Fuzzy Minimax Nets

Linh Anh Nguyen, Ivana Micić and Stefan Stanimirović

***Abstract*—We introduce fuzzy minimax nets as a novel tool to compute the greatest fuzzy bisimulation/simulation between two finite fuzzy labeled graphs. Fuzzy labeled graphs are a universal data structure for representing fuzzy systems such as fuzzy automata, fuzzy labeled transition systems, fuzzy Kripke models, fuzzy social networks and fuzzy interpretations in description logic. The greatest fuzzy bisimulation between two such systems characterizes the similarity between their states, actors or individuals. Using fuzzy minimax nets, we design the first algorithms for the mentioned computational problems in the case of using the product t-norm, as well as the first algorithms whose complexity order does not depend on the fuzzy values occurring in the inputs for those problems in the case of using the Łukasiewicz t-norm.**

***Index Terms*—Fuzzy bisimulation, Łukasiewicz t-norm, product t-norm.**

I. Introduction

This work is devoted to developing algorithms for computing the greatest fuzzy bisimulation/simulation between two finite fuzzy graph-based structures (FGBSs for short). Recently, scholars have been studying bisimulations and simulations for various FGBSs, including fuzzy/weighted automata [1]–[5], fuzzy (labeled) transition systems [6]–[12], fuzzy/many-valued Kripke models [13]–[16], fuzzy/weighted social networks [17], [18] and fuzzy interpretations in description logic [19]–[21]. To compare the behaviors of objects (states, actors or individuals) in such structures, one can use crisp or fuzzy bisimulation/simulation. While crisp bisimulations characterize the indiscernibility of objects, fuzzy bisimulations characterize the similarity between objects. The greatest fuzzy bisimulation Z between two image-finite FGBSs S and S' has the property that, for every object x of S and x' of S', $Z(x, x') = \inf\{\varphi^S(x) \Leftrightarrow \varphi^{S'}(x') \mid \varphi$ is a formula of a certain fuzzy modal/description logic$\}$, where $\Leftrightarrow$ denotes the fuzzy equivalence in the chosen logic. This property is called the Hennessy-Milner property of fuzzy bisimulations [14], [16], [17], [20]. The Hennessy-Milner property of crisp bisimulations [14], [17], [20] (respectively, fuzzy simulations [8], [9], [12]) uses crisp equality '$=$' (respectively, fuzzy implication '$\Rightarrow$') in the place of '$\Leftrightarrow$'.

L.A. Nguyen is with the Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Banacha 2, 02-097 Warsaw, Poland (e-mail: nguyen@mimuw.edu.pl) and the Faculty of Information Technology, Nguyen Tat Thanh University, Ho Chi Minh City, Vietnam. I. Micić and S. Stanimirović are with the Faculty of Sciences and Mathematics, University of Niš, Višegradska 33, 18000 Niš, Serbia (e-mail: ivana.micic@pmf.edu.rs and stefan.stanimirovic@pmf.edu.rs). I. Micić and S. Stanimirović acknowledge the support of the Science Fund of the Republic of Serbia, GRANT No 7750185, Quantitative Automata Models: Fundamental Problems and Applications - QUAM, and the Ministry of Education, Science and Technological Development, Republic of Serbia, Contract No. 451-03-68/2022-14/200124. This work is partially developed during Stanimirović's visit at the University of Warsaw within the framework of the Polish National Agency for Academic Exchange (NAWA).

A. Related Work

In [2], Ćirić *et al.* gave a method for computing the greatest fuzzy bisimulation/simulation between two finite fuzzy automata, but did not present any complexity analysis. Following [2], Ignjatović *et al.* [18] provided a procedure that runs in time $O(ln^5)$ for determining the greatest fuzzy bisimulation between two fuzzy social networks, where n is the size of the networks and l is the number of different fuzzy values generated by the procedure. Fuzzy bisimulations and simulations studied in [2], [18] are defined over a complete residuated lattice.

In [21], Nguyen and Tran designed an algorithm with the complexity $O((m+n)n)$ for computing the greatest fuzzy bisimulation between two finite fuzzy interpretations in the fuzzy description logic $f\mathcal{ALC}$ under the Gödel semantics, where n is the number of individuals and m is the number of non-zero instances of roles in the given fuzzy interpretations. They also adapted that algorithm for computing fuzzy bisimulations/simulations between finite fuzzy automata.

In [22], Nguyen gave an algorithm with the complexity $O((m \log l + n) \log n)$ for computing the fuzzy partition corresponding to the greatest fuzzy auto-bisimulation of a finite fuzzy labeled graph G under the Gödel semantics, where n, m and l are the number of vertices, the number of non-zero edges and the number of different fuzzy degrees of edges of G, respectively. By using that algorithm, he also provided an algorithm with the complexity $O(m. \log l. \log n + n^2)$ for computing the greatest fuzzy bisimulation between two finite fuzzy labeled graphs under the Gödel semantics.

Other related works include the work [23] by Micić *et al.* on computing the greatest right/left invariant fuzzy quasi-order/equivalence of a finite fuzzy automaton and the works [11], [24]–[26] on computing the largest crisp bisimulation/simulation between fuzzy graph-based structures.

B. Motivation

As stated above, fuzzy bisimulations and simulations studied in [2], [18] are defined over a complete residuated lattice. The notion of such a structure serves as a general setting for the underlying set of truth values, as it includes the well-known and most applied t-norms which are named after Gödel, Goguen (also known as product) and Łukasiewicz. However, the procedures with the time complexity $O(ln^5)$ developed in [2], [18] for computing the greatest fuzzy bisimulations/simulations behave quite differently depending on the choice of a concrete t-norm. When the Gödel t-norm is used, l is finite and these procedures have the termination property. However, l may be infinite when the product t-norm is used and is dependent on the fuzzy values occurring in the inputs when the Łukasiewicz t-norm is used. Thus, the procedures from [2], [18] are not algorithms (with the termination property) when the product t-norm is used, and their complexity may not be bounded by any function of the size of the inputs when the Łukasiewicz t-norm is used. To see this, consider the social networks N and N' illustrated below, which contain the only actor a or a', respectively. The signature consists of a proposition p and a relation r, while $0 < \varepsilon < 1$ is a parameter.

Applying the procedure given in [18] to compute the greatest fuzzy bisimulation between N and N', when the product t-norm is used, l is infinity because all the values $(1-\varepsilon)^n$ with $n \in \mathbb{N}$ appear during the computation. When the Łukasiewicz t-norm is used instead, we have $l = \lceil 1/\varepsilon \rceil$.

Recall that the algorithms given in [21], [22] were designed (and work) only for the Gödel semantics. To the best of our knowledge, before the current work there were no known algorithms for computing the greatest fuzzy bisimulation/simulation between two finite fuzzy systems under the product semantics, and there were no known algorithms with a complexity expressed in the size of the inputs for computing the greatest fuzzy bisimulation/simulation between two finite fuzzy systems under the Łukasiewicz semantics. On the other hand, such algorithms are very needed in practical situations. For example, consider the closeness of relatives. One can assume that the closeness of a person to his/her mother is greater than the closeness of that person to his/her great-grandmother. To model that relationship, it is more suitable to use the product t-norm than the Gödel t-norm.

Another important application is in social network analysis. Consider a social network in which the actors are connected to the degree they are mutually interested in a product that needs to be advertised. It is natural to impose that the people following a page linked with this product have a higher degree of connectivity than their friends who do not follow it. Again, the product t-norm is more suitable in determining the degree to which two actors are connected. In order to obtain target groups that are suitable for showing an advertisement, we need to group actors according to the similarity based on a fuzzy auto-bisimulation defined for that social network. However, such networks have complex and nonrandom structures, which means that the previously developed procedures [2], [18] may fail to determine such a bisimulation. An algorithm (with the termination property) for the task is needed.

C. Our Contributions

In this work, we introduce fuzzy minimax nets as a novel tool to compute the greatest fuzzy bisimulation/simulation between two finite fuzzy labeled graphs. Fuzzy labeled graphs are a universal data structure for representing FGBSs. Using fuzzy minimax nets, we design the *first* algorithms for the mentioned computational problems in the case of using the product t-norm, as well as the *first* algorithms whose complexity order does not depend on the fuzzy values used by the input graphs for those problems in the case of using the Łukasiewicz t-norm.

A fuzzy minimax net contains two non-empty disjoint sets of nodes, called min-nodes and max-nodes, respectively. Each min-node has an upper marking limit, which belongs to $[0, 1]$. An edge of such a net connects a min-node to a max-node, or vice versa, and has a fuzzy weight, which belongs to $[0, 1]$. We study the problem of computing the so-called greatest correct marking of a finite fuzzy minimax net.

Fuzzy minimax nets present a very convenient way to detect and overcome the problem of infinite computations that can occur in computing the greatest bisimulation or simulation between two finite FGBSs. The computation is reduced to computing the greatest correct marking of the corresponding fuzzy minimax net. Despite being a theoretical notion, fuzzy minimax nets can be exploited in all application fields where a computation of fuzzy bisimulations is needed.

D. The Structure of the Work

The rest of this work is structured as follows. In Section II-A, we recall basic definitions of fuzzy sets and operators. In Section II-B, we recall definitions of fuzzy labeled graphs, fuzzy bisimulations and fuzzy simulations. In Section III, we formally define fuzzy minimax nets, study their relationship to fuzzy bisimulations/simulations, and introduce some related notions. In Section IV (respectively, V), we present our algorithm of computing the greatest correct marking of a finite fuzzy minimax net and exploit it to obtain algorithms for computing the greatest fuzzy bisimulation/simulation between two finite fuzzy graphs over the same signature in the case of using the product (respectively, Łukasiewicz) t-norm. Concluding remarks are given in Section VI. The Supplementary Material contains optimizations for our algorithms and some additional results.

II. Preliminaries

A. Fuzzy Sets and Operators

We use the fuzzy operators $\wedge, \vee : [0,1] \times [0,1] \to [0,1]$:

$$a \wedge b = \min\{a, b\}, \qquad a \vee b = \max\{a, b\}.$$

An operator $\otimes : [0,1] \times [0,1] \to [0,1]$ is called a *t-norm* if it is commutative and associative, has 1 as the neutral element, and is increasing with respect to both the arguments. If $\otimes$ is a left-continuous t-norm and $\Rightarrow : [0,1] \times [0,1] \to [0,1]$ is the operator defined by $(a \Rightarrow b) = \bigvee\{c \mid c \otimes a \leq b\}$, where $\bigvee$ denotes the supremum, then $\Rightarrow$ is called the *residuum* of $\otimes$. From now on, if not stated otherwise, we assume that $\otimes$ is a left-continuous t-norm and $\Rightarrow$ is its residuum. Thus, for every $a, a', b, b' \in [0, 1]$,

$$\text{if } a \leq a' \text{ and } b \leq b', \text{ then } a \otimes b \leq a' \otimes b', \tag{1}$$

$$\text{if } a' \leq a \text{ and } b \leq b', \text{ then } (a \Rightarrow b) \leq (a' \Rightarrow b'). \tag{2}$$

Furthermore, for every $a, b, c \in [0, 1]$,

$$a \le (b \Rightarrow c) \text{ iff } a \otimes b \le c, \tag{3}$$
$$0 \otimes a = 0, \tag{4}$$
$$(0 \Rightarrow a) = 1. \tag{5}$$

We define $(a \Leftrightarrow b) = (a \Rightarrow b) \wedge (b \Rightarrow a)$.

The most well-known t-norms are named after Gödel, Łukasiewicz and product. They are specified below together with their corresponding residua.

	Gödel	Product	Łukasiewicz
$a \otimes b$	$\min\{a, b\}$	$a \cdot b$	$\max\{0, a + b - 1\}$
$a \Rightarrow b$	$\begin{cases} 1 & \text{if } a \le b \\ b & \text{otherwise} \end{cases}$	$\begin{cases} 1 & \text{if } a \le b \\ b/a & \text{otherwise} \end{cases}$	$\min\{1, 1 - a + b\}$

Given a set X, a function $f : X \to [0, 1]$ is called a *fuzzy subset* of X. It is *empty* if $f(x) = 0$ for all $x \in X$. If f is a fuzzy subset of X and $x \in X$, then $f(x)$ means the fuzzy degree in which x belongs to the subset. For $\{x_1, \ldots, x_n\} \subseteq X$ and $\{a_1, \ldots, a_n\} \subset [0, 1]$, we write $\{x_1 : a_1, \ldots, x_n : a_n\}$ to denote the fuzzy subset f of X such that $f(x_i) = a_i$ for $1 \le i \le n$ and $f(x) = 0$ for $x \in X \setminus \{x_1, \ldots, x_n\}$.

If f and g are fuzzy subsets of X, then we write $f \le g$ to denote that $f(x) \le g(x)$ for all $x \in X$. If $f \le g$, then we say that g is *greater than or equal to* f. We write $f \lneq g$ to denote that $f \le g$ and $f \ne g$. If F is a set of fuzzy subsets of X, then by $\bigvee F$ we denote the fuzzy subset of X specified by $(\bigvee F)(x) = \bigvee\{f(x) \mid f \in F\}$. As usual, if $f \in F$ and $f = \bigvee F$, then f is called the *greatest* element of F. We write $f \vee g$ to denote $\bigvee\{f, g\}$.

Given non-empty sets X and Y, a fuzzy subset of $X \times Y$ is called a *fuzzy relation* between X and Y.

B. Fuzzy Bisimulations and Simulations

A *fuzzy labeled graph*, hereafter called a *fuzzy graph* for short, is a structure $G = \langle V, E, L, \Sigma_V, \Sigma_E \rangle$, where V is a non-empty set of vertices, Σ_V (respectively, Σ_E) is a set of vertex labels (respectively, edge labels), $E : V \times \Sigma_E \times V \to [0, 1]$ is called the fuzzy set of labeled edges, and $L : V \to (\Sigma_V \to [0, 1])$ is called the labeling function of vertices. Given vertices $x, y \in V$, a vertex label $p \in \Sigma_V$ and an edge label $r \in \Sigma_E$, $L(x)(p)$ means the degree in which p is a member of the label of x, and $E(x, r, y)$ means the degree in which there is an edge from x to y labeled by r. For $r \in \Sigma_E$, we write E_r to denote the fuzzy subset of $V \times V$ such that $E_r(x, y) = E(x, r, y)$ for $x, y \in V$. The graph G is *finite* if all the sets V, Σ_V and Σ_E are finite. It is *image-finite* if the set $\{y \mid E(x, r, y) > 0\}$ is finite for all $x \in V$ and $r \in \Sigma_E$.

Fuzzy graphs are used as fuzzy labeled transition systems (FLTSs), fuzzy automata, fuzzy Kripke models and fuzzy interpretations in fuzzy description logics. For example, in the terminology of FLTSs, vertices, edges, edge labels and vertex labels represent states, transitions, actions and atomic properties of states, respectively. Recall that fuzzy bisimulations have been defined and studied for fuzzy automata [1], [2], weighted/fuzzy social networks [17], [18], fuzzy modal logics [14], [16] and fuzzy description logics [20], [21]. We give below their definition, which is based on [16], [20] and equivalent to the one in [14] when $|\Sigma_E| = 1$ and the graphs are image-finite.

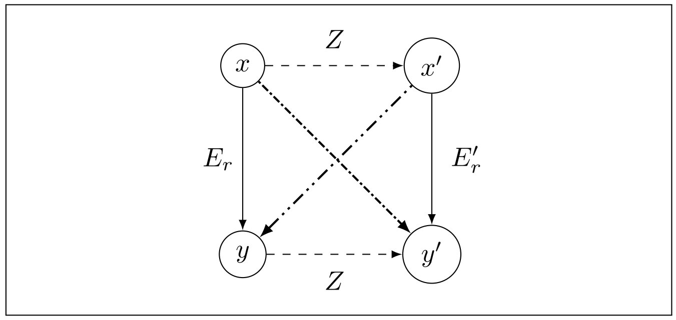


Fig. 1. An illustration for Definitions 1 and 7.

Definition 1: Let $G = \langle V, E, L, \Sigma_V, \Sigma_E \rangle$ and $G' = \langle V', E', L', \Sigma_V, \Sigma_E \rangle$ be fuzzy graphs over the same signature $\langle \Sigma_V, \Sigma_E \rangle$. A fuzzy relation $Z : V \times V' \to [0, 1]$ is called a *fuzzy bisimulation* between G and G' if the following conditions hold for all $p \in \Sigma_V$, $r \in \Sigma_E$ and all possible values for the free variables:

$$Z(x, x') \le (L(x)(p) \Leftrightarrow L'(x')(p)) \tag{6}$$
$$\exists y' \in V' \; (Z(x, x') \otimes E_r(x, y) \le E'_r(x', y') \otimes Z(y, y')) \tag{7}$$
$$\exists y \in V \; (Z(x, x') \otimes E'_r(x', y') \le E_r(x, y) \otimes Z(y, y')). \tag{8}$$

A fuzzy relation $Z : V \times V' \to [0, 1]$ is called a *fuzzy simulation* between G and G' if the following condition and the condition (7) hold for all $p \in \Sigma_V$, $r \in \Sigma_E$ and all possible values for the free variables:

$$Z(x, x') \le (L(x)(p) \Rightarrow L'(x')(p)). \tag{9}$$

The conditions (7) and (8) are illustrated in Figure 1.

III. Fuzzy Minimax Nets

A *(fuzzy) minimax net* is a tuple $N = \langle V_{min}, V_{max}, E, L \rangle$, where V_{min} and V_{max} are non-empty disjoint sets of *nodes*, called *min-nodes* and *max-nodes*, respectively, $E : (V_{min} \times V_{max}) \cup (V_{max} \times V_{min}) \to [0, 1]$ is a fuzzy set of *edges*, and $L : V_{min} \to [0, 1]$ is the *marking limit for min-nodes*. It is *coimage-finite* if, for every $y \in V_{min} \cup V_{max}$, the set $\{x \in V_{min} \cup V_{max} \mid E(x, y) > 0\}$ is finite. A pair $\langle x, y \rangle \in (V_{min} \times V_{max}) \cup (V_{max} \times V_{min})$ is called a *positive* edge of N if $E(x, y) > 0$. The *size* of E, denoted by $|E|$, is defined to be the number of positive edges of N.

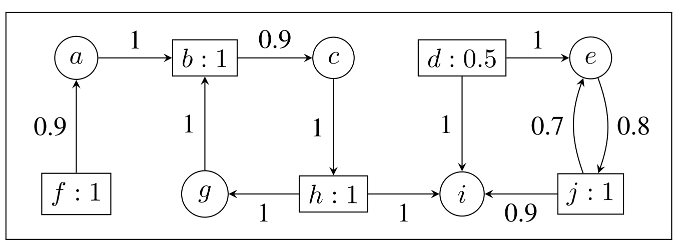


Fig. 2. An illustration of a minimax net $N = \langle V_{min}, V_{max}, E, L \rangle$, where $V_{max} = \{a, c, e, g, i\}$ (max-nodes are represented by circles), $V_{min} = \{b, d, f, h, j\}$ (min-nodes are represented by rectangles), the number in a rectangle representing a min-node x denotes the marking limit $L(x)$, a positive edge $\langle x, y \rangle$ is represented by an arrow from x to y with label $E(x, y)$.

An example of a minimax net is given in Fig. 2.

Given a minimax net $N = \langle V_{min}, V_{max}, E, L\rangle$, a fuzzy set $M : V_{min} \cup V_{max} \to [0,1]$ is called a *marking* of N. It is a *correct marking* of N if the following conditions hold for every $x \in V_{min}$ and $y \in V_{max}$:

$$M(x) \leq L(x) \quad (10)$$
$$M(x) \leq \bigwedge_{z\in V_{max}} (E(z,x) \Rightarrow M(z)) \quad (11)$$
$$M(y) \leq \bigvee_{z\in V_{min}} (E(z,y) \otimes M(z)). \quad (12)$$

A marking M of N is *stable* if the following conditions hold for every $x \in V_{min}$ and $y \in V_{max}$:

$$M(x) = L(x) \wedge \bigwedge_{z\in V_{max}} (E(z,x) \Rightarrow M(z)) \quad (13)$$
$$M(y) = \bigvee_{z\in V_{min}} (E(z,y) \otimes M(z)). \quad (14)$$

Note that, by definition, stable markings are correct.

Example 2: Consider the minimax net N specified in Fig. 2 and assume that $\otimes$ is the product t-norm. Let M be the marking of N specified as follows:

- $M(f) = 1$, $M(a) = 0.9$,
- $M(b) = M(c) = M(h) = M(g) = 0$,
- $M(d) = M(e) = 0.5$,
- $M(j) = (0.8 \Rightarrow 0.5) = 0.625$,
- $M(i) = 0.9 \otimes 0.625 = 0.5625$.

It is easy to check that M is a stable marking of N. ■

Lemma 3: Let $N = \langle V_{min}, V_{max}, E, L\rangle$ be a minimax net and $\mathcal{M}$ a set of correct markings of N. Then, $\bigvee\mathcal{M}$ is also a correct marking of N.

Proof: Clearly, $(\bigvee\mathcal{M})(x) \leq L(x)$ for all $x \in V_{min}$. We need to prove that, for every $x \in V_{min}$ and $y \in V_{max}$,

$$(\textstyle\bigvee\mathcal{M})(x) \leq \bigwedge_{z\in V_{max}} (E(z,x) \Rightarrow (\textstyle\bigvee\mathcal{M})(z)) \quad (15)$$
$$(\textstyle\bigvee\mathcal{M})(y) \leq \bigvee_{z\in V_{min}} (E(z,y) \otimes (\textstyle\bigvee\mathcal{M})(z)). \quad (16)$$

Let M be an arbitrary element of $\mathcal{M}$. To prove (15), it suffices to show that, for every $x \in V_{min}$ and $z \in V_{max}$,

$$M(x) \leq (E(z,x) \Rightarrow (\textstyle\bigvee\mathcal{M})(z)).$$

This holds due to (11) and (2). To prove (16), it suffices to show that, for every $y \in V_{max}$,

$$M(y) \leq \bigvee_{z\in V_{min}} (E(z,y) \otimes (\textstyle\bigvee\mathcal{M})(z)).$$

This holds due to (12) and (1). ■

Corollary 4: Every minimax net has the greatest correct marking.

Proposition 5: The greatest correct marking of a minimax net is stable.

Proof: Let $N = \langle V_{min}, V_{max}, E, L\rangle$ be a minimax net and M the greatest correct marking of N. By (10) and (11), we have that, for every $x \in V_{min}$,

$$M(x) \leq L(x) \wedge \bigwedge_{z\in V_{max}} (E(z,x) \Rightarrow M(z)).$$

To prove (13), for a contradiction, assume that there exists $x_0 \in V_{min}$ such that

$$M(x_0) < L(x_0) \wedge \bigwedge_{z\in V_{max}} (E(z,x_0) \Rightarrow M(z)).$$

Let M' be the marking of N that differs from M only in that

$$M'(x_0) = L(x_0) \wedge \bigwedge_{z\in V_{max}} (E(z,x_0) \Rightarrow M(z)).$$

It is easy to see that, due to (1) and (2), M' is a correct marking of N. This contradicts the fact that M is the greatest correct marking of N. Therefore, (13) holds. The assertion (14) can be proved analogously. ■

Given a minimax net $N = \langle V_{min}, V_{max}, E, L\rangle$, denote the set $(V_{min} \cup V_{max} \to [0,1])$ of all markings of N as $\mathcal{M}_N$. It is a complete lattice. Let *fire* : $\mathcal{M}_N \to \mathcal{M}_N$ be the function defined as follows, for $M \in \mathcal{M}_N$, $x \in V_{min}$ and $y \in V_{max}$:

$$\mathit{fire}(M)(x) = L(x) \wedge \bigwedge_{z\in V_{max}} (E(z,x) \Rightarrow M(z)),$$
$$\mathit{fire}(M)(y) = \bigvee_{z\in V_{min}} (E(z,y) \otimes M(z)).$$

By (1) and (2), *fire* is a monotonic function. By the Knaster-Tarski theorem, the greatest fixpoint of *fire* is $\bigvee\{M \in \mathcal{M}_N \mid M \leq \mathit{fire}(M)\}$. It is the greatest stable marking of N. By Proposition 5, we can derive the following result.

Lemma 6: The greatest correct marking of a minimax net N is $\bigvee\{M \in \mathcal{M}_N \mid M \leq \mathit{fire}(M)\}$.

A. The Relationship between Minimax Nets and Fuzzy Bisimulations/Simulations

As stated before, minimax nets are a tool to compute the greatest fuzzy bisimulation between two finite fuzzy graphs when the product or Łukasiewicz t-norm is used.

Definition 7: Let $G = \langle V, E, L, \Sigma_V, \Sigma_E\rangle$ and $G' = \langle V', E', L', \Sigma_V, \Sigma_E\rangle$ be fuzzy graphs over the same signature $\langle\Sigma_V, \Sigma_E\rangle$. The *minimax net bisimulatedly corresponding to the pair* $\langle G, G'\rangle$ is $N = \langle V_{min}, V_{max}, E'', L''\rangle$ specified as follows:

- $V_{min} = V \times V'$,
- $V_{max} = (V \times V' \times \Sigma_E) \cup (V' \times V \times \Sigma_E)$,
- $E'' : (V_{min} \times V_{max}) \cup (V_{max} \times V_{min}) \to [0,1]$ is the following fuzzy set

$$\{\langle\langle y,y'\rangle, \langle x',y,r\rangle\rangle : E'_r(x',y') \mid y \in V, x',y' \in V', r \in \Sigma_E\} \vee$$
$$\{\langle\langle x',y,r\rangle, \langle x,x'\rangle\rangle : E_r(x,y) \mid x,y \in V, x' \in V', r \in \Sigma_E\} \vee$$
$$\{\langle\langle y,y'\rangle, \langle x,y',r\rangle\rangle : E_r(x,y) \mid x,y \in V, y' \in V', r \in \Sigma_E\} \vee$$
$$\{\langle\langle x,y',r\rangle, \langle x,x'\rangle\rangle : E'_r(x',y') \mid x \in V, x',y' \in V', r \in \Sigma_E\},$$

- $L'' : V_{min} \to [0,1]$ is the fuzzy set specified as follows, for all $x \in V$ and $x' \in V'$:
$$L''(\langle x, x' \rangle) = \bigwedge_{p \in \Sigma_V} (L(x)(p) \Leftrightarrow L'(x')(p)).$$

The *minimax net simulatedly corresponding to* $\langle G, G' \rangle$ is $N = \langle V_{min}, V_{max}, E'', L'' \rangle$ specified as follows:

- $V_{min} = V \times V'$,
- $V_{max} = V \times V' \times \Sigma_E$,
- $E'' : (V_{min} \times V_{max}) \cup (V_{max} \times V_{min}) \to [0,1]$ is the following fuzzy set
$$\{\langle \langle y, y' \rangle, \langle x', y, r \rangle \rangle : E'_r(x', y') \mid y \in V, x', y' \in V', r \in \Sigma_E\} \vee \{\langle \langle x', y, r \rangle, \langle x, x' \rangle \rangle : E_r(x, y) \mid x, y \in V, x' \in V', r \in \Sigma_E\},$$
- $L'' : V_{min} \to [0,1]$ is the fuzzy set specified as follows, for all $x \in V$ and $x' \in V'$:
$$L''(\langle x, x' \rangle) = \bigwedge_{p \in \Sigma_V} (L(x)(p) \Rightarrow L'(x')(p)).$$

Figure 1 provides intuition behind the definition of E''. Observe that, in a certain sense, the directions of the edges of N are opposite to the directions of the edges of G and G'. This reflects the backward propagation (of constraints) in computing the greatest fuzzy bisimulation/simulation between G and G'.

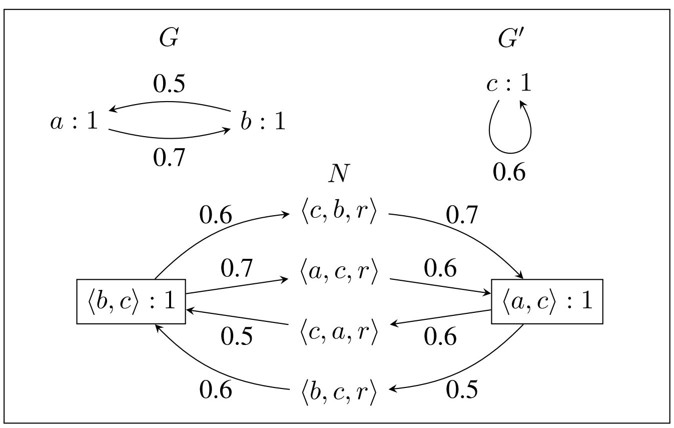


Fig. 3. An illustration for Example 8.

Example 8: Let $\Sigma_V = \{p\}$, $\Sigma_E = \{r\}$ and let $G = \langle V, E, L, \Sigma_V, \Sigma_E \rangle$ and $G' = \langle V', E', L', \Sigma_V, \Sigma_E \rangle$ be the fuzzy graphs depicted in Fig. 3 and specified as follows:

- $V = \{a, b\}$, $E = \{\langle a, r, b \rangle : 0.7,\ \langle b, r, a \rangle : 0.5\}$, $L(a)(p) = L(b)(p) = 1$;
- $V' = \{c\}$, $E' = \{\langle c, r, c \rangle : 0.6\}$, $L'(c)(p) = 1$.

The minimax net bisimulatedly corresponding to $\langle G, G' \rangle$ is the minimax net $N = \langle V_{min}, V_{max}, E'', L'' \rangle$ with:

- $V_{min} = \{\langle a, c \rangle, \langle b, c \rangle\}$,
- $V_{max} = \{\langle a, c, r \rangle, \langle b, c, r \rangle, \langle c, a, r \rangle, \langle c, b, r \rangle\}$,
- $L''(\langle a, c \rangle) = L''(\langle b, c \rangle) = (1 \Leftrightarrow 1) = 1$,
- the fuzzy set E'' of edges depicted in Fig. 3. ■

Proposition 9: If N is the minimax net simulatedly or bisimulatedly corresponding to $\langle G, G' \rangle$, where G and G' are image-finite fuzzy graphs over the same signature $\langle \Sigma_V, \Sigma_E \rangle$ with Σ_E being finite, then N is coimage-finite.

This proposition clearly holds.

Lemma 10: Let $G = \langle V, E, L, \Sigma_V, \Sigma_E \rangle$ and $G' = \langle V', E', L', \Sigma_V, \Sigma_E \rangle$ be image-finite fuzzy graphs over the same signature and let $N = \langle V_{min}, V_{max}, E'', L'' \rangle$ be their bisimulatedly corresponding minimax net. Let Z be a fuzzy subset of $V \times V'$ and let $M : V_{min} \cup V_{max} \to [0,1]$ be

$$Z \vee \{\langle x', y, r \rangle : (E'_r \circ Z^{-1})(x', y) \mid x' \in V', y \in V, r \in \Sigma_E\} \vee \{\langle x, y', r \rangle : (E_r \circ Z)(x, y') \mid x \in V, y' \in V', r \in \Sigma_E\}.$$

Then, Z is a fuzzy bisimulation between G and G' iff M is a correct marking of N.

Proof: Assume that Z is a fuzzy bisimulation between G and G'. To prove that M is a correct marking of N, we need to show that, for every $x, y \in V$, $x', y' \in V'$ and $r \in \Sigma_E$,

$$M(\langle x, x' \rangle) \le L''(\langle x, x' \rangle)$$
$$M(\langle x, x' \rangle) \le \bigwedge_{y \in V, r \in \Sigma_E} (E''(\langle x', y, r \rangle, \langle x, x' \rangle) \Rightarrow M(\langle x', y, r \rangle))$$
$$M(\langle x, x' \rangle) \le \bigwedge_{y' \in V, r \in \Sigma_E} (E''(\langle x, y', r \rangle, \langle x, x' \rangle) \Rightarrow M(\langle x, y', r \rangle))$$
$$M(\langle x', y, r \rangle) \le \bigvee_{y' \in V'} (E''(\langle y, y' \rangle, \langle x', y, r \rangle) \otimes M(\langle y, y' \rangle))$$
$$M(\langle x, y', r \rangle) \le \bigvee_{y \in V} (E''(\langle y, y' \rangle, \langle x, y', r \rangle) \otimes M(\langle y, y' \rangle)).$$

Denote these assertions by $(*)$. We need to prove that, for every $x, y \in V$, $x', y' \in V'$ and $r \in \Sigma_E$,

$$Z(x, x') \le \bigwedge_{p \in \Sigma_V} (L(x)(p) \Leftrightarrow L'(x')(p)) \quad (17)$$
$$Z(x, x') \le (E_r(x, y) \Rightarrow (E'_r \circ Z^{-1})(x', y)) \quad (18)$$
$$Z(x, x') \le (E'_r(x', y') \Rightarrow (E_r \circ Z)(x, y')) \quad (19)$$
$$(E'_r \circ Z^{-1})(x', y) \le \bigvee_{y' \in V'} (E'_r(x', y') \otimes Z(y, y')) \quad (20)$$
$$(E_r \circ Z)(x, y') \le \bigvee_{y \in V} (E_r(x, y) \otimes Z(y, y')). \quad (21)$$

The assertion (17) follows from (6). The assertion (18) follows from (3) and (7). The assertion (19) follows from (3) and (8). The assertions (20) and (21) clearly hold.

For the converse, assume that M is a correct marking of N, that is, the assertions $(*)$ hold for all $x, y \in V$, $x', y' \in V'$ and $r \in \Sigma_E$. We need to show that Z satisfies the conditions (6)–(8). The assertions $(*)$ imply (17)–(21). The condition (6) follows from (17). The condition (7) follows from (18), (3) and the assumption that G' is image-finite. Similarly, the condition (8) follows from (19), (3) and the assumption that G is image-finite. ■

Theorem 11: Let $G = \langle V, E, L, \Sigma_V, \Sigma_E \rangle$ and $G' = \langle V', E', L', \Sigma_V, \Sigma_E \rangle$ be image-finite fuzzy graphs over the same signature and let $N = \langle V_{min}, V_{max}, E'', L'' \rangle$ be their bisimulatedly corresponding minimax net. Let M be the greatest correct marking of N. Then, $M|_{V_{min}}$ is the greatest fuzzy bisimulation between G and G'.

Proof: Let $Z = M|_{V_{min}}$. By the assertion (14) of Proposition 5, for every $x, y \in V$, $x', y' \in V'$ and $r \in \Sigma_E$,

$$M(\langle x', y, r\rangle) = \bigvee_{y' \in V'} (E'_r(x', y') \otimes Z(y, y')),$$
$$M(\langle x, y', r\rangle) = \bigvee_{y \in V} (E_r(x, y) \otimes Z(y, y')),$$

which mean

$$M(\langle x', y, r\rangle) = (E'_r \circ Z^{-1})(x', y),$$
$$M(\langle x, y', r\rangle) = (E_r \circ Z)(x, y').$$

Thus, Z and M satisfy the conditions stated in Lemma 10. Therefore, Z is a fuzzy bisimulation between G and G'. For a contradiction, assume that there exists a fuzzy bisimulation Z_2 between G and G' such that $Z \lneq Z_2$. Let M_2 be defined as M in Lemma 10 but using Z_2 instead of Z. By Lemma 10, M_2 is a correct marking of N, and we have $M \lneq M_2$, which contradicts the assumption that M is the greatest correct marking of N. Therefore, Z is the greatest fuzzy bisimulation between G and G'. ∎

Lemma 12 and Theorem 13 given below are counterparts of Lemma 10 and Theorem 11, respectively, devoted to fuzzy simulations instead of fuzzy bisimulations. They can be proved analogously.

Lemma 12: Let $G = \langle V, E, L, \Sigma_V, \Sigma_E\rangle$ and $G' = \langle V', E', L', \Sigma_V, \Sigma_E\rangle$ be image-finite fuzzy graphs over the same signature and let $N = \langle V_{min}, V_{max}, E'', L''\rangle$ be their simulatedly corresponding minimax net. Let Z be a fuzzy subset of $V \times V'$ and let $M : V_{min} \cup V_{max} \to [0, 1]$ be

$$Z \vee \{\langle x', y, r\rangle : (E'_r \circ Z^{-1})(x', y) \mid x' \in V', y \in V, r \in \Sigma_E\}.$$

Then, Z is a fuzzy simulation between G and G' iff M is a correct marking of N.

Theorem 13: Let $G = \langle V, E, L, \Sigma_V, \Sigma_E\rangle$ and $G' = \langle V', E', L', \Sigma_V, \Sigma_E\rangle$ be image-finite fuzzy graphs over the same signature and let $N = \langle V_{min}, V_{max}, E'', L''\rangle$ be their simulatedly corresponding minimax net. Let M be the greatest correct marking of N. Then, $M|_{V_{min}}$ is the greatest fuzzy simulation between G and G'.

B. Traces of Markings

We define traces of markings and related notions as a preparation for developing algorithms to compute the greatest correct marking of a finite minimax net.

Definition 14: Let $N = \langle V_{min}, V_{max}, E, L\rangle$ be a coimage-finite minimax net and M a stable marking of N. Let $V = V_{min} \cup V_{max}$. A *trace* of M is a function $\pi : V \to (V \cup \{null\})$ satisfying the following conditions, where the name π stands for "predecessor":

- if $x \in V_{max}$, then
  - if there is no $v \in V_{min}$ such that $E(v, x) > 0$, then $\pi(x) = null$,
  - else $\pi(x)$ is an element $v \in V_{min}$ such that $E(v, x) > 0$ and $M(x) = E(v, x) \otimes M(v)$;
- if $x \in V_{min}$, then
  - if $L(x) \leq (E(v, x) \Rightarrow M(v))$ for all $v \in V_{max}$, then $\pi(x) = null$,
  - else $\pi(x)$ is an element $v \in V_{max}$ such that $E(v, x) > 0$ and $M(x) = (E(v, x) \Rightarrow M(v))$.

Observe that this definition is well-specified due to the assertions (4) and (5).

Definition 15: Let $N = \langle V_{min}, V_{max}, E, L\rangle$ be a coimage-finite minimax net and M a stable marking of N. The *tracing rank* of a node v of N with respect to M, denoted by $t\_rank(v, M)$, is

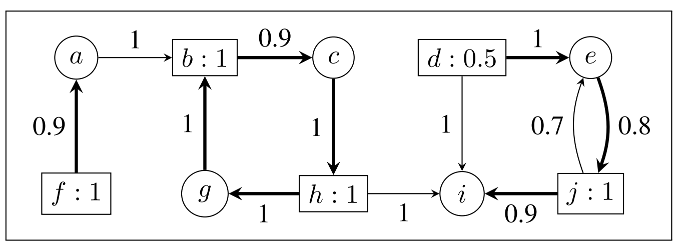
- either the smallest natural number k such that there exist a trace π of M and a sequence $v_0, v_1, \ldots, v_k$ of nodes of N with the properties that $v_0 = v$, $v_i = \pi(v_{i-1})$ for $0 < i \leq k$, and $\pi(v_k) = null$;
- or ω otherwise (where $\omega > k$ for all $k \in \mathbb{N}$).

Definition 16: Let $N = \langle V_{min}, V_{max}, E, L\rangle$ be a coimage-finite minimax net and M a stable marking of N. A trace π of M is *canonical* if it satisfies the following conditions for all $x \in V_{min} \cup V_{max}$ with $\pi(x) \neq null$:

- if $x \in V_{max}$, then
  if there is $v \in V_{min}$ such that $v \neq \pi(x)$, $E(v, x) > 0$ and $M(x) = E(v, x) \otimes M(v)$, then $t\_rank(\pi(x), M) \leq t\_rank(v, M)$;
- if $x \in V_{min}$, then
  if there is $v \in V_{max}$ such that $v \neq \pi(x)$, $E(v, x) > 0$ and $M(x) = (E(v, x) \Rightarrow M(v))$, then $t\_rank(\pi(x), M) \leq t\_rank(v, M)$.

Remark 17: Given a coimage-finite minimax net N and a stable marking M of N, there exists a canonical trace of M.

Example 18: Assume that $\otimes$ is the product t-norm. Reconsider the minimax net N specified in Fig. 2 and the marking M of N specified in Example 2. This marking has only one trace π, which is highlighted by thick arrows in the following figure and formally specified below.



- $\pi(f) = null$, $\pi(a) = f$,
- $\pi(b) = g$, $\pi(g) = h$, $\pi(h) = c$, $\pi(c) = b$,
- $\pi(i) = j$, $\pi(j) = e$, $\pi(e) = d$, $\pi(d) = null$.

As π is the unique trace of M, it is canonical. We have:

- $t\_rank(f, M) = 0$, $t\_rank(a, M) = 1$,
- $t\_rank(b, M) = t\_rank(g, M) =$ $= t\_rank(h, M) = t\_rank(c) = \omega$,
- $t\_rank(d, M) = 0$, $t\_rank(e, M) = 1$, $t\_rank(j, M) = 2$, $t\_rank(i, M) = 3$. ∎

Definition 19: Let $N = \langle V_{min}, V_{max}, E, L\rangle$ be a minimax net and π a function from V to $V \cup \{null\}$, where $V = V_{min} \cup V_{max}$. A finite set $W \subseteq V$ is called a *cycle* of π if there exists an enumeration $(v_0, \ldots, v_k)$ of the nodes of W such that $v_i = \pi(v_{i-1})$, for all $0 < i \le k$, and $v_0 = \pi(v_k)$. We call such a sequence $(v_0, \ldots, v_k)$ an *enumeration* of the cycle and sometimes refer to it just as a cycle.

Note that the size of a cycle of a trace is an even number and different cycles of a trace are disjoint.

Lemma 20: Let $N = \langle V_{min}, V_{max}, E, L\rangle$ be a coimage-finite minimax net, M a stable marking of N, and π a canonical trace of M. If $u, v \in V_{min} \cup V_{max}$, $t\_rank(v, M) = \omega$ and $u = \pi^k(v)$ for some k, then $t\_rank(u, M) = \omega$. If W is a cycle of π, then $t\_rank(v, M) = \omega$ for all $v \in W$.

Proof: The first assertion clearly holds. Consider the second assertion and, for a contradiction, suppose $t\_rank(v_0, M) = n$ for some $v_0 \in W$. Let $k = |W|$ and $v_i = \pi(v_{i-1})$ for $0 < i < k$. Thus, $v_0 = \pi(v_{k-1})$. Since $t\_rank(v_0, M) = n$, we must have $t\_rank(v_1, M) = n-1$. Repeating this understanding, we can derive $t\_rank(v_{k-1}, M) = n - k + 1$. Since $v_0 = \pi(v_k)$, we can further derive $t\_rank(v_0, M) = n - k$, which contradicts the assumption that $t\_rank(v_0, M) = n$ since $k > 0$. ■

IV. The Case of Using the Product t-Norm

In this section, assume that $\otimes$ is the product t-norm. We study the problem of computing the greatest correct marking of a finite minimax net under this assumption.

Lemma 21: Assume that $\otimes$ is the product t-norm. Let $N = \langle V_{min}, V_{max}, E, L\rangle$ be a finite minimax net and M the greatest correct marking of N. Then, for every node v of N with $t\_rank(v, M) = \omega$, $M(v) = 0$.

Proof: By Proposition 5, $\mathit{fire}(M) = M$ and M is the greatest fixpoint of fire. Let $V = V_{min} \cup V_{max}$, $W = \{v \in V \mid t\_rank(v, M) = \omega\}$ and let π be a canonical trace of M. For a contradiction, suppose W contains some nodes v with $M(v) > 0$. Let p be the greatest real number such that, for every min-node $v \in W$,

- if $M(v) > 0$, then $p \le L(v)/M(v)$;
- if $w \in V_{max}$ and
$$(E(w, v) \Rightarrow M(w)) > (E(\pi(v), v) \Rightarrow M(\pi(v))) > 0,$$
then
$$p \le (E(w, v) \Rightarrow M(w))/(E(\pi(v), v) \Rightarrow M(\pi(v))). \quad (22)$$

By Lemma 20, there must exist $v \in W \cap V_{min}$ such that $M(v) > 0$. The value p is well-defined due to this reason and the assumption that N is finite. Moreover, $p > 1$. Let $M' \in \mathcal{M}_N$ be the marking specified as follows: if $v \in W$, then $M'(v) = M(v) \cdot p$, else $M'(v) = M(v)$. Clearly, $M \lneq M'$. We show that $M' \le \mathit{fire}(M')$.

- For $v \in V - W$, since $M' \gneq M$, by (1) and (2),
$$\mathit{fire}(M')(v) \ge \mathit{fire}(M)(v) = M(v) = M'(v).$$
- Consider any node $v \in W \cap V_{max}$ and let $u = \pi(v)$. By Lemma 20, $u \in W$. Since $v \in V_{max}$, π is a canonical trace of M and $\pi(v) = u$, we have
$$M(v) = E(u, v) \otimes M(u) \ge E(w, v) \otimes M(w)$$
for all $w \in V_{min}$. Hence, for every $w \in V_{min}$,
$$E(u, v) \otimes M'(u) = E(u, v) \otimes (M(u) \cdot p) \ge$$
$$\ge E(w, v) \otimes (M(w) \cdot p) \ge E(w, v) \otimes M'(w).$$
It follows that
$$\mathit{fire}(M')(v) = E(u, v) \otimes M'(u) =$$
$$= E(u, v) \otimes (M(u) \cdot p) = M(v) \cdot p = M'(v).$$
- Consider any node $v \in W \cap V_{min}$ and let $u = \pi(v)$. By Lemma 20, $u \in W$. If $M(v) = 0$, then $M'(v) = 0 \le \mathit{fire}(M')(v)$. Consider the case $M(v) > 0$. Since $v \in V_{min}$, π is a canonical trace of M and $\pi(v) = u$, we have
$$M(v) = (E(u, v) \Rightarrow M(u)) \le (E(w, v) \Rightarrow M(w))$$
for all $w \in V_{max}$, and $0 < (E(u, v) \Rightarrow M(u)) < L(v)$. Hence, $E(u, v) > M(u)$ and, for every $w \in V_{max}$, either $E(w, v) = 0$ or $M(u)/E(u, v) \le M(w)/E(w, v)$. Since $M(v) = (E(u, v) \Rightarrow M(u)) = M(u)/E(u, v)$ and $p \le L(v)/M(v)$, we have $M(u) \cdot p \le E(u, v) \cdot L(v) \le E(u, v)$. Hence,
$$(E(u, v) \Rightarrow M'(u)) = (E(u, v) \Rightarrow (M(u) \cdot p)) =$$
$$= (E(u, v) \Rightarrow M(u)) \cdot p = M(v) \cdot p \le L(v).$$
For every $w \in V_{max}$,
  - if $(E(w, v) \Rightarrow M(w)) > (E(u, v) \Rightarrow M(u))$, then, by (22),
$$(E(u, v) \Rightarrow M'(u)) = (E(u, v) \Rightarrow M(u)) \cdot p \le$$
$$\le (E(w, v) \Rightarrow M(w)) \le (E(w, v) \Rightarrow M'(w));$$
  - else $(E(w, v) \Rightarrow M(w)) = (E(u, v) \Rightarrow M(u))$, which implies $w \in W$ and
$$(E(u, v) \Rightarrow M'(u)) = (E(w, v) \Rightarrow M(w)) \cdot p =$$
$$= (E(w, v) \Rightarrow M(w) \cdot p) = (E(w, v) \Rightarrow M'(w)).$$

  Therefore,
$$\mathit{fire}(M')(v) = (E(u, v) \Rightarrow M'(u)) = M(v) \cdot p = M'(v).$$

We have shown that $M' \le \mathit{fire}(M')$. By Lemma 6, it follows that $M' \le M$, which contradicts the fact $M \lneq M'$. ■

Lemma 22: Assume that $\otimes$ is the product t-norm. Let $N = \langle V_{min}, V_{max}, E, L\rangle$ be a finite minimax net and π a function of type $V \to (V \cup \{null\})$, where $V = V_{min} \cup V_{max}$. There exists at most one stable marking M of N such that π is a canonical trace of M and, for every $v \in V$, $t\_rank(v, M) = \omega$ implies $M(v) = 0$. Checking the existence of such a marking and computing it, if it exists, can be done by the function $\mathsf{ComputeMarkingFromTrace}_P(N, \pi)$ given below, which runs in time $O(m + n)$, where $n = |V|$ and $m = |E|$.

Function 23: $\mathsf{ComputeMarkingFromTrace}_P(N, \pi)$

Input: a finite minimax net $N = \langle V_{min}, V_{max}, E, L\rangle$ and a function $\pi : V \to (V \cup \{null\})$, where $V = V_{min} \cup V_{max}$.

Output: the stable marking M of N (with respect to the product t-norm) such that π is a canonical trace of M and, for every $v \in V$, $t\_rank(v, M) = \omega$ implies $M(v) = 0$, if it exists, and *false* otherwise.

1. if there exists $v \in V_{max}$ such that
   $\pi(v) \neq null$ and $E(\pi(v), v) = 0$:
2. return *false*;
3. initialize $M : V \to [0, 1]$ by
   setting $M[v] = 0$ for all $v \in V$;
4. apply depth-first-search to identify the set $W \subseteq V$ consisting of nodes of all cycles of π and compute a bijection $\sigma : 1..|V - W| \to (V - W)$ such that, if $u, v \in V - W$ and $u = \pi(v)$, then $\sigma^{-1}(u) < \sigma^{-1}(v)$ (like topologically sorting the graph represented by π without the nodes from W);
5. for each i in $1..|V - W|$:
6. let $v = \sigma[i]$ and $u = \pi(v)$;
7. if $v \in V_{min}$ and $\pi(v) = null$:
8. $M[v] := L(v)$;
9. else if $v \in V_{min}$ and $u = \pi(v) \neq null$:
10. $M[v] := (E(u, v) \Rightarrow M[u])$;
11. else if $v \in V_{max}$ and $u = \pi(v) \neq null$:
12. $M[v] := E(u, v) \otimes M[u]$;
13. if M is not stable (check by using (13) and (14)):
14. return *false*;
15. return M;

Example 24: Consider the minimax net $N = \langle V_{min}, V_{max}, E, L\rangle$ specified in Fig. 2, the function $\pi : V \to (V \cup \{null\})$ specified in Example 18, where $V = V_{min} \cup V_{max}$, and the run of $\mathsf{ComputeMarkingFromTrace}_P(N, \pi)$. Executing the statement 4 of the function, we obtain $W = \{b, c, h, g\}$ and can assume that the obtained order σ is f, a, d, e, j, i. Executing the "for" loop of the function, we obtain the marking M specified in Example 2. As it is stable, it is returned as the result. ■

Proof of Lemma 22: Consider the above given function $\mathsf{ComputeMarkingFromTrace}_P(N, \pi)$. If there exists $v \in V_{max}$ such that $\pi(v) \neq null$ and $E(\pi(v), v) = 0$, then π cannot be a trace of any stable marking of N and the function returns *false*. Let M be the marking of N built by the statements 3–12 of the function. If M' is a stable marking of N such that π is a canonical trace of M' and, for every $v \in V$, $t\_rank(v, M') = \omega$ implies $M(v) = 0$, then it must be equal to M. Checking whether M is stable is done by checking the condition (13) for all $x \in V_{min}$ and the condition (14) for all $y \in V_{max}$.

Clearly, the statements 1–3 and the "for each" loop are executed in time $O(n)$. The statement 4 can also be executed in time $O(n)$ by using the standard depth-first-search techniques for detecting cycles and topologically sorting the graph represented by the function π without the cycles. Checking whether M is stable by the statement 13 can be done in time $O(m+n)$. Therefore, the function $\mathsf{ComputeMarkingFromTrace}_P(N, \pi)$ runs in time $O(m + n)$. ■

Algorithm 25 given below computes the greatest correct marking (with respect to the product t-norm) of a given finite minimax net $N = \langle V_{min}, V_{max}, E, L\rangle$. It initializes rs to the empty marking of N. Then, for each function $\pi : V \to (V \cup \{null\})$, where $V = V_{min} \cup V_{max}$, it extends rs with $M = \mathsf{ComputeMarkingFromTrace}_P(N, \pi)$ if M is a marking. At the end, the algorithm returns rs.

Algorithm 25: $\mathsf{ComputeTheGreatestCorrectMarking}_P$
Input: a finite minimax net $N = \langle V_{min}, V_{max}, E, L\rangle$.
Output: the greatest correct marking M of N (with respect to the product t-norm).

1. let $V = V_{min} \cup V_{max}$;
2. initialize $rs : V \to [0, 1]$ by setting $rs[v] = 0$ for all $v \in V$;
3. for each function $\pi : V \to (V \cup \{null\})$:
4. $M := \mathsf{ComputeMarkingFromTrace}_P(N, \pi)$;
5. if M is a marking (i.e., different from "*false*"):
6. $rs := rs \vee M$;
7. return rs;

Example 26: Consider the minimax net $N = \langle V_{min}, V_{max}, E, L\rangle$ specified in Fig. 2 and any function $\pi : V \to (V \cup \{null\})$ that is a trace of a stable marking M of N, where $V = V_{min} \cup V_{max}$. We must have $\pi(f) = \pi(d) = null$. On the other hand, none of $\pi(x)$ for $x \in \{b, h, j\}$ can be $null$. Furthermore, we must have $\pi(a) = f$, $\pi(c) = b$, $\pi(g) = h$, $\pi(h) = c$ and $\pi(j) = e$. If $\pi(b) = a$, then after using π to compute $M(x)$ according to Definition 14 for each x in the sequence f, a, b, c, h, g, it can be checked that M cannot be stable since it does not satisfy (13) for $x = b$. Hence, $\pi(b) = g$ and (b, g, h, c) is a cycle of π. By Proposition 41 (given in the Supplementary Material), it follows that $M(x) = 0$ for all $x \in \{b, g, h, c\}$. If $\pi(e) = j$, then (e, j) is a cycle of π and, by Proposition 41, $M(e) = M(j) = 0$, which contradicts the assumption that M is stable and satisfies (14) for $y = e$. Hence, $\pi(e) = d$. After using π to compute $M(x)$ according to Definition 14 for each x in the sequence d, e, j, it can be seen that, since M satisfies (14) for $y = i$, we must have $\pi(i) = j$. Thus, we have shown that π must be the same as the one specified in Example 18. Consequently, M must be as specified in Example 2 and it is the only stable marking of N. Executing Algorithm 25 for N, only in the iteration of the "for" loop with π specified as in Example 18, the statement 6 is executed; therefore that stable marking is returned as the greatest correct marking of N. ■

Theorem 27: Algorithm 25 is correct. It runs in time $2^{O(n \log n)}$, where n is the number of the nodes of the given minimax net.

Proof: Consider the execution of Algorithm 25 for a finite minimax net $N = \langle V_{min}, V_{max}, E, L\rangle$. Let M_0 be the greatest correct marking of N and π_0 a canonical trace of M_0. By Lemma 21, for every node v of N with $t\_rank(v, M_0) = \omega$, $M_0(v) = 0$. Due to the correctness of the function $\mathsf{ComputeMarkingFromTrace}_P$ (see Lemma 22), the "for each" loop of Algorithm 25 has the invariant that $rs \leq M_0$. For the iteration with $\pi = \pi_0$, by Lemma 22, the value returned by the call $\mathsf{ComputeMarkingFromTrace}_P(N, \pi)$ is identical to M_0, and after executing that iteration we have $M_0 \leq rs$. Therefore, the result rs returned by the algorithm is equal to M_0. That is, the algorithm is correct. Its complexity is $2^{O(n \log n)}$ because $(n + 1)^n \cdot (m + n)$ is within $2^{O(n \log n)}$. ■

We give below an algorithm for computing the greatest fuzzy bisimulation with respect to the product t-norm between two finite fuzzy graphs over the same signature.

Algorithm 28: ComputeTheGreatestFuzzyBisimulation$_P$

Input: finite fuzzy graphs $G = \langle V, E, L, \Sigma_V, \Sigma_E \rangle$ and $G' = \langle V', E', L', \Sigma_V, \Sigma_E \rangle$.

Output: the greatest fuzzy bisimulation with respect to the product t-norm between G and G'.

1. construct the minimax net $N = \langle V_{min}, V_{max}, E'', L'' \rangle$ that bisimulatedly corresponds to $\langle G, G' \rangle$;
2. execute Algorithm 25 for N to construct the greatest correct marking M of N;
3. return $M|_{V_{min}}$;

Let Algorithm 28′ be obtained from Algorithm 28 by replacing the word "bisimulatedly" with "simulatedly". It is an algorithm for computing the greatest fuzzy simulation with respect to the product t-norm between two finite fuzzy graphs over the same signature.

Theorem 29: Algorithms 28 and 28′ are correct. They run in time $2^{O(n^2 \log n)}$, where $n = |V| + |V'|$, assuming that the sizes of Σ_V and Σ_E are constants.

This theorem follows directly from Theorems 11, 13 and 27.

V. The Case of Using the Łukasiewicz t-Norm

In this section, assume that $\otimes$ is the Łukasiewicz t-norm. We study the problem of computing the greatest correct marking of a finite minimax net under this assumption. Proofs of the results of this section are presented in the Supplementary Material.

The following lemma is crucial for this section.

Lemma 30: Assume that $\otimes$ is the Łukasiewicz t-norm. Let $N = \langle V_{min}, V_{max}, E, L \rangle$ be a finite minimax net, M the greatest correct marking of N and let $V = V_{min} \cup V_{max}$. Then, for every $v \in V$ with $t\_rank(v, M) = \omega$, there exist a canonical trace π of M and $k \geq 0$ such that $u = \pi^k(v) \in V_{max}$ and $M(u) = 0$.

Corollary 31: Assume that $\otimes$ is the Łukasiewicz t-norm. Let $N = \langle V_{min}, V_{max}, E, L \rangle$ be a finite minimax net, M the greatest correct marking of N and let $V = V_{min} \cup V_{max}$. Then, there exists a canonical trace π of M such that, for every $v \in V$ with $t\_rank(v, M) = \omega$, there exists $k \geq 0$ such that $u = \pi^k(v) \in V_{max}$ and $M(u) = 0$.

The following lemma is a counterpart of Lemma 22.

Lemma 32: Assume that $\otimes$ is the Łukasiewicz t-norm. Let $N = \langle V_{min}, V_{max}, E, L \rangle$ be a finite minimax net, π a function of type $V \to (V \cup \{null\})$, where $V = V_{min} \cup V_{max}$, C the collection of all cycles of π and f a choice function on C such that $f(C) \subseteq V_{max}$. There exists at most one stable marking M of N such that π is a canonical trace of M and, for every $W \in C$, $M(f(W)) = 0$. Checking the existence of such a marking and computing it, if it exists, can be done by the function ComputeMarkingFromTrace$_L(N, \pi, C, f)$ *given below, which runs in time $O(m + n)$, where $n = |V|$ and $m = |E|$.*

Function 33: ComputeMarkingFromTrace$_L(N, \pi, C, f)$

Input: a finite minimax net $N = \langle V_{min}, V_{max}, E, L \rangle$, a function $\pi : V \to (V \cup \{null\})$, where $V = V_{min} \cup V_{max}$, the collection C of all cycles of π and a choice function f on C such that $f(C) \subseteq V_{max}$.

Output: the stable marking M of N (with respect to the Łukasiewicz t-norm) such that π is a canonical trace of M and, for every $W \in C$, $M(f(W)) = 0$, if it exists, and $false$ otherwise.

1. if there exists $v \in V_{max}$ such that
 $\pi(v) \neq null$ and $E(\pi(v), v) = 0$:
2. return $false$;
3. initialize $M : V \to [0, 1]$ by
 setting $M[v] = 0$ for all $v \in V$;
4. for each $W \in C$:
5. let $(v_0, \ldots, v_k)$ be the enumeration of W such that
 $v_k = f(W)$;
6. for each i from $k - 1$ to 0:
7. if $v_i \in V_{min}$:
8. $M[v_i] := (E(v_{i+1}, v_i) \Rightarrow M[v_{i+1}])$;
9. else:
10. $M[v_i] := E(v_{i+1}, v_i) \otimes M[v_{i+1}])$;
11. let $U = \bigcup C$;
12. apply depth-first-search to compute a bijection $\sigma : 1..|V - U| \to (V - U)$ such that, if $u, v \in V - U$ and $u = \pi(v)$, then $\sigma^{-1}(u) < \sigma^{-1}(v)$ (like topologically sorting the graph represented by π without the nodes from U);
13. for each i in $1..|V - U|$:
14. let $v = \sigma[i]$ and $u = \pi(v)$;
15. if $v \in V_{min}$ and $\pi(v) = null$:
16. $M[v] := L(v)$;
17. else if $v \in V_{min}$ and $u = \pi(v) \neq null$:
18. $M[v] := (E(u, v) \Rightarrow M[u])$;
19. else if $v \in V_{max}$ and $u = \pi(v) \neq null$:
20. $M[v] := E(u, v) \otimes M[u]$;
21. if M is not stable (check by using (13) and (14)):
22. return $false$;
23. return M;

Algorithm 34 given below computes the greatest correct marking (with respect to the Łukasiewicz t-norm) of a given finite minimax net $N = \langle V_{min}, V_{max}, E, L \rangle$. It initializes rs to the empty marking of N. Then, for each function $\pi : V \to (V \cup \{null\})$, where $V = V_{min} \cup V_{max}$, it computes the collection C of all cycles of π and then, for each choice function f on C such that $f(C) \subseteq V_{max}$, it extends rs with $M =$ ComputeMarkingFromTrace$_L(N, \pi, C, f)$ if M is a marking. At the end, the algorithm returns rs.

Algorithm 34: ComputeTheGreatestCorrectMarking$_L$

Input: a finite minimax net $N = \langle V_{min}, V_{max}, E, L \rangle$.

Output: the greatest correct marking M of N (with respect to the Łukasiewicz t-norm).

1. let $V = V_{min} \cup V_{max}$;
2. initialize $rs : V \to [0, 1]$ by setting $rs[v] = 0$ for all $v \in V$;
3. for each function $\pi : V \to (V \cup \{null\})$:
4. compute the collection C of all cycles of π;
5. for each choice function f on C such that $f(C) \subseteq V_{max}$:
6. $M :=$ ComputeMarkingFromTrace$_L(N, \pi, C, f)$;
7. if M is a marking (i.e., different from "*false*"):
8. $rs := rs \vee M$;

9. return rs;

Theorem 35: Algorithm 34 is correct. It runs in time $2^{O(n \log n)}$, where n is the number of the nodes of the given minimax net.

We give below an algorithm for computing the greatest fuzzy bisimulation with respect to the Łukasiewicz t-norm between two finite fuzzy graphs over the same signature.

Algorithm 36: $\mathsf{ComputeTheGreatestFuzzyBisimulation}_L$
Input: finite fuzzy graphs $G = \langle V, E, L, \Sigma_V, \Sigma_E \rangle$ and $G' = \langle V', E', L', \Sigma_V, \Sigma_E \rangle$.
Output: the greatest fuzzy bisimulation with respect to the Łukasiewicz t-norm between G and G'.

1. construct the minimax net $N = \langle V_{min}, V_{max}, E'', L'' \rangle$ that bisimulatedly corresponds to $\langle G, G' \rangle$;
2. execute Algorithm 34 for N to construct the greatest correct marking M of N;
3. return $M|_{V_{min}}$;

Let Algorithm 36′ be obtained from Algorithm 36 by replacing the word "bisimulatedly" with "simulatedly". It is an algorithm for computing the greatest fuzzy simulation with respect to the Łukasiewicz t-norm between two finite fuzzy graphs over the same signature.

Theorem 37: Algorithms 36 and 36′ are correct. They run in time $2^{O(n^2 \log n)}$, where $n = |V| + |V'|$, assuming that the sizes of Σ_V and Σ_E are constants.

This theorem follows directly from Theorems 11, 13 and 35.

VI. Concluding Remarks

We have introduced fuzzy minimax nets and used them to design the first algorithms for computing the greatest fuzzy bisimulation/simulation between two finite fuzzy labeled graphs in the case of using the product t-norm, as well as the first algorithms whose complexity order does not depend on the fuzzy values used by the input graphs for those problems in the case of using the Łukasiewicz t-norm. These algorithms can be restated for fuzzy automata, fuzzy labeled transition systems, fuzzy Kripke models, fuzzy social networks and fuzzy interpretations in description logic.

There are some similarities between fuzzy minimax nets and other structures like neuron networks, Petri nets and game trees, but the notion of a fuzzy minimax net is novel and essentially different from the others. We hope that fuzzy minimax nets may have other applications than just for computing fuzzy bisimulations/simulations.

Our algorithms run in exponential time. Some optimizations for them are presented in the Supplementary Material. There remains a challenging open problem of whether there exist PTIME algorithms for computing the greatest fuzzy bisimulation/simulation between two finite fuzzy labeled graphs in the case of using the product or Łukasiewicz t-norm. We will try to deal with it as future work.

References

[1] M. Ćirić, J. Ignjatović, N. Damljanović, and M. Bašic, "Bisimulations for fuzzy automata," *Fuzzy Sets and Systems*, vol. 186, no. 1, pp. 100–139, 2012.
[2] M. Ćirić, J. Ignjatović, I. Jančić, and N. Damljanović, "Computation of the greatest simulations and bisimulations between fuzzy automata," *Fuzzy Sets and Systems*, vol. 208, pp. 22–42, 2012.
[3] N. Damljanović, M. Ćirić, and J. Ignjatović, "Bisimulations for weighted automata over an additively idempotent semiring," *Theor. Comput. Sci.*, vol. 534, pp. 86–100, 2014.
[4] C. Yang and Y. Li, "Approximate bisimulations and state reduction of fuzzy automata under fuzzy similarity measures," *Fuzzy Sets Syst.*, vol. 391, pp. 72–95, 2020.
[5] S. Stanimirović, I. Micić, and M. Ćirić, "Approximate bisimulations for fuzzy automata over complete Heyting algebras," *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 2, pp. 437–447, 2022.
[6] Y. Cao, G. Chen, and E. Kerre, "Bisimulations for fuzzy-transition systems," *IEEE Trans. Fuzzy Systems*, vol. 19, no. 3, pp. 540–552, 2011.
[7] Y. Cao, S. Sun, H. Wang, and G. Chen, "A behavioral distance for fuzzy-transition systems," *IEEE Trans. Fuzzy Systems*, vol. 21, no. 4, pp. 735–747, 2013.
[8] H. Pan, Y. Cao, M. Zhang, and Y. Chen, "Simulation for lattice-valued doubly labeled transition systems," *Int. J. Approx. Reason.*, vol. 55, no. 3, pp. 797–811, 2014.
[9] H. Pan, Y. Li, and Y. Cao, "Lattice-valued simulations for quantitative transition systems," *Int. J. Approx. Reason.*, vol. 56, pp. 28–42, 2015.
[10] H. Wu and Y. Deng, "Logical characterizations of simulation and bisimulation for fuzzy transition systems," *Fuzzy Sets and Systems*, vol. 301, pp. 19–36, 2016.
[11] H. Wu, Y. Chen, T. Bu, and Y. Deng, "Algorithmic and logical characterizations of bisimulations for non-deterministic fuzzy transition systems," *Fuzzy Sets and Systems*, vol. 333, pp. 106–123, 2018.
[12] L. Nguyen, "Characterizing fuzzy simulations for fuzzy labeled transition systems in fuzzy propositional dynamic logic," *Int. J. Approx. Reason.*, vol. 135, pp. 21–37, 2021.
[13] P. Eleftheriou, C. Koutras, and C. Nomikos, "Notions of bisimulation for Heyting-valued modal languages," *J. Log. Comput.*, vol. 22, no. 2, pp. 213–235, 2012.
[14] T.-F. Fan, "Fuzzy bisimulation for Gödel modal logic," *IEEE Trans. Fuzzy Systems*, vol. 23, no. 6, pp. 2387–2396, 2015.
[15] D. Diaconescu, "Modal equivalence and bisimilarity in many-valued modal logics with many-valued accessibility relations," *Fundam. Informaticae*, vol. 173, no. 2-3, pp. 177–189, 2020.
[16] L. Nguyen, "Logical characterizations of fuzzy bisimulations in fuzzy modal logics over residuated lattices," *Fuzzy Sets Syst.*, vol. 431, pp. 70–93, 2022.
[17] T. Fan and C. Liau, "Logical characterizations of regular equivalence in weighted social networks," *Artif. Intell.*, vol. 214, pp. 66–88, 2014.
[18] J. Ignjatović, M. Ćirić, and I. Stanković, "Bisimulations in fuzzy social network analysis," in *Proceedings of IFSA-EUSFLAT-15*. Atlantis Press, 2015.
[19] L. Nguyen, "Bisimilarity in fuzzy description logics under the Zadeh semantics," *IEEE Trans. Fuzzy Systems*, vol. 27, no. 6, pp. 1151–1161, 2019.
[20] L. Nguyen, Q. Ha, N. Nguyen, T. Nguyen, and T. Tran, "Bisimulation and bisimilarity for fuzzy description logics under the Gödel semantics," *Fuzzy Sets and Systems*, vol. 388, pp. 146–178, 2020.
[21] L. Nguyen and D. Tran, "Computing fuzzy bisimulations for fuzzy structures under the Gödel semantics," *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 7, pp. 1715–1724, 2021.
[22] L. Nguyen, "Computing the fuzzy partition corresponding to the greatest fuzzy auto-bisimulation of a fuzzy graph-based structure," *CoRR*, vol. abs/2107.08542, 2021.
[23] I. Micić, Z. Jančić, and S. Stanimirović, "Computation of the greatest right and left invariant fuzzy quasi-orders and fuzzy equivalences," *Fuzzy Sets and Systems*, vol. 339, pp. 99–118, 2018.
[24] L. Nguyen and D. Tran, "Computing crisp bisimulations for fuzzy structures," *CoRR*, vol. abs/2010.15671, 2020.
[25] L. Nguyen, "Computing crisp simulations for fuzzy labeled transition systems," *J. Intell. Fuzzy Syst.*, vol. 42, no. 4, pp. 3067–3078, 2022.
[26] S. Stanimirović, A. Stamenković, and M. Ćirić, "Improved algorithms for computing the greatest right and left invariant boolean matrices and their application," *Filomat*, vol. 33, no. 9, pp. 2809—-2831, 2019.

Supplementary Material

In this supplementary material, we provide proofs of the results of Section V, optimizations for Algorithms 25 and 34, as well as some additional results.

A. Proofs of the Results of Section V

Proof of Lemma 30: By Proposition 5, $\mathit{fire}(M) = M$ and M is the greatest fixpoint of fire. Let W be the set of all nodes $v \in V$ such that $t\_rank(v, M) = \omega$ and, for every canonical trace π of M and every node $u = \pi^k(v) \in V_{max}$ with $k \geq 0$, $M(u) > 0$. We need to prove that $W = \emptyset$. For a contradiction, suppose $W \neq \emptyset$. Let p be the greatest real number such that, for every min-node $v \in W$,

- $p \leq L(v) - M(v)$,
- if $w \in V_{max}$, π is a canonical trace of M and
$$(E(w, v) \Rightarrow M(w)) > (E(\pi(v), v) \Rightarrow M(\pi(v))),$$
then
$$p \leq (E(w, v) \Rightarrow M(w)) - (E(\pi(v), v) \Rightarrow M(\pi(v))). \quad (23)$$

Observe that $W \cap V_{min} \neq \emptyset$ and, for every $v \in W \cap V_{min}$, $M(v) < L(v)$. The value p is well-defined due to this reason and the assumption that N is finite. Moreover, $p > 0$. Let $M' \in \mathcal{M}_N$ be the marking specified as follows: if $v \in W$, then $M'(v) = M(v) + p$, else $M'(v) = M(v)$. Clearly, $M \lneq M'$. We show that $M' \leq \mathit{fire}(M')$.

- For $v \in V - W$, since $M' \gneq M$, by (1) and (2),
$$\mathit{fire}(M')(v) \geq \mathit{fire}(M)(v) = M(v) = M'(v).$$
- Consider any node $v \in W \cap V_{max}$. Let π be a canonical trace of M and let $u = \pi(v)$. Thus, $u \in W$ and
$$0 < M(v) = E(u, v) \otimes M(u) = E(u, v) + M(u) - 1.$$
Since $v \in V_{max}$, π is a canonical trace of M and $\pi(v) = u$, we have $E(u, v) \otimes M(u) \geq E(w, v) \otimes M(w)$ for all $w \in V_{min}$. Hence, for every $w \in V_{min}$,
$$E(u, v) + M(u) \geq E(w, v) + M(w),$$
$$E(u, v) \otimes M'(u) = E(u, v) \otimes (M(u) + p) \geq$$
$$\geq E(w, v) \otimes (M(w) + p) \geq E(w, v) \otimes M'(w).$$
It follows that
$$\mathit{fire}(M')(v) = E(u, v) \otimes M'(u) =$$
$$= E(u, v) \otimes (M(u) + p) = M(v) + p = M'(v).$$
- Consider any node $v \in W \cap V_{min}$. Let π be a canonical trace of M and let $u = \pi(v)$. Thus, $u \in W$. Since $v \in W \cap V_{min}$, π is a canonical trace of M and $\pi(v) = u$, we have $(E(u, v) \Rightarrow M(u)) \leq (E(w, v) \Rightarrow M(w))$ for all $w \in V_{max}$, and $(E(u, v) \Rightarrow M(u)) < L(v)$. Hence, $E(u, v) > M(u)$. Since $M(v) = (E(u, v) \Rightarrow M(u)) = M(u) - E(u, v) + 1$ and $p \leq L(v) - M(v)$, we have $M(u) + p \leq E(u, v) + L(v) - 1 \leq E(u, v)$. Hence,
$$(E(u, v) \Rightarrow M'(u)) = (E(u, v) \Rightarrow (M(u) + p)) =$$
$$= (E(u, v) \Rightarrow M(u)) + p = M(v) + p \leq L(v).$$
For every $w \in V_{max}$,
  - if $(E(w, v) \Rightarrow M(w)) > (E(u, v) \Rightarrow M(u))$, then, by (23),
$$(E(u, v) \Rightarrow M'(u)) = (E(u, v) \Rightarrow M(u)) + p \leq$$
$$\leq (E(w, v) \Rightarrow M(w)) \leq (E(w, v) \Rightarrow M'(w)),$$
  - else $(E(w, v) \Rightarrow M(w)) = (E(u, v) \Rightarrow M(u))$, which implies $w \in W$ and
$$(E(u, v) \Rightarrow M'(u)) = (E(w, v) \Rightarrow M(w)) + p =$$
$$= (E(w, v) \Rightarrow (M(w)+p)) = (E(w, v) \Rightarrow M'(w)).$$
Therefore,
$$\mathit{fire}(M')(v) = (E(u, v) \Rightarrow M'(u)) = M(v)+p = M'(v).$$

We have shown that $M' \leq \mathit{fire}(M')$. By Lemma 6, it follows that $M' \leq M$, which contradicts the fact $M \lneq M'$. ∎

Proof of Corollary 31: By Lemma 30, for every $v \in V$ with $t\_rank(v, M) = \omega$, there exist a canonical trace π_v of M and $k \geq 0$ such that $u = \pi_v^k(v) \in V_{max}$ and $M(u) = 0$. For $v \in V$ with $t\_rank(v, M) \in \mathbb{N}$, let π_v be an arbitrary canonical trace of M. Let π be the trace of M specified by $\pi(v) = \pi_v(v)$ for $v \in V$. Clearly, π is a canonical trace of M and, for every $v \in V$ with $t\_rank(v, M) = \omega$, there exists $k \geq 0$ such that $u = \pi^k(v) \in V_{max}$ and $M(u) = 0$. ∎

Proof of Lemma 32: Consider the above given function $\mathsf{ComputeMarkingFromTrace}_L(N, \pi, C, f)$. If there exists $v \in V_{max}$ such that $\pi(v) \neq \mathit{null}$ and $E(\pi(v), v) = 0$, then π cannot be a trace of any stable marking of N and the function returns *false*. Let M be the marking of N built by the statements 3–20 of the function. If M' is a stable marking of N such that π is a canonical trace of M' and, for every $W \in C$, $M'(f(W)) = 0$, then it must be equal to M. Checking whether M is stable is done by checking the condition (13) for all $x \in V_{min}$ and the condition (14) for all $y \in V_{max}$.

Clearly, the statements 1–11 and 13–20 can be executed in time $O(n)$ (recall that the cycles are pairwise disjoint). The statement 12 can also be executed in time $O(n)$ by using the standard depth-first-search techniques for topologically sorting the graph represented by the function π without the cycles. Checking whether M is stable by the statement 21 can be done in time $O(m + n)$. Therefore, the function $\mathsf{ComputeMarkingFromTrace}_L(N, \pi, C, f)$ runs in time $O(m + n)$. ∎

Proof of Theorem 35: Consider the execution of Algorithm 34 for a finite minimax net $N = \langle V_{min}, V_{max}, E, L \rangle$. Let M_0 be the greatest correct marking of N. By Corollary 31, there exists a canonical trace π_0 of M_0 such that, for every $v \in V$ with $t\_rank(v, M_0) = \omega$, there exists $k \geq 0$ such that $u = \pi_0^k(v) \in V_{max}$ and $M_0(u) = 0$. Let C_0 be the collection of all cycles of π_0. There exists a choice function f_0 on C_0 such that $f_0(C_0) \subseteq V_{max}$ and, for every $W \in C_0$, $M_0(f_0(W)) = 0$.

Due to the correctness of the function $\mathsf{ComputeMarkingFromTrace}_L$ (see Lemma 32), both the "for each" loops of Algorithm 34 have the invariant that $rs \leq M_0$. For the iteration of the outer "for each" loop with $\pi = \pi_0$ and the iteration of the inner "for each" loop with $f = f_0$, by Lemma 32, the value returned by the call $\mathsf{ComputeMarkingFromTrace}_L(N, \pi, C, f)$ is identical to M_0, and hence, after executing that inner iteration, $M_0 \leq rs$. Therefore, the result rs returned by the algorithm is equal to M_0. That is, the algorithm is correct. Its complexity is $2^{O(n \log n)}$ because $(n+1)^n \cdot n^n \cdot (m+n)$ is within $2^{O(n \log n)}$. ■

B. Optimizations

In this section, we present two optimizations for Algorithms 25 and 34 (which compute the greatest correct marking of a finite minimax net in the case of using the product or Łukasiewicz t-norm). This results in Algorithm 38.

The first optimization relies on dividing the input minimax net N into nearly independent components and applying the task of computing the greatest correct marking for each of the components in an appropriate order. In particular, we first compute the strong directed components of the graph representing N and sort them topologically. Then, for each strong directed component V_i in the resulted order, we compute the greatest correct marking of the minimax net N restricted to V_i in the way that only marking values $M[x]$ for $x \in V_i$ are updated, while marking values $M[y]$ for positive edges $\langle y, x\rangle$ of N with $y \notin V_i$ are used when updating $M[x]$.

As the second optimization, when computing the greatest correct marking of the minimax net N restricted to a strong directed component V_i, we first check whether such a fixpoint can be reached just by applying the operation *fire* a number of times to the whole component V_i. The number of repetitions of applying *fire* to V_i is set to $|V_i|+1$. Only when the stability on V_i cannot be reached in that way, we apply Algorithm 25 if $\otimes$ is the product t-norm or Algorithm 34 if $\otimes$ is the Łukasiewicz t-norm to the minimax net N restricted to V_i in the way mentioned in the previous paragraph.

Algorithm 38 is formally specified below.

Algorithm 38: $\mathsf{ComputeTheGreatestCorrectMarking}_{PL}$
Input: a finite minimax net $N = \langle V_{min}, V_{max}, E, L\rangle$.
Output: the greatest correct marking M of N for the case where $\otimes$ is the product or Łukasiewicz t-norm.

1. let $G = \langle V, E'\rangle$ be the directed graph with $V = V_{min} \cup V_{max}$ and $E' = \{\langle x, y\rangle \mid \langle x, y\rangle$ is a positive edge of $N\}$;
2. find strong directed components of G, topologically sort the condensation of G and let $V_1, \ldots, V_k$ be the strong directed components of G in the resulted order;
3. initialize M by setting $M[x] := L(x)$ for all $x \in V_{min}$;
4. for each i from 1 to k:
5.   repeat $|V_i| + 1$ times:
6. $changed := false$;
7.   for each $x \in V_i$:
8.   if $x \in V_{min}$:
9. $a := L(x) \wedge \bigwedge\{E(z, x) \Rightarrow M[z] \mid z \in V_{max}\}$;
10.   else:
11. $a := \bigvee\{E(z, y) \otimes M[z] \mid z \in V_{min}\}$;
12.   if $a \neq M(x)$:
13. $M[x] := a$;
14. $changed := true$;
15.   if not $changed$:
16.   break;
17.   if $changed$:
18.   apply Algorithm 25 if $\otimes$ is the product t-norm or Algorithm 34 if $\otimes$ is the Łukasiewicz t-norm to the minimax net N restricted to V_i in the way that only values $M[x]$ for $x \in V_i$ are updated, while values $M[y]$ for positive edges $\langle y, x\rangle$ of N with $y \notin V_i$ are used when updating $M[x]$;

Example 39: Applying Algorithm 38 to the minimax net N specified in Fig. 2, we can assume that the topological order of the strong directed components of the graph representing the net is: $V_1 = \{f\}$, $V_2 = \{a\}$, $V_3 = \{b, c, h, g\}$, $V_4 = \{d\}$, $V_5 = \{e, j\}$, $V_6 = \{i\}$. When $\otimes$ is the product t-norm, repeatedly applying *fire* to V_3 or applying *fire* to V_5 only a few times does not allow us to reach a fixpoint. So, in this case, the second kind of optimization does not help. However, the first kind of optimization already speeds up the computation significantly. ■

C. Additional Results

Let $N = \langle V_{min}, V_{max}, E, L\rangle$ be a minimax net and π a function from V to $V \cup \{null\}$, where $V = V_{min} \cup V_{max}$. A cycle W of π is said to be *all-zeros-marked* (respectively, *some-zeros-marked*) with respective to a marking M of N if $M(v) = 0$ for all (respectively, some) $v \in W$.

Definition 40: Let $N = \langle V_{min}, V_{max}, E, L\rangle$ be a coimage-finite minimax net, M a stable marking of N and π a trace of M. Let $(v_0, \ldots, v_{2k-1})$ be a cycle of π, where v_i is a min-node (respectively, max-node) if i is even (respectively, odd), for $0 \leq i < 2k$. Let v_{2k} denote v_0.

The *P-weight* of the cycle $(v_0, \ldots, v_{2k-1})$ is defined to be the value of the following expression:[1]

$$\prod_{k \geq i \geq 1} (E(v_{2i}, v_{2i-1})/E(v_{2i-1}, v_{2i-2})).$$

The *Ł-weight* of the cycle $(v_0, \ldots, v_{2k-1})$ is defined to be the value of the following expression:

$$\sum_{k \geq i \geq 1} (E(v_{2i}, v_{2i-1}) - E(v_{2i-1}, v_{2i-2})).$$

Proposition 41: Assume that $\otimes$ is the product t-norm. Let N, M, π and $(v_0, \ldots, v_{2k-1})$ be as in Definition 40. Then, either the cycle $(v_0, \ldots, v_{2k-1})$ has the P-weight 1 or it is all-zeros-marked with respect to M.

Proof: Let v_{2k} denote v_0. For $k \geq i \geq 1$, since v_{2i} is a min-node, v_{2i-1} is a max-node and $(v_0, \ldots, v_{2k-1})$ is a cycle of π, we have

$$M(v_{2i-2}) = (E(v_{2i-1}, v_{2i-2}) \Rightarrow M(v_{2i-1})) =$$

[1] Since $(v_0, \ldots, v_{2k-1})$ is a cycle of π, for every $1 \leq i \leq k$, we have $E(v_{2i-1}, v_{2i-2}) > 0$.

$$= (E(v_{2i-1}, v_{2i-2}) \Rightarrow (E(v_{2i}, v_{2i-1}) \otimes M(v_{2i}))),$$

which implies

$$M(v_{2i-2}) = M(v_{2i}) \cdot (E(v_{2i}, v_{2i-1})/E(v_{2i-1}, v_{2i-2})),$$

since v_{2i-2} is a min-node and $M(v_{2i-2}) < L(v_{2i-2}) \leq 1$. Therefore,

$$M(v_0) = M(v_{2k}) \cdot \prod_{k \geq i \geq 1} (E(v_{2i}, v_{2i-1})/E(v_{2i-1}, v_{2i-2})).$$

Since $v_{2k} = v_0$, we must have that either $M(v_0) = 0$ or the P-weight of the cycle $(v_0, \ldots, v_{2k-1})$ is equal to 1. If $M(v_0) = 0$, then $M(v_j) = 0$ for all $2k > j \geq 1$. This completes the proof. ■

Proposition 42: Assume that $\otimes$ *is the Łukasiewicz t-norm. Let* N, M, π *and* $(v_0, \ldots, v_{2k-1})$ *be as in Definition 40. Then, either the cycle* $(v_0, \ldots, v_{2k-1})$ *has the Ł-weight 0 or it is some-zeros-marked with respect to* M.

Proof: Assume that the cycle $(v_0, \ldots, v_{2k-1})$ is not some-zeros-marked with respect to M, which means $M(v_i) > 0$ for all $0 \leq i < 2k$. We show that its Ł-weight is equal to 0. Let v_{2k} denote v_0. For $k \geq i \geq 1$, since v_{2i} is a min-node, v_{2i-1} is a max-node and $(v_0, \ldots, v_{2k-1})$ is a cycle of π, we have

$$M(v_{2i-2}) = (E(v_{2i-1}, v_{2i-2}) \Rightarrow M(v_{2i-1})) =$$
$$= (E(v_{2i-1}, v_{2i-2}) \Rightarrow (E(v_{2i}, v_{2i-1}) \otimes M(v_{2i}))),$$

which implies

$$\begin{aligned} M(v_{2i-2}) &= \\ &= (E(v_{2i}, v_{2i-1}) \otimes M(v_{2i})) + 1 - E(v_{2i-1}, v_{2i-2}) \\ &= (M(v_{2i}) + E(v_{2i}, v_{2i-1}) - 1) + 1 - E(v_{2i-1}, v_{2i-2}) \\ &= M(v_{2i}) + (E(v_{2i}, v_{2i-1}) - E(v_{2i-1}, v_{2i-2})), \end{aligned}$$

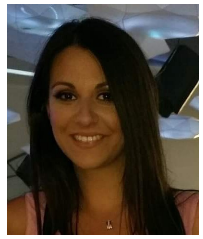
since v_{2i-2} is a min-node and $M(v_{2i-2}) < L(v_{2i-2}) \leq 1$. Therefore,

$$M(v_0) = M(v_{2k}) + \sum_{k \geq i \geq 1} (E(v_{2i}, v_{2i-1}) - E(v_{2i-1}, v_{2i-2})).$$

Since $v_{2k} = v_0$, the Ł-weight of the cycle $(v_0, \ldots, v_{2k-1})$ must be 0. ■

Linh Anh Nguyen received the M.Sc., Ph.D., and Dr.Sc. (Habilitation) degrees in computer science from the University of Warsaw, Warsaw, Poland, in 1997, 2000, and 2009, respectively.

He is currently an Associate Professor with the Institute of Informatics, University of Warsaw, as well as a researcher at the Faculty of Information Technology, Nguyen Tat Thanh University. His current research interests include simulations and bisimulations for fuzzy systems, automated reasoning and rule-based languages in modal/description logics.



Ivana Micić was born in 1984 in Niš, where she has finished her studies in the fields of mathematics and informatics in 2008, and obtained the Ph.D. degree in computer science in 2014 at the Faculty of Sciences and Mathematics, University of Niš, Serbia.

She is currently an Associate Professor of computer science at the same faculty. Her research interests include several branches of mathematics and computer science, such as mathematical logic, fuzzy relations and fuzzy relation equations, automata and formal languages, social networks, etc. She has also been working at the high school "Svetozar Markovic" in Niš, at the department for talented students of physics and mathematics. She is an author or co-author of more than thirty publications, where fourteen of them are articles in international mathematical and computer science journals with IF.

Stefan Stanimirović was born in 1989. He obtained the M.Sc. and Ph.D. degrees in computer science from the University of Niš, Faculty of Science and Mathematics, Serbia, in 2013 and 2019, respectively. In 2020, he has received the Annual Award of the Mathematical Institute of the Serbian Academy of Sciences and Arts in the field of computing for Ph.D. students.

He is currently an Assistant Professor at the same university. As an author and co-author, he has published more than fifteen publications, most of them in international journals with impact factor. His research interests include fuzzy automata, theory of computing, weighted logic, etc.