PRESS

FSS:6499

ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

Fuzzy Sets and Systems ••• (••••) •••–•••

FUZZY sets and systems

www.elsevier.com/locate/fss

Brzozowski type determinization for fuzzy automata ☆

Zorana Jančić, Miroslav Ćirić *

University of Niš, Faculty of Sciences and Mathematics, Višegradska 33, 18000 Niš, Serbia

Received 17 July 2013; received in revised form 9 February 2014; accepted 27 February 2014

Abstract

In this paper we adapt the well-known Brzozowski determinization method to fuzzy automata. This method gives better results than all previously known methods for determinization of fuzzy automata developed by Bělohlávek [4], Li and Pedrycz [20], Ignjatović et al. [15], and Jančić et al. [18]. Namely, as in the case of ordinary nondeterministic automata, Brzozowski type determinization of a fuzzy automaton results in a minimal crisp-deterministic fuzzy automaton equivalent to the starting fuzzy automaton, and we show that there are cases when all previous methods result in infinite automata, while Brzozowski type determinization results in a finite one. The paper deals with fuzzy automata over complete residuated lattices, but identical results can also be obtained in a more general context, for fuzzy automata over lattice-ordered monoids, and even for weighted automata over commutative semirings.

Keywords: Fuzzy automata; Fuzzy languages; Crisp-deterministic fuzzy automata; Determinization; Minimal automata; Nerode automaton; Complete residuated lattices

1. Introduction

The well-known Brzozowski's double reversal algorithm, presented for the first time in [10], is a concise and elegant algorithm having two purposes. When its input is a nondeterministic automaton, the algorithm alternates two reverse and determinization operations (more precisely, the accessible subset construction) and produces a minimal deterministic automaton equivalent to the starting automaton. In other words, the algorithm performs both determinization and minimization. On the other hand, when the input is a deterministic automaton, the algorithm performs its minimization applying just one reverse and determinization operation. Despite its worst-case exponential time complexity, the algorithm has recently gained popularity due to its excellent performance in practice, where it frequently outperforms theoretically faster algorithms (cf. [1,2,27,30]). For more information about Brzozowski's double reversal algorithm, and about algorithms for determinization of nondeterministic automata in general, we refer to [11,26,28,29].

☆ Research supported by Ministry of Education, Science and Technological Development, Republic of Serbia, Grant No. 174013.
* Corresponding author. Tel.: +38118224492; fax: +38118533014.
E-mail addresses: zoranajancic329@gmail.com (Z. Jančić), miroslav.ciric@pmf.edu.rs (M. Ćirić).

http://dx.doi.org/10.1016/j.fss.2014.02.021
0165-0114/© 2014 Published by Elsevier B.V.

The purpose of this paper is to adapt Brzozowski's double reversal algorithm to fuzzy automata. We start from an arbitrary fuzzy automaton and we show that applying twice the construction of a reverse Nerode automaton we obtain an equivalent minimal crisp-deterministic fuzzy automaton. We also demonstrate that this fuzzy version of Brzozowski's double reversal algorithm outperforms all previous methods for determinization of fuzzy automata developed by Bělohlávek [4], Li and Pedrycz [20], Ignjatović et al. [15], and Jančić et al. [18], in the sense that it not only produces a smaller automaton than all the above mentioned methods, but even when all these methods produce infinite automata, Brzozowski type determinization can produce a finite one. Moreover, when the starting fuzzy automaton is crisp-deterministic and accessible, its minimization is performed applying just one construction of a reverse Nerode automaton

The paper is organized as follows. In the preliminary section we recall basic notions and notation concerning fuzzy sets and relations, fuzzy automata and languages and crisp-deterministic fuzzy automata, we recall the concept of a Nerode automaton and introduce the concept of a reverse Nerode automaton. The main results are presented in Section 3. We first introduce the notion of a right language associated with a state of a fuzzy automaton and describe some basic properties of right languages. After that, we construct the right language automaton of a fuzzy automaton $\mathcal{A}$, we prove that it is isomorphic to the derivative automaton of the fuzzy language recognized by $\mathcal{A}$ (Theorem 3.3), and consequently, if all right fuzzy languages associated with states of an accessible crisp-deterministic fuzzy automaton $\mathcal{A}$ are pairwise different, we show that $\mathcal{A}$ is minimal. Then we prove that the reverse Nerode automaton of any accessible crisp-deterministic fuzzy automaton $\mathcal{A}$ is a minimal crisp-deterministic fuzzy automaton equivalent to the reverse automaton of $\mathcal{A}$ (Theorem 3.5), and further, we define the concept of a Brzozowski automaton of a fuzzy automaton $\mathcal{A}$ and prove that it is a minimal crisp-deterministic fuzzy automaton equivalent to $\mathcal{A}$ (Theorem 3.6). Finally, we give a simple example of a fuzzy automaton $\mathcal{A}$ for which all previously known determinization methods produce an infinite crisp-deterministic fuzzy automaton, while Brzozowski automaton of $\mathcal{A}$ is finite and has only three states.

The most popular structure of membership values that has recently been used in the theory of fuzzy sets, especially in the theory of fuzzy automata, are complete residuated lattices. For this reason, this paper also deals with fuzzy automata over complete residuated lattices. However, identical results can also be obtained in a more general context, for fuzzy automata over lattice-ordered monoids, and even for weighted automata over commutative semirings.

2. Preliminaries

2.1. Fuzzy sets and relations

In this work we will use complete residuated lattices as structures of membership values. A *residuated lattice* is an algebra $\mathcal{L} = (L, \wedge, \vee, \otimes, \to, 0, 1)$ such that

(L1) $(L, \wedge, \vee, 0, 1)$ is a lattice with the least element 0 and the greatest element 1,
(L2) $(L, \otimes, 1)$ is a commutative monoid with the unit 1,
(L3) $\otimes$ and $\to$ form an *adjoint pair*, i.e., they satisfy the *adjunction property*: for all $x, y, z \in L$,

$$x \otimes y \leqslant z \quad \Leftrightarrow \quad x \leqslant y \to z. \tag{1}$$

If, additionally, $(L, \wedge, \vee, 0, 1)$ is a complete lattice, then $\mathcal{L}$ is called a *complete residuated lattice*.

The operations $\otimes$ (called *multiplication*) and $\to$ (called *residuum*) are intended for modeling the conjunction and implication of the corresponding logical calculus, and supremum ($\bigvee$) and infimum ($\bigwedge$) are intended for modeling of the existential and general quantifier, respectively. For basic properties of complete residuated lattices we refer to [3,6].

The most studied and applied structures of truth values, defined on the real unit interval $[0, 1]$ with $x \wedge y = \min(x, y)$ and $x \vee y = \max(x, y)$, are the *Łukasiewicz structure* ($x \otimes y = \max(x + y - 1, 0)$, $x \to y = \min(1 - x + y, 1)$), the *Goguen* (*product*) *structure* ($x \otimes y = x \cdot y$, $x \to y = 1$ if $x \leqslant y$ and $= y/x$ otherwise) and the *Gödel structure* ($x \otimes y = \min(x, y)$, $x \to y = 1$ if $x \leqslant y$ and $= y$ otherwise). Another important set of truth values is the set $\{a_0, a_1, \dots, a_n\}$, $0 = a_0 < \cdots < a_n = 1$, with $a_k \otimes a_l = a_{\max(k+l-n,0)}$ and $a_k \to a_l = a_{\min(n-k+l,n)}$. A special case of the latter algebras is the two-element Boolean algebra of classical logic with the support $\{0, 1\}$. The only adjoint pair on the two-element Boolean algebra consists of the classical conjunction and implication operations. This structure of truth values we call the *Boolean structure*.

Please cite this article in press as: Z. Jančić, M. Ćirić, Brzozowski type determinization for fuzzy automata, Fuzzy Sets and Systems (2014), http://dx.doi.org/10.1016/j.fss.2014.02.021

In the sequel $\mathcal{L}$ will be a complete residuated lattice. A *fuzzy subset* of a set A *over* $\mathcal{L}$, or simply a *fuzzy subset* of A, is any function from A into L. Ordinary crisp subsets of A are considered as fuzzy subsets of A taking membership values in the set $\{0, 1\} \subseteq L$. Let f and g be two fuzzy subsets of A. The *equality* of f and g is defined as the usual equality of mappings, i.e., $f = g$ if and only if $f(x) = g(x)$, for every $x \in A$. The *inclusion* $f \leqslant g$ is also defined pointwise: $f \leqslant g$ if and only if $f(x) \leqslant g(x)$, for every $x \in A$. Endowed with this partial order the set L^A of all fuzzy subsets of A forms a complete residuated lattice, in which the meet (intersection) $\bigwedge_{i \in I} f_i$ and the join (union) $\bigvee_{i \in I} f_i$ of an arbitrary family $\{f_i\}_{i \in I}$ of fuzzy subsets of A are functions from A into L defined by

$$\Big(\bigwedge_{i \in I} f_i\Big)(x) = \bigwedge_{i \in I} f_i(x), \qquad \Big(\bigvee_{i \in I} f_i\Big)(x) = \bigvee_{i \in I} f_i(x),$$

for every $x \in A$, and $f \otimes g$ and $f \to g$ are defined by $f \otimes g(x) = f(x) \otimes g(x)$ and $f \to g(x) = f(x) \to g(x)$, for all $f, g \in L^A$ and $x \in A$.

A *fuzzy relation* between sets A and B (in this order) is any mapping from $A \times B$ into L, i.e., any fuzzy subset of $A \times B$, and the equality, inclusion (ordering), joins and meets of fuzzy relations are defined as for fuzzy sets. The set of all fuzzy relations between A and B will be denoted by $L^{A \times B}$. In particular, a fuzzy relation on a set A is any function from $A \times A$ into L, i.e., any fuzzy subset of $A \times A$. The *reverse* of a fuzzy relation $\varphi \in L^{A \times B}$ is a fuzzy relation $\overline{\varphi} \in L^{B \times A}$ defined by $\overline{\varphi}(b, a) = \varphi(a, b)$, for all $a \in A$ and $b \in B$. A crisp relation is a fuzzy relation which takes values only in the set $\{0, 1\}$, and if φ is a crisp relation of A to B, then expressions "$\varphi(a, b) = 1$" and "$(a, b) \in \varphi$" will have the same meaning.

For non-empty sets A, B and C, and fuzzy relations $\varphi \in L^{A \times B}$ and $\psi \in L^{B \times C}$, their *composition* is a fuzzy relation $\varphi \circ \psi \in L^{A \times C}$ defined by

$$(\varphi \circ \psi)(a, c) = \bigvee_{b \in B} \varphi(a, b) \otimes \psi(b, c), \tag{2}$$

for all $a \in A$ and $c \in C$. Moreover, for $f \in L^A$, $\varphi \in L^{A \times B}$ and $g \in L^B$, compositions $f \circ \varphi \in L^B$ and $\varphi \circ g \in L^A$ and the scalar product $f \circ g \in L$ are defined by

$$(f \circ \varphi)(b) = \bigvee_{a' \in A} f\big(a'\big) \otimes \varphi\big(a', b\big), \quad (\varphi \circ g)(a) = \bigvee_{b' \in B} \varphi\big(a, b'\big) \otimes g\big(b'\big), \quad f \circ g = \bigvee_{a \in A} f(a) \otimes g(a), \tag{3}$$

for all $a \in A$ and $b \in B$.

It is easy to check that $(\varphi_1 \circ \varphi_2) \circ \varphi_3 = \varphi_1 \circ (\varphi_2 \circ \varphi_3)$, $(f_1 \circ \varphi_1) \circ \varphi_2 = f_1 \circ (\varphi_1 \circ \varphi_2)$, $(f_1 \circ \varphi_1) \circ f_2 = f_1 \circ (\varphi_1 \circ f_2)$ and $(\varphi_1 \circ \varphi_2) \circ f_1 = \varphi_1 \circ (\varphi_2 \circ f_1)$, for all fuzzy relations φ_1, φ_2 and φ_3 and fuzzy sets f_1 and f_2 for which these compositions are defined, and consequently, all parentheses in these expressions can be omitted. Moreover, the composition of fuzzy relations is isotone in both arguments.

2.2. *Fuzzy automata*

In the further text, let $\mathcal{L} = (L, \wedge, \vee, \otimes, \to, 0, 1)$ be a complete residuated lattice and X a finite alphabet.

A *fuzzy automaton over* $\mathcal{L}$ and X, or simply a *fuzzy automaton*, is a quadruple $\mathcal{A} = (A, \delta, \sigma, \tau)$, where A is a non-empty set, called the *set of states*, $\delta : A \times X \times A \to L$ is a fuzzy subset of $A \times X \times A$, called the *fuzzy transition relation*, and $\sigma : A \to L$ and $\tau : A \to L$ are fuzzy subsets of A, called the *fuzzy set of initial states* and the *fuzzy set terminal states*, respectively. We can interpret $\delta(a, x, b)$ as the degree to which an input letter $x \in X$ causes a transition from a state $a \in A$ into a state $b \in A$, whereas we can interpret $\sigma(a)$ and $\tau(a)$ as the degrees to which a is respectively an input state and a terminal state. For methodological reasons we sometimes allow the set of states A to be infinite. A fuzzy automaton whose set of states is finite is called a *fuzzy finite automaton*.

Let X^* denote the free monoid over the alphabet X, and let $\varepsilon \in X^*$ be the empty word. The function δ can be extended up to a function $\delta^* : A \times X^* \times A \to L$ as follows: For $a, b \in A$ and the empty word ε we set

$$\delta^*(a, \varepsilon, b) = \begin{cases} 1, & \text{if } a = b, \\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

and for $a, b \in A$, $u \in X^*$ and $x \in X$ we set

Please cite this article in press as: Z. Jančić, M. Ćirić, Brzozowski type determinization for fuzzy automata, Fuzzy Sets and Systems (2014), http://dx.doi.org/10.1016/j.fss.2014.02.021

$$\delta^*(a,ux,b)=\bigvee_{c\in A}\delta^*(a,u,c)\otimes\delta(c,x,b). \tag{5}$$

For each $u\in X^*$ we define a fuzzy relation $\delta_u\in L^{A\times A}$ by $\delta_u(a,b)=\delta^*(a,u,b)$, for all $a,b\in A$. It is easy to check that $\delta_{uv}=\delta_u\circ\delta_v$, for all $u,v\in X^*$.

A *fuzzy language* in X^* over $\mathcal{L}$, or briefly a *fuzzy language*, is any fuzzy subset of the free monoid X^*. A *fuzzy language recognized by a fuzzy automaton* $\mathcal{A}=(A,\delta,\sigma,\tau)$ is a fuzzy language in $[\![\mathcal{A}]\!]\in L^{X^*}$ defined by

$$[\![\mathcal{A}]\!](u)=\bigvee_{a,b\in A}\sigma(a)\otimes\delta^*(a,u,b)\otimes\tau(b)=\sigma\circ\delta_u\circ\tau, \tag{6}$$

for any $u\in X^*$. In other words, the membership degree of the word u to $[\![\mathcal{A}]\!]$, i.e., the degree of recognition or acceptance of the word u, is equal to the degree to which u leads from some initial to some terminal state. Fuzzy automata $\mathcal{A}$ and $\mathcal{B}$ are called *language equivalent*, or shortly just *equivalent*, if they recognize the same fuzzy language, i.e., if $[\![\mathcal{A}]\!]=[\![\mathcal{B}]\!]$.

For more information on the recognizability of fuzzy languages we refer to [7–9], and for information on fuzzy automata over complete residuated lattices we refer to [13–17,22–25,31–33].

2.3. Crisp-deterministic fuzzy automata

A *crisp-deterministic fuzzy automaton* (for short: *cdfa*) over X and $\mathcal{L}$ is a quadruple $\mathcal{A}=(A,\delta,a_0,\tau)$, where A is a non-empty *set of states*, $\delta: A\times X\to A$ is a *transition function*, $a_0\in A$ is an *initial state* and $\tau: A\to L$ is a *fuzzy set of final states*. Equivalently, a crisp-deterministic fuzzy automaton can be considered as a fuzzy automaton $\mathcal{A}=(A,\delta,\sigma,\tau)$ whose fuzzy transition function δ and fuzzy set of initial states σ satisfy the following conditions: for all $x\in X$ and $a\in A$ there exists $a'\in A$ such that $\delta_x(a,a')=1$, and $\delta_x(a,b)=0$, for all $b\in A\setminus\{a'\}$, and $\sigma(a_0)=1$, and $\sigma(a)=0$ for every $a\in A\setminus\{a_0\}$. If the set of states A is finite, then $\mathcal{A}$ is called a *crisp-deterministic fuzzy finite automaton* (for short: *cdffa*).

For a crisp-deterministic fuzzy automaton $\mathcal{A}=(A,\delta,a_0,\tau)$, the transition function δ can be extended to a function $\delta^*: A\times X^*\to A$ by putting $\delta^*(a,\varepsilon)=a$, and $\delta^*(a,ux)=\delta(\delta^*(a,u),x)$, for all $a\in A$, $u\in X^*$ and $x\in X$. A state $a\in A$ is called *accessible* if there exists $u\in X^*$ such that $\delta^*(a_0,u)=a$. If every state of $\mathcal{A}$ is accessible, then $\mathcal{A}$ is called an *accessible crisp-deterministic fuzzy automaton*. The *fuzzy language* recognized by $\mathcal{A}$ is the fuzzy language $[\![\mathcal{A}]\!]\in L^{X^*}$ given by

$$[\![\mathcal{A}]\!](u)=\tau\big(\delta^*(a_0,u)\big), \tag{7}$$

for every $u\in X^*$. Obviously, the image of $[\![\mathcal{A}]\!]$ is contained in the image of τ, which is finite if the set of states is finite. A fuzzy language $f\in L^{X^*}$ is called *cdffa-recognizable* if there exists a crisp-deterministic fuzzy finite automaton $\mathcal{A}$ over X and $\mathcal{L}$ such that $[\![\mathcal{A}]\!]=f$.

Let $\mathcal{A}=(A,\delta,a_0,\tau)$ and $\mathcal{A}'=(A',\delta',a_0',\tau')$ be crisp-deterministic fuzzy automata. A function $\phi: A\to A'$ is called a *homomorphism* of $\mathcal{A}$ into $\mathcal{A}'$ if $\phi(a_0)=a_0'$, $\phi(\delta(a,x))=\delta'(\phi(a),x)$ and $\tau(a)=\tau'(\phi(a))$, for all $a\in A$ and $x\in X$. A bijective homomorphism is called an *isomorphism*. By $|\mathcal{A}|$ we denote the cardinality of the set of states of a fuzzy automaton $\mathcal{A}$. A crisp-deterministic fuzzy automaton $\mathcal{A}$ is called a *minimal crisp-deterministic fuzzy automaton* of a fuzzy language $f\in L^{X^*}$ if it recognizes f and $|\mathcal{A}|\leqslant|\mathcal{A}'|$, for any crisp-deterministic fuzzy automaton $\mathcal{A}'$ which recognizes f. Note that minimal crisp-deterministic fuzzy automata and minimization procedures that result in such automata were studied in [17,21].

For a fuzzy language $f\in L^{X^*}$ and $u\in X^*$, we define a fuzzy language $u^{-1}f\in L^{X^*}$ by $(u^{-1}f)(v)=f(uv)$, for each $v\in X^*$. We call $u^{-1}f$ the *left derivative* of f with respect to u. Let $A_f=\{u^{-1}f\mid u\in X^*\}$ denote the set of all left derivatives of f, and let $\delta_f: A_f\times X\to A_f$ and $\tau_f: A_f\to L$ be functions defined by

$$\delta_f(g,x)=x^{-1}g\quad\text{and}\quad\tau_f(g)=g(\varepsilon), \tag{8}$$

for all $g\in A_f$ and $x\in X$. Then $\mathcal{A}_f=(A_f,\delta_f,f,\tau_f)$ is an accessible crisp-deterministic fuzzy automaton, and it is called the *left derivative automaton*, or just the *derivative automaton*, of the fuzzy language f [14,17]. It was proved in [17] that the derivative automaton $\mathcal{A}_f$ is a minimal crisp-deterministic fuzzy automaton which recognizes f, and therefore, $\mathcal{A}_f$ is finite if and only if the fuzzy language f is cdffa-recognizable. An algorithm for construction of the derivative automaton of a fuzzy language, based on simultaneous construction of the derivative automata of ordinary languages $f^{-1}(a)$, for all $a \in \mathrm{Im}(f)$, was also given in [17].

2.4. Nerode and reverse Nerode automaton

Let $\mathcal{A} = (A, \delta, \sigma, \tau)$ be a fuzzy automaton over $\mathcal{L}$ and X. The *reverse fuzzy automaton* of $\mathcal{A}$ is a fuzzy automaton $\bar{\mathcal{A}} = (A, \bar{\delta}, \bar{\sigma}, \bar{\tau})$, where $\bar{\sigma} = \tau$, $\bar{\tau} = \sigma$, and $\bar{\delta} : A \times X \times A \to L$ is defined by:

$$\bar{\delta}(a, x, b) = \delta(b, x, a),$$

for all $a, b \in A$ and $x \in X$. Roughly speaking, the reverse automaton of $\mathcal{A}$ is obtained from $\mathcal{A}$ by exchanging fuzzy sets of initial and final states and "reversing" all the transitions.

Due to the fact that the multiplication $\otimes$ is commutative, we have that $\bar{\delta}_u(a, b) = \delta_{\bar{u}}(b, a)$, for all $a, b \in A$ and $u \in X^*$. For a fuzzy language $f \in L^{X^*}$, the *reverse fuzzy language* of f is a fuzzy language $\overline{f} \in L^{X^*}$ defined by $\overline{f}(u) = f(\bar{u})$, for each $u \in X^*$. As $\overline{(\bar{u})} = u$ for all $u \in X^*$, we have that $\overline{(\overline{f})} = f$, for each fuzzy language f.

If $\mathcal{A}$ is a fuzzy automaton over $\mathcal{L}$ and X, it is easy to see that the reverse fuzzy automaton $\bar{\mathcal{A}}$ recognizes the reverse fuzzy language $\overline{[\![\mathcal{A}]\!]}$ of the fuzzy language $[\![\mathcal{A}]\!]$ recognized by $\mathcal{A}$, i.e., $[\![\bar{\mathcal{A}}]\!] = \overline{[\![\mathcal{A}]\!]}$.

Let $\mathcal{A} = (A, \delta, \sigma, \tau)$ be a fuzzy automaton over X and $\mathcal{L}$. For each $u \in X^*$ we define fuzzy sets $\sigma_u, \tau_u \in L^A$ as follows:

$$\sigma_u(a) = \bigvee_{b \in A} \sigma(b) \otimes \delta^*(b, u, a), \qquad \tau_u(a) = \bigvee_{b \in A} \delta^*(b, u, a) \otimes \tau(b),$$

for each $a \in A$. Equivalently,

$$\sigma_u = \sigma \circ \delta_u, \qquad \tau_u = \delta_u \circ \tau.$$

The *Nerode automaton* of $\mathcal{A} = (A, \delta, \sigma, \tau)$ is a crisp-deterministic automaton $\mathcal{A}_N = (A_N, \delta_N, \sigma_\varepsilon, \tau_N)$ whose set of states is $A_N = \{\sigma_u \mid u \in X^*\}$, and functions $\delta_N : A_N \times X \longrightarrow A_N$ and $\tau_N : A_N \to L$ are defined by

$$\delta_N(\sigma_u, x) = \sigma_{ux}, \qquad \tau_N(\sigma_u) = \sigma_u \circ \tau, \tag{9}$$

for all $u \in X^*$ and $x \in X$. The concept of the Nerode automaton of a fuzzy automaton was first introduced by Ignjatović et al. in [15,17], for fuzzy automata over a complete residuated lattice, but it was also pointed out that the same construction can be extended to fuzzy automata over a lattice-ordered monoid, weighted automata over a semiring, and even to weighted automata over a strong bimonoid (cf. [12,18]). In [15] it was also shown that the Nerode automaton of a fuzzy automaton $\mathcal{A}$ over a complete residuated lattice is a crisp-deterministic fuzzy automaton equivalent to $\mathcal{A}$, i.e., $[\![\mathcal{A}_N]\!] = [\![\mathcal{A}]\!]$.

By the *reverse Nerode automaton* of $\mathcal{A}$ we will mean the Nerode automaton of the reverse fuzzy automaton $\bar{\mathcal{A}}$ of $\mathcal{A}$. For the sake of simplicity, we denote the reverse Nerode automaton of $\mathcal{A}$ by $\mathcal{A}_{\overline{N}}$ (instead of $(\bar{\mathcal{A}})_N$). Let us note that $\mathcal{A}_{\overline{N}} = (A_{\overline{N}}, \delta_{\overline{N}}, \tau_\varepsilon, \tau_{\overline{N}})$, where $A_{\overline{N}} = \{\tau_u \mid u \in X^*\}$, and the functions $\delta_{\overline{N}} : A_{\overline{N}} \times X \to A_{\overline{N}}$ and $\tau_{\overline{N}} : A_{\overline{N}} \to L$ are given by

$$\delta_{\overline{N}}(\tau_u, x) = \tau_{xu}, \qquad \tau_{\overline{N}}(\tau_u) = \tau_u \circ \sigma, \tag{10}$$

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

3. The main results

Let $\mathcal{A} = (A, \delta, \sigma, \tau)$ be a fuzzy automaton over X and $\mathcal{L}$.

For any state $a \in A$, the *right fuzzy language associated with a* is the fuzzy language $\tau_a \in L^{X^*}$ defined by

$$\tau_a(u) = \bigvee_{b \in A} \delta^*(a, u, b) \otimes \tau(b),$$

for each $u \in X^*$. In other words, τ_a is the fuzzy language recognized by a fuzzy automaton $\mathcal{A}' = (A, \delta, a, \tau)$ obtained from $\mathcal{A}$ by replacing σ with the single crisp initial state a. The *left fuzzy language associated with a* is the fuzzy language $\sigma_a \in L^{X^*}$ given by

$$\sigma_a(u) = \bigvee_{b\in A} \sigma(b) \otimes \delta^*(b, u, a),$$

for each $u \in X^*$, i.e., the fuzzy language recognized by a fuzzy automaton $\mathcal{A}' = (A, \delta, \sigma, \{a\})$ obtained from $\mathcal{A}$ by replacing τ with the single crisp terminal state a.

We can easily show that the following is true.

Lemma 3.1. *The right fuzzy language associated with the state a of a fuzzy automaton $\mathcal{A}$ is equal to the reverse of the left fuzzy language associated with the state a in the reverse fuzzy automaton $\overline{\mathcal{A}}$.*

For a crisp-deterministic fuzzy automaton $\mathcal{A} = (A, \delta, a_0, \tau)$, the right fuzzy language associated with a state a in $\mathcal{A}$ is given by

$$\tau_a(u) = \tau\big(\delta^*(a, u)\big), \tag{11}$$

for each $u \in X^*$, and in particular, $\tau_{a_0} = [\![\mathcal{A}]\!]$, i.e., the right fuzzy language associated with the initial state a_0 is the fuzzy language recognized by $\mathcal{A}$. It can be also easily verified that the following is true.

Lemma 3.2. *Let $\mathcal{A} = (A, \delta, a_0, \tau)$ be a crisp-deterministic fuzzy automaton. Then*

$$\tau_{\delta^*(a,u)} = u^{-1}\tau_a, \tag{12}$$

for all $a \in A$ and $u \in X^*$.

If $\mathcal{A} = (A, \delta, a_0, \tau)$ is a crisp-deterministic fuzzy automaton, we define another crisp-deterministic fuzzy automaton $\mathcal{A}_r = (A_r, \delta_r, \tau_{a_0}, \tau_r)$ as follows: the set of states A_r is the set of all right fuzzy languages associated with states of $\mathcal{A}$, and $\delta_r : A_r \times X \to A_r$ and $\tau_r : A_r \to L$ are given by:

$$\delta_r(\tau_a, x) = \tau_{\delta(a,x)}, \qquad \tau_r(\tau_a) = \tau_a(\varepsilon),$$

for each $\tau_a \in A_r$. We have the following:

Theorem 3.3. *Let $\mathcal{A} = (A, \delta, a_0, \tau)$ be an accessible crisp-deterministic fuzzy automaton. Then $\mathcal{A}_r$ is an accessible crisp-deterministic fuzzy automaton isomorphic to the derivative automaton $\mathcal{A}_f$ of the fuzzy language $f = [\![\mathcal{A}]\!]$.*

Proof. Define a mapping $\phi : A_f \to A_r$ by

$$\phi\big(u^{-1}f\big) = \tau_{\delta^*(a_0,u)},$$

for each $u \in X^*$. If $u, v \in X^*$ such that $u^{-1}f = v^{-1}f$, then according to (12) we obtain that $\tau_{\delta^*(a_0,u)} = \tau_{\delta^*(a_0,v)}$, and hence $\phi(u^{-1}f) = \phi(v^{-1}f)$. Thus, ϕ is well-defined. On the other hand, let $u, v \in X^*$ such that $\phi(u^{-1}f) = \phi(v^{-1}f)$, i.e., $\tau_{\delta^*(a_0,u)} = \tau_{\delta^*(a_0,v)}$. Then by (12) it follows that

$$u^{-1}f = u^{-1}\tau_{a_0} = \tau_{\delta^*(a_0,u)} = \tau_{\delta^*(a_0,v)} = v^{-1}\tau_{a_0} = v^{-1}f.$$

Therefore, ϕ is injective. Due to the fact that $\mathcal{A}$ is accessible, it is easy to show that ϕ is a surjective mapping.

In order to prove that ϕ is a homomorphism, consider arbitrary $u \in X^*$ and $x \in X$. Then

$$\phi\big(\delta_f\big(u^{-1}f, x\big)\big) = \phi\big((ux)^{-1}f\big) = \tau_{\delta^*(a_0,ux)} = \tau_{\delta(\delta^*(a_0,u),x)} = \delta_r(\tau_{\delta^*(a_0,u)}, x) = \delta_r\big(\phi\big(u^{-1}f\big), x\big).$$

Moreover, $\phi(\varepsilon^{-1}f) = \tau_{a_0}$ and $\tau_f(u^{-1}f) = \tau_r(\phi(u^{-1}f))$. Hence, ϕ is an isomorphism. $\square$

The automaton $\mathcal{A}_r$ will be called the *right language automaton* of $\mathcal{A}$.

By the previous theorem we obtain the following consequence.

Corollary 3.4. *Let $\mathcal{A} = (A, \delta, a_0, \tau)$ be an accessible crisp-deterministic fuzzy automaton. If all right fuzzy languages associated with states of $\mathcal{A}$ are pairwise different, then $\mathcal{A}$ is minimal.*

Please cite this article in press as: Z. Jančić, M. Ćirić, Brzozowski type determinization for fuzzy automata, Fuzzy Sets and Systems (2014), http://dx.doi.org/10.1016/j.fss.2014.02.021

Proof. It is clear that the function $\phi : A \to A_r$ defined by $\phi(a) = \tau_a$ is a homomorphism of $\mathcal{A}$ onto $\mathcal{A}_r$. Therefore, if all right fuzzy languages associated with states of $\mathcal{A}$ are pairwise different, then ϕ is an isomorphism of $\mathcal{A}$ onto $\mathcal{A}_r$, and according to Theorem 3.3, $\mathcal{A}$ is minimal. □

Let us note that if $\mathcal{A} = (A, \delta, a_0, \tau)$ is a crisp-deterministic fuzzy automaton, then its reverse Nerode automaton is $\mathcal{A}_{\overline{N}} = (A_{\overline{N}}, \delta_{\overline{N}}, \tau_\varepsilon, \tau_{\overline{N}})$, where $A_{\overline{N}}$ and $\delta_{\overline{N}} : A_{\overline{N}} \times X \to A_{\overline{N}}$ have the same form as in the general case, whereas the function $\tau_{\overline{N}} : A_{\overline{N}} \to L$ is given by

$$\tau_{\overline{N}}(\tau_u) = \tau_u(a_0), \tag{13}$$

for each $u \in X^*$.

Now, we are ready to prove the following

Theorem 3.5. *For any accessible crisp-deterministic fuzzy automaton $\mathcal{A} = (A, \delta, a_0, \tau)$, the reverse Nerode automaton $\mathcal{A}_{\overline{N}}$ is a minimal crisp-deterministic fuzzy automaton equivalent to $\overline{\mathcal{A}}$.*

Proof. As we have already noted, $\mathcal{A}_{\overline{N}}$ is a crisp-deterministic fuzzy automaton equivalent to $\overline{\mathcal{A}}$, so it remains to show that it is minimal. According to Corollary 3.4, it is enough to prove that all right fuzzy languages associated with states of $\mathcal{A}_{\overline{N}}$ are pairwise different.

Let $\tau_u, \tau_v \in A_{\overline{N}}$, where $u, v \in X^*$, be two different states of $\mathcal{A}_{\overline{N}}$. Then there is $a \in A$ such that $\tau_u(a) \neq \tau_v(a)$, and since $\mathcal{A}$ is accessible, there is $w \in X^*$ such that $a = \delta^*(a_0, w)$. According to (11) we obtain that

$$\begin{aligned}\tau_{\tau_u}(\overline{w}) &= \tau_{\overline{N}}\big(\delta^*_{\overline{N}}(\tau_u, \overline{w})\big) = \tau_{\overline{N}}(\tau_{wu}) = \tau_{wu}(a_0) = \tau_{a_0}(wu) = \tau\big(\delta^*(a_0, wu)\big) \\ &= \tau\big(\delta\big(\delta^*(a_0, w), u\big)\big) = \tau\big(\delta^*(a, u)\big) = \tau_u(a),\end{aligned}$$

whence $\tau_{\tau_u}(\overline{w}) = \tau_u(a)$, and in the same way we show that $\tau_{\tau_v}(\overline{w}) = \tau_v(a)$. Since $\tau_u(a) \neq \tau_v(a)$, we conclude that $\tau_{\tau_u}(\overline{w}) \neq \tau_{\tau_v}(\overline{w})$, and hence, τ_{τ_u} and τ_{τ_v} are different right fuzzy languages associated with states of $\mathcal{A}_{\overline{N}}$. □

Let $\mathcal{A}$ be a fuzzy automaton over an alphabet X and a complete residuated lattice $\mathcal{L}$. The *Brzozowski automaton* of $\mathcal{A}$, in notation $\mathcal{A}_B$, is a fuzzy automaton obtained from $\mathcal{A}$ applying twice the construction of the reverse Nerode automaton, i.e.,

$$\mathcal{A}_B = (\mathcal{A}_{\overline{N}})_{\overline{N}} = \big(\overline{(\overline{\mathcal{A}})_N}\big)_N.$$

Now we are ready to state and prove the main result of this paper.

Theorem 3.6. *Let $\mathcal{A}$ be a fuzzy automaton over an alphabet X and a complete residuated lattice $\mathcal{L}$. The Brzozowski automaton $\mathcal{A}_B$ is a minimal crisp-deterministic fuzzy automaton equivalent to $\mathcal{A}$.*

Proof. This is an immediate consequence of Theorem 4.1 of [15] and Theorem 3.5.

Namely, according to Theorem 4.1 of [15], the reverse Nerode automaton $\mathcal{A}_{\overline{N}} = (\overline{\mathcal{A}})_N$ is a crisp-deterministic fuzzy automaton equivalent to $\overline{\mathcal{A}}$, and its reverse Nerode automaton $(\mathcal{A}_{\overline{N}})_{\overline{N}}$ is a crisp-deterministic fuzzy automaton equivalent to $\mathcal{A}$. Therefore, by Theorem 3.5, $\mathcal{A}_B$ is a minimal crisp-deterministic fuzzy automaton equivalent to $\mathcal{A}$. □

Let $\mathcal{A} = (A, \delta, \sigma, \tau)$ be a fuzzy finite automaton with n states and m input letters, and suppose that the subsemiring $\mathcal{L}^*(\delta, \sigma, \tau)$ of the semiring $(L, \vee, \otimes, 0, 1)$, generated by all membership values taken by δ, σ and τ, is finite and has k elements. An algorithm which constructs the Nerode automaton of $\mathcal{A}$ was provided in [15], and it can easily be transformed to an algorithm which constructs the reverse Nerode automaton of $\mathcal{A}$. Any of these algorithms builds the *transition tree* of the crisp-deterministic fuzzy automaton that it constructs (Nerode or reverse Nerode), and it is an m-ary tree with at most k^n internal vertices which correspond to the states of the automaton under construction. Computationally most demanding part of the algorithm is the one where for any newly-constructed fuzzy set (a vertex of the tree) the algorithm checks whether it has already been computed before. The computational time of this part, and the whole algorithm, is $O(mnk^{2n})$ (cf. [19]). Therefore, the first round of the application of the double reversal procedure to $\mathcal{A}$ produces an automaton with at most k^n states, and the computational time of this round is $O(mnk^{2n})$.

Please cite this article in press as: Z. Jančić, M. Ćirić, Brzozowski type determinization for fuzzy automata, Fuzzy Sets and Systems (2014), http://dx.doi.org/10.1016/j.fss.2014.02.021

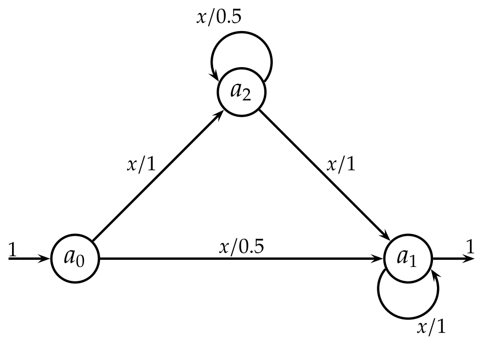

Fig. 1. The transition graph of the fuzzy automaton $\mathcal{A}$.

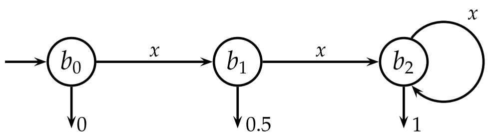

Fig. 2. The transition graph of the reverse Nerode automaton $\mathcal{A}_{\overline{N}}$ and the Brzozowski automaton $\mathcal{A}_B$ of $\mathcal{A}$.

The second round may start from an exponentially larger automaton, but despite that, this round produces a minimal crisp-deterministic fuzzy automaton equivalent to $\mathcal{A}$, an automaton that is not greater than the Nerode automaton of $\mathcal{A}$, which cannot have more than k^n states. Thus, the resulting transition tree cannot have more than k^n internal vertices, and consequently, the second round has the same computational time $O(mnk^{2n})$. This means that the total computational time of the Brzozowski's double reversal algorithm for the fuzzy automaton $\mathcal{A}$ is $O(mnk^{2n})$, the same as for constructions of the Nerode and the reverse Nerode automaton of $\mathcal{A}$.

Finally, we give the following example.

Example 3.7. Let $\mathcal{A} = (A, \delta, \sigma, \tau)$ be a fuzzy finite automaton over the alphabet $X = \{x\}$ and the Goguen (product) structure, given by the transition graph shown in Fig. 1.

In matrix form, σ, δ_x and τ are represented as follows:

$$\sigma = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \qquad \delta_x = \begin{bmatrix} 0 & 0.5 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0.5 \end{bmatrix}, \qquad \tau = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$$

It is easy to verify that $\sigma_x = \begin{bmatrix} 0 & 0.5 & 1 \end{bmatrix}$ and

$$\sigma_{x^n} = \begin{bmatrix} 0 & 1 & 0.5^{n-1} \end{bmatrix},$$

for each $n \in \mathbb{N}$, $n \geqslant 2$, which means that the Nerode automaton of $\mathcal{A}$ has infinitely many states.

On the other hand, it is not hard to check that the reverse Nerode automaton $\mathcal{A}_{\overline{N}}$ and the Brzozowski automaton $\mathcal{A}_B$ are mutually isomorphic, and they are represented by the graph in Fig. 2.

This example demonstrates that there is a fuzzy finite automaton $\mathcal{A}$ whose Nerode automaton $\mathcal{A}_N$ is infinite, but Brzozowski automaton $\mathcal{A}_B$ is finite. Note that the determinization methods developed by Bělohlávek [4] and Li and Pedrycz [20] always result in automata whose cardinality is greater than or equal to the cardinality of the related Nerode automaton (cf. [15]), and therefore, in this case these methods also give infinite automata. On the other hand, the method developed by Jančić et al. [18] always results in an automaton whose cardinality is less than or equal to the cardinality of the related Nerode automaton, but according to Theorem 3.7 of [18], this automaton is finite if and only if the related Nerode automaton is finite. Hence, in this case the method from [18] also gives an infinite automaton. Summing up, we conclude that all the above mentioned methods applied to the fuzzy finite automaton $\mathcal{A}$ from this example produce infinite automata, but Brzozowski automaton $\mathcal{A}_B$ is finite.

4. Concluding remarks

Brzozowski's double reversal algorithm is a well-known determinization–minimization algorithm which, despite its worst-case exponential time complexity, has excellent performance in practice and often outperforms theoretically faster algorithms. Here we have developed a Brzozowski type algorithm for fuzzy automata. We have shown that this algorithm outperforms all previously known methods for determinization of fuzzy automata, in the sense that it not only produces a smaller automaton than all previous methods, but even when all these methods produce infinite automata, Brzozowski type determinization can produce a finite one. No matter that Brzozowski type algorithm has been developed here for fuzzy automata over complete residuated lattice, without any modifications it can be also applied to fuzzy automata over lattice-ordered monoids and weighted automata over commutative semirings.

In our future research, we will search for determinization methods that produce automata having even smaller number of states than the Nerode automaton or the reverse Nerode automaton. Such an improvement of the construction of the reverse Nerode automaton could also significantly improve the performance of the double reversal algorithm for fuzzy automata. In addition, it could be interesting to exploit the concept of an approximate equivalence of fuzzy automata, introduced in [5], and study approximate determinization of a fuzzy automaton, a procedure of constructing a crisp-deterministic fuzzy automaton whose fuzzy language is approximately equal to the fuzzy language of the given one.

References

[1] M. Almeida, N. Moreira, R. Reis, On the performance of automata minimization algorithms, Technical Report DCC-2007-03, DCC-FC & LIACC, Universidade do Porto, 2007.
[2] M. Almeida, N. Moreira, R. Reis, Finite automata minimization, in: J. Wang (Ed.), Handbook of Finite State Based Models and Applications, CRC Press/Taylor & Francis Group, 2013, pp. 145–170.
[3] R. Bělohlávek, Fuzzy Relational Systems: Foundations and Principles, Kluwer, New York, 2002.
[4] R. Bělohlávek, Determinism and fuzzy automata, Inf. Sci. 143 (2002) 205–209.
[5] R. Bělohlávek, M. Krupka, Approximate minimization of fuzzy automata, J. Mult.-Valued Log. Soft Comput. 15 (2009) 125–135.
[6] R. Bělohlávek, V. Vychodil, Fuzzy Equational Logic, Studies in Fuzziness and Soft Computing, Springer, Berlin–Heidelberg, 2005.
[7] S. Bozapalidis, O. Louscou-Bozapalidou, On the recognizability of fuzzy languages I, Fuzzy Sets Syst. 157 (2006) 2394–2402.
[8] S. Bozapalidis, O. Louscou-Bozapalidou, On the recognizability of fuzzy languages II, Fuzzy Sets Syst. 159 (2008) 107–113.
[9] S. Bozapalidis, O. Louscou-Bozapalidou, Fuzzy tree language recognizability, Fuzzy Sets Syst. 161 (2010) 716–734.
[10] J.A. Brzozowski, Canonical regular expression and minimal state graphs for definite events, in: Mathematical Theory of Automata, in: MRI Symposia Series, vol. 12, Polytechnic Press, Polytechnic Institute of Brooklyn, NY, 1962, pp. 529–561.
[11] J.-M. Champarnaud, A. Khorsi, T. Paranthoën, Split and join for minimizing: Brzozowski's algorithm, in: M. Balík, M. Simánek (Eds.), Proceedings of the Prague Stringology Conference, 2002, pp. 96–104.
[12] M. Ćirić, M. Droste, J. Ignjatović, H. Vogler, Determinization of weighted finite automata over strong bimonoids, Inf. Sci. 180 (2010) 3497–3520.
[13] M. Ćirić, A. Stamenković, J. Ignjatović, T. Petković, Fuzzy relation equations and reduction of fuzzy automata, J. Comput. Syst. Sci. 76 (2010) 609–633.
[14] J. Ignjatović, M. Ćirić, Formal power series and regular operations on fuzzy languages, Inf. Sci. 180 (2010) 1104–1120.
[15] J. Ignjatović, M. Ćirić, S. Bogdanović, Determinization of fuzzy automata with membership values in complete residuated lattices, Inf. Sci. 178 (2008) 164–180.
[16] J. Ignjatović, M. Ćirić, S. Bogdanović, Fuzzy homomorphisms of algebras, Fuzzy Sets Syst. 160 (2009) 2345–2365.
[17] J. Ignjatović, M. Ćirić, S. Bogdanović, T. Petković, Myhill–Nerode type theory for fuzzy languages and automata, Fuzzy Sets Syst. 161 (2010) 1288–1324.
[18] Z. Jančić, J. Ignjatović, M. Ćirić, An improved algorithm for determinization of weighted and fuzzy automata, Inf. Sci. 181 (2011) 1358–1368.
[19] Z. Jančić, I. Micić, J. Ignjatović, M. Ćirić, Two-in-one: determinization and state reduction of fuzzy automata, Inf. Sci (2014), submitted for publication.
[20] 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.
[21] Y.M. Li, W. Pedrycz, Minimization of lattice finite automata and its application to the decomposition of lattice languages, Fuzzy Sets Syst. 158 (2007) 1423–1436.
[22] A. Stamenković, M. Ćirić, J. Ignjatović, Reduction of fuzzy automata by means of fuzzy quasi-orders, Inf. Sci. (2014), http://dx.doi.org/10.1016/j.ins.2014.02.028.
[23] D.W. Qiu, Automata theory based on completed residuated lattice-valued logic (I), Sci. China, Ser. F 44 (6) (2001) 419–429.
[24] D.W. Qiu, Automata theory based on completed residuated lattice-valued logic (II), Sci. China, Ser. F 45 (6) (2002) 442–452.
[25] D.W. Qiu, Pumping lemma in automata theory based on complete residuated lattice-valued logic: A note, Fuzzy Sets Syst. 157 (2006) 2128–2138.

Please cite this article in press as: Z. Jančić, M. Ćirić, Brzozowski type determinization for fuzzy automata, Fuzzy Sets and Systems (2014), http://dx.doi.org/10.1016/j.fss.2014.02.021

[26] J. Sakarovitch, Elements of Automata Theory, Cambridge University Press, 2009.

[27] D. Tabakov, M.Y. Vardi, Experimental evaluation of classical automata constructions, in: G. Sutcliffe, A. Voronkov (Eds.), LPAR 2005, in: Lecture Notes in Artificial Intelligence, vol. 3835, 2005, pp. 396–411.

[28] R. van Glabbeek, B. Ploeger, Five determinization algorithms, in: O.H. Ibarra, B. Ravikumar (Eds.), CIAA 2008, in: Lecture Notes in Computer Science, vol. 5148, 2008, pp. 161–170.

[29] R. van Glabbeek, B. Ploeger, Five determinization algorithms, CS-Report 08-14, Eindhoven University of Technology, 2008.

[30] B.W. Watson, Taxonomies and toolkit of regular languages algorithms, PhD thesis, Eindhoven University of Technology, 1995.

[31] W. Wechler, The Concept of Fuzziness in Automata and Language Theory, Akademie-Verlag, Berlin, 1978.

[32] L.H. Wu, D.W. Qiu, Automata theory based on complete residuated lattice-valued logic: Reduction and minimization, Fuzzy Sets Syst. 161 (2010) 1635–1656.

[33] H. Xing, D.W. Qiu, F.C. Liu, Automata theory based on complete residuated lattice-valued logic: Pushdown automata, Fuzzy Sets Syst. 160 (2009) 1125–1140.

Please cite this article in press as: Z. Jančić, M. Ćirić, Brzozowski type determinization for fuzzy automata, Fuzzy Sets and Systems (2014), http://dx.doi.org/10.1016/j.fss.2014.02.021