HOMEWORK
Transform the academic article into the lecture class

Lecture (Spoken)

Today we are going to delve into the techniques of a powerful tool, which will help us to study the equivalences of fuzzy automata to a greater degree. We have already introduced the notions of bisimulations and uniform fuzzy relations. Now, we will learn to utilise both of them in conjunction! We will prove that a uniform fuzzy relation between fuzzy automata $\mathscr{A}$ and $\mathscr{B}$ is a forward bisimulation if and only if its kernel and co-kernel are forward bisimulation fuzzy equivalence relations on $\mathscr{A}$ and $\mathscr{B}$. Therefore, we can deduce that special isomorphism between factor fuzzy automata with respect to these fuzzy equivalence relations exists. Then we will turn our attention to solving another important problem which reduces the problem of testing UFB-equivalence to the problem of testing isomorphism of fuzzy automata, and it is closely related to the well-known graph isomorphism problem. We will use the obtained results to solve similar problems for backward-forward bisimulations what will allow us to see the differences clearly. I want to point out that backward and forward-backward bisimulations can not be considered separately, because these concepts are essentially dual.

Journal article (written)

INTRODUCTION

Bisimulations have been widely used in many areas of computer science to model equivalence between various systems, and to reduce the number of states of these systems, whereas uniform fuzzy relations have recently been introduced as a means to model the fuzzy equivalence between elements of two possible different sets. Here we use the conjunction of these two concepts as a powerful tool in the study of equivalence between fuzzy automata. We prove that a uniform fuzzy relation between fuzzy automata $\mathscr{A}$ and $\mathscr{B}$ is a forward bisimulation if and only if its kernel and co-kernel are forward bisimulation fuzzy equivalence relations on $\mathscr{A}$ and $\mathscr{B}$ and there is a special isomorphism between factor fuzzy automata with respect to these fuzzy equivalence relations. As a consequence we get that fuzzy automata $\mathscr{A}$ and $\mathscr{B}$ are UFB-equivalent, i.e., there is a uniform forward bisimulation between them, if and only if there is a special isomorphism between the factor fuzzy automata of $\mathscr{A}$ and $\mathscr{B}$ with respect to their greatest forward bisimulation fuzzy equivalence relations. This result reduces the problem of testing UFB-equivalence to the problem of testing isomorphism of fuzzy automata, which is closely related to the well-known graph isomorphism problem. We prove some similar results for backward-forward bisimulations, and we point to fundamental differences. Because of the duality with the studied concepts, backward and forward-backward bisimulations are not considered separately.

Lecture (spoken)

During the previous class, we introduced fuzzy automata with membership values in complete residuated lattices. We will be staying at the same structure during this class, as well. Let's review some concepts we talked about last time. As you remember, a fuzzy automaton over complete residuated lattice $\mathscr{L}$ is an algebraic structure $\mathscr{A} = (A, \delta^A, \sigma^A, \tau^A)$, where A is a set of states, $\delta^A : A \times X \times A \to L$ is a fuzzy transition function, and $\sigma^A : A \to L$ and $\tau^A : A \to L$ are a fuzzy set of initial states and a fuzzy set terminal states, respectively. In the general case, the set of states of automata can be infinite, but only finite abstract machines have practical significance, so without losing generality, we can assume that the set A is finite. If we get back to what we were discussing earlier, an isomorphism between two fuzzy automata $\mathscr{A} = (A, \delta^A, \sigma^A, \tau^A)$ and $\mathscr{B} = (B, \delta^B, \sigma^B, \tau^B)$ is a bijective mapping $\varphi : A \longrightarrow B$ consistent to the fuzzy transition functions of A and B. Let me remaind you that it means that for all $a, a_1, a_2 \in A$ and $x \in X$, the following is true:

$$\delta_x^A(a_1, a_2) = \delta_x^B(\varphi(a_1), \varphi(a_2)),$$
$$\sigma^A(a) = \sigma^B(\varphi(a)),$$
$$\tau^A(a) = \tau^B(\varphi(a)).$$

PRELIMINARIES

In this paper we study fuzzy automata with membership values in complete residuated lattices.

In the rest of the paper, if not noted otherwise, let $\mathscr{L}$ be a complete residuated lattice and let X be an (finite) alphabet. A *fuzzy automaton over* $\mathscr{L}$ and X, or simply a *fuzzy automaton*, is a quadruple $\mathscr{A} = (A, \delta^A, \sigma^A, \tau^A)$, where A is a non-empty set, called the *set of states*, $\delta^A : A \times X \times A \to L$ is a fuzzy subset of $A \times X \times A$, called the *fuzzy transition function*, and $\sigma^A : A \to L$ and $\tau^A : A \to L$ are the fuzzy subsets of A, called the *fuzzy set of initial states* and the *fuzzy set terminal states*, respectively. We can interpret $\delta^A(a, x, b)$ 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 $\sigma^A(a)$ and $\tau^A(a)$ 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 $\mathscr{A} = (A, \delta^A, \sigma^A, \tau^A)$ and $\mathscr{B} = (B, \delta^B, \sigma^B, \tau^B)$ be fuzzy automata. A function $\varphi : A \longrightarrow B$ is called an *isomorphism* between $\mathscr{A}$ and $\mathscr{B}$ if it is bijective and for all $a, a_1, a_2 \in A$ and $x \in X$, the following is true:

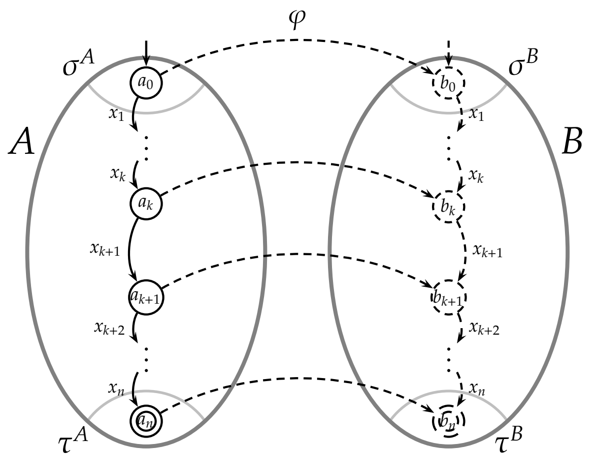
$$\delta_x^A(a_1, a_2) = \delta_x^B(\varphi(a_1), \varphi(a_2)), \quad (1)$$
$$\sigma^A(a) = \sigma^B(\varphi(a)), \quad (2)$$
$$\tau^A(a) = \tau^B(\varphi(a)). \quad (3)$$

Lecture (spoken)

Now I want to draw your attention to bisimulations for fuzzy automata.I would like to give you definitions and discuss the basic properties of bisimulations.

In my opininon, you might be more familiar with nondeterministic (Boolean) automata then with fuzzy automata. Hence, I will explain the meaning of forward and backward simulations between the nondeterministic automata $\mathscr{A}$ and $\mathscr{B}$. I'd like you to look at the following diagram. We will suppose that φ is a forward simulation between $\mathscr{A}$ and $\mathscr{B}$ and let $a_0, a_1, \ldots, a_n$ be an arbitrary successful run of the automaton $\mathscr{A}$ on a word $u = x_1x_2\cdots x_n$ ($x_1, x_2, \ldots, x_n \in X$), i.e., a sequence of states of $\mathscr{A}$ such that $a_0 \in \sigma^A$, $(a_k, a_{k+1}) \in \delta^A_{x_{k+1}}$, for $0 \leqslant k \leqslant n-1$, and $a_n \in \tau^A$. Let us note that there is an initial state $b_0 \in \sigma^B$ such that $(a_0, b_0) \in \varphi$. Obviously, if we built the successful path $(b_{i-1}, b_i) \in \delta^B_{x_i}$ in $\mathscr{B}$, for each i, $1 \leq i \leq k$, for $0 \leq k \leq n-1$, such that $(a_i, b_i) \in \varphi$, we can conclude that. Therefore, we obtain that $(b_k, b_{k+1}) \in \delta^B_{x_{k+1}}$ and $(a_{k+1}, b_{k+1}) \in \varphi$. It allows us to built a sequence $b_0, b_1, \ldots, b_n$ of states of $\mathscr{B}$ such that $b_0 \in \sigma^B$, $(b_k, b_{k+1}) \in \delta^B_{x_{k+1}}$ and $b_n \in \tau^B$.

I would like to emphasize that $b_0, b_1, \ldots, b_n$ is a successful run of the automaton $\mathscr{B}$ on the word u which simulates the original run $a_0, a_1, \ldots, a_n$ of $\mathscr{A}$ on u. In a similar way we can understand forward and backward simulations between arbitrary fuzzy automata, taking into account degrees of possibility of transitions and degrees of relationship.



BISIMULATIONS FOR FUZZY AUTOMATA

Now we are ready to move to consideration of bisimulations for fuzzy automata. In this section we give definitions and discuss the basic properties of bisimulations.

Let $\mathscr{A} = (A, \delta^A, \sigma^A, \tau^A)$ and $\mathscr{B} = (B, \delta^B, \sigma^B, \tau^B)$ be fuzzy automata, and let $\varphi \in \mathscr{F}(A \times B)$ be a non-empty fuzzy relation. We call φ a *forward simulation* if

$$\sigma^A \leqslant \sigma^B \circ \varphi^{-1}, \tag{4}$$

$$\varphi^{-1} \circ \delta^A_x \leqslant \delta^B_x \circ \varphi^{-1}, \quad \text{for every } x \in X, \tag{5}$$

$$\varphi^{-1} \circ \tau^A \leqslant \tau^B, \tag{6}$$

and a *backward simulation* if

$$\sigma^A \circ \varphi \leqslant \sigma^B, \tag{7}$$

$$\delta^A_x \circ \varphi \leqslant \varphi \circ \delta^B_x, \quad \text{for every } x \in X, \tag{8}$$

$$\tau^A \leqslant \varphi \circ \tau^B. \tag{9}$$

Furthermore, we call φ a *forward bisimulation* if both φ and φ^{-1} are forward simulations, that is, if φ satisfies (4)–(6) and

$$\sigma^B \leqslant \sigma^A \circ \varphi, \tag{10}$$

$$\varphi \circ \delta^B_x \leqslant \delta^A_x \circ \varphi, \quad \text{for every } x \in X, \tag{11}$$

$$\varphi \circ \tau^B \leqslant \tau^A, \tag{12}$$

and a *backward bisimulation*, if both φ and φ^{-1} are backward simulations, i.e., if φ satisfies (7)–(9) and

$$\sigma^B \circ \varphi^{-1} \leqslant \sigma^A, \tag{13}$$

$$\delta^B_x \circ \varphi^{-1} \leqslant \varphi^{-1} \circ \delta^A_x, \quad \text{for every } x \in X, \tag{14}$$

$$\tau^B \leqslant \varphi^{-1} \circ \tau^A. \tag{15}$$

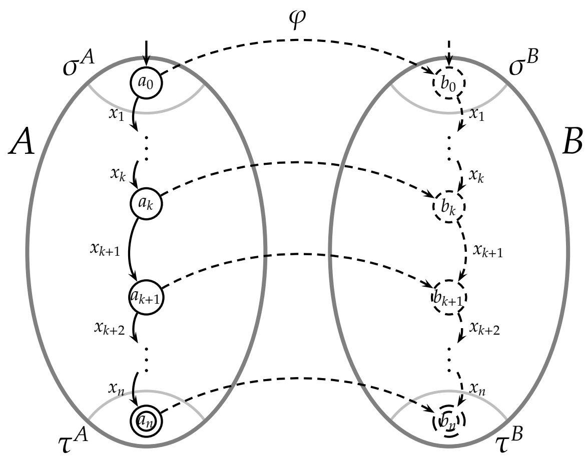


Figure 1: Forward and backward simulation

Also, if φ is a forward simulation and φ^{-1} is a backward simulation, i.e., if

$$\sigma^A = \sigma^B \circ \varphi^{-1}, \tag{16}$$

$$\varphi^{-1} \circ \delta_x^A = \delta_x^B \circ \varphi^{-1}, \quad \text{for every } x \in X, \tag{17}$$

$$\varphi^{-1} \circ \tau^A = \tau^B, \tag{18}$$

then φ is called a *forward-backward bisimulation*, or briefly a *FB-bisimulation*, and if φ is a backward simulation and φ^{-1} is a forward simulation, i.e., if

$$\sigma^A \circ \varphi = \sigma^B, \tag{19}$$

$$\delta_x^A \circ \varphi = \varphi \circ \delta_x^B, \quad \text{for every } x \in X, \tag{20}$$

$$\tau^A = \varphi \circ \tau^B. \tag{21}$$

Lecture (spoken)

We will now briefly refer to the key points.

As I pointed out the conjunction of bisimulations and uniform fuzzy relations is a very powerful tool in the study of equivalence between fuzzy automata. We can use acquired knowledge to solve the various problems related to bisimulations of deterministic, nondeterministic, fuzzy, and weighted automata. Getting back to what we discussed it is completely clear that forward and backward bisimulation fuzzy equivalence relations give equally good results in the state reduction of fuzzy automata.

Next time we will try to answer the question of how to overcome the lack of suitable ordering in semirings and develop the theory of forward bisimulations of weighted automata.

CONCLUDING REMARKS

In this article we have formed a a conjunction of bisimulations and uniform fuzzy relations as a very powerful tool in the study of equivalence between fuzzy automata. In this symbiosis, uniform fuzzy relations serve as fuzzy equivalence relations which relate elements of two possibly different sets, while bisimulations provide compatibility with the transitions, initial and terminal states of fuzzy automata. We have proved that a uniform fuzzy relation between fuzzy automata $\mathscr{A}$ and $\mathscr{B}$ is a forward bisimulation if and only if its kernel and co-kernel are forward bisimulation fuzzy equivalence relations on $\mathscr{A}$ and $\mathscr{B}$

and there is a special isomorphism between factor fuzzy automata with respect to these fuzzy equivalence relations.

In further research we will focus on weighted automata and try to answer the question of how to overcome the lack of suitable ordering in semirings and develop the theory of forward bisimulations. We also intend to further study backward-forward bisimulations for weighted automata. In the theory of fuzzy automata we will even more deeply explore relationships between the language-equivalence and various types of structural equivalence between fuzzy automata, and we will try to apply the methodology developed here in the study of fuzzy automata with outputs.

Jelena Ignjatović