

Are Fuzzy Description Logics with General Concept Inclusion Axioms Decidable?

Franz Baader and Rafael Peñaloza
Theoretical Computer Science, TU Dresden
Email: {baader, penaloza}@tcs.inf.tu-dresden.de

Abstract—This paper concentrates on a fuzzy Description Logic with product t-norm and involutive negation. It does not answer the question posed in its title for this logic, but it gives strong indications that the answer might in fact be “no.” On the one hand, it shows that an algorithm that was claimed to answer the question affirmatively for this logic is actually incorrect. On the other hand, it proves undecidability of a variant of this logic.

Index Terms—Fuzzy Description Logics, Undecidability, Reasoning

I. INTRODUCTION

Description logics (DLs) [1] are a well-investigated family of logic-based knowledge representation formalisms, which can be used to model a given application domain using terminological axioms and assertional axioms. On the terminological side, modern DL systems provide their users with so-called concept inclusion axioms. The availability of these axioms increases the complexity of reasoning, both from a complexity theoretic and a practical point of view. For example, for the standard DL $\mathcal{ALC}$, the complexity rises from PSpace to ExpTime if these axioms are added. In addition, the tableau-based reasoning procedure used by most DL systems run into a termination problem: one has to add so-called blocking conditions in order to ensure their termination [2].

Fuzzy variants of DLs were introduced in order to deal with applications where precise definitions are not possible [3]. Decidability of fuzzy DLs is often shown by adapting the tableau-based algorithms for the corresponding crisp DL to the fuzzy case. This was first done for the case of DLs without concept inclusion axioms [4], [5] or with only *crisp* concept inclusion axioms [6], [7], but then also extended to DLs with concept inclusion axioms [8], [9]. In particular, Bobillo and Straccia [10], [11] have developed such an algorithm for a DL with product t-norm and involutive negation. Whereas [11] focuses only on so-called acyclic TBoxes, the algorithm from [10] allows for general concept inclusion axioms. More precisely, this algorithm is supposed to check whether an ontology expressed in this DL has a so-called witnessed model. Actually, the proof of correctness of this algorithm implies that, whenever such an ontology has a witnessed model, then it has a finite model. However, it was recently shown in [12] that this is not the case in the presence of general concept inclusion axioms, i.e., there is an ontology written in this logic that has a witnessed model, but does not have a finite model. Of course, this does not automatically imply that the algorithm itself is wrong. In fact, if one applies the algorithm from [10] to the

ontology used in [12] to demonstrate the failure of the finite model property, then one obtains the correct answer.

In this paper, we will first show that the algorithm is indeed incorrect by exhibiting an example ontology on which it gives an incorrect answer. This shows that the question of whether (witnessed) consistency of this logic is decidable or not is an open question. The second result shown in this paper gives an indication that the answer could in fact be “no.” We prove undecidability for a slightly modified problem where (i) we allow not only $\geq$ but also $>$ in our fuzzy concept inclusion axioms, and (ii) consider a somewhat weaker notion of witnessed model.

II. PRELIMINARIES

We introduce the logic *fuzzy $\mathcal{ALC}$ with product t-norm and involutive negation* and the tableau-based algorithm for this logic presented in [10], and then give an example that shows that this algorithm is not correct.

A. Fuzzy $\mathcal{ALC}$ with Product t-norm and involutive negation

The syntax of fuzzy $\mathcal{ALC}$ concepts is the same as for crisp $\mathcal{ALC}$. From two disjoint sets $\mathcal{N}_C$ and $\mathcal{N}_R$ of *concept-* and *role-*names, respectively, fuzzy $\mathcal{ALC}$ concepts are built using the syntax rule:

$$C ::= \top \mid \perp \mid A \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \neg C \mid \exists r.C \mid \forall r.C,$$

where $A \in \mathcal{N}_C$ and $r \in \mathcal{N}_R$.

The syntax of the axioms in this logic is slightly different from the crisp case since they need to state a degree of truth with which they hold. A *fuzzy $\mathcal{ALC}$ ABox* is a finite set of *concept assertion axioms* of the form $\langle a : C \geq q \rangle$ and *role assertion axioms* $\langle (a, b) : r \geq q \rangle$, where C is a fuzzy $\mathcal{ALC}$ concept, $r \in \mathcal{N}_R$, q is a rational number in $[0, 1]$, and $a, b \in \mathcal{N}_I$, with $\mathcal{N}_I$ the set of *individual names*. A *fuzzy $\mathcal{ALC}$ TBox* is a finite set of *concept inclusion axioms* of the form $\langle C \sqsubseteq D \geq q \rangle$, where C, D are $\mathcal{ALC}$ concepts and q is a rational number in $[0, 1]$. A *fuzzy ontology* $\mathcal{O} = (\mathcal{A}, \mathcal{T})$ consists of a fuzzy ABox $\mathcal{A}$ together with a fuzzy TBox $\mathcal{T}$.

The semantics of this logic extends the classical semantics of $\mathcal{ALC}$ by interpreting concepts and roles as fuzzy sets over an interpretation domain. Given a non-empty set Δ (the domain), a *fuzzy set* over Δ is a function $F : \Delta \rightarrow [0, 1]$, with the intuition that an element $\delta \in \Delta$ belongs to F with *degree* $F(\delta)$. The semantics of the different concept constructors depends on the class of fuzzy operators chosen to interpret them. In this

paper, we use the binary operators *product t-norm* $\otimes$, *product t-conorm* $\oplus$, and *residuum* $\rightarrow$, and the unary operator *involutive negation* $\ominus$ over the interval $[0, 1]$. These operators are defined as follows, for every $\alpha, \beta \in [0, 1]$:

$$\begin{aligned}\alpha \otimes \beta &:= \alpha \cdot \beta, \\ \alpha \oplus \beta &:= \alpha + \beta - \alpha \cdot \beta, \\ \alpha \rightarrow \beta &:= \begin{cases} 1 & \text{if } \alpha \leq \beta \\ \beta/\alpha & \text{otherwise,} \end{cases} \\ \ominus \alpha &:= 1 - \alpha.\end{aligned}$$

It is useful to notice that, for every $\alpha, \beta, q \in [0, 1]$, we have $\alpha \rightarrow \beta \geq q$ iff $\beta \geq q \cdot \alpha$. In particular, this means that (i) $\alpha \rightarrow \beta \geq 1$ iff $\beta \geq \alpha$ and (ii) $1 \rightarrow \beta = \beta$. We will often make use of these properties in the rest of this paper.

The semantics of fuzzy $\mathcal{ALC}$ is based on the notion of an *interpretation*, which is a tuple $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ where $\Delta^{\mathcal{I}}$ is a non-empty set, called the *domain*, and the function $\cdot^{\mathcal{I}}$ maps each individual name a to an element of $\Delta^{\mathcal{I}}$, each concept name A to a function $A^{\mathcal{I}} : \Delta^{\mathcal{I}} \rightarrow [0, 1]$ and each role name r to a function $r^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \rightarrow [0, 1]$. The interpretation function is extended to arbitrary fuzzy $\mathcal{ALC}$ concepts as follows. For every $\delta \in \Delta^{\mathcal{I}}$,

$$\begin{aligned}\top^{\mathcal{I}}(\delta) &= 1 \\ \perp^{\mathcal{I}}(\delta) &= 0 \\ (C_1 \sqcap C_2)^{\mathcal{I}}(\delta) &= C_1^{\mathcal{I}}(\delta) \cdot C_2^{\mathcal{I}}(\delta) \\ (C_1 \sqcup C_2)^{\mathcal{I}}(\delta) &= C_1^{\mathcal{I}}(\delta) + C_2^{\mathcal{I}}(\delta) - C_1^{\mathcal{I}}(\delta) \cdot C_2^{\mathcal{I}}(\delta) \\ (\neg C)^{\mathcal{I}}(\delta) &= 1 - C^{\mathcal{I}}(\delta) \\ (\exists r.C)^{\mathcal{I}}(\delta) &= \sup_{\gamma \in \Delta^{\mathcal{I}}} (r^{\mathcal{I}}(\delta, \gamma) \cdot C^{\mathcal{I}}(\gamma)) \\ (\forall r.C)^{\mathcal{I}}(\delta) &= \inf_{\gamma \in \Delta^{\mathcal{I}}} (r^{\mathcal{I}}(\delta, \gamma) \rightarrow C^{\mathcal{I}}(\gamma)).\end{aligned}$$

The interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ satisfies the concept assertion $\langle a : C \geq q \rangle$ iff $C^{\mathcal{I}}(a^{\mathcal{I}}) \geq q$, it satisfies the role assertion $\langle (a, b) : r \geq q \rangle$ iff $r^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}}) \geq q$, and it satisfies the concept inclusion $\langle C \sqsubseteq D \geq q \rangle$ iff $\inf_{\delta \in \Delta^{\mathcal{I}}} (C^{\mathcal{I}}(\delta) \rightarrow D^{\mathcal{I}}(\delta)) \geq q$. This interpretation is called a *model* of the ontology $\mathcal{O}$ if it satisfies all the axioms in $\mathcal{O}$.

An important notion in fuzzy DLs is that of *witnessed models* [13], [14]. A model is called *witnessed* if for every $\delta \in \Delta^{\mathcal{I}}$, role r and concepts C, D there exist $\gamma, \gamma' \in \Delta^{\mathcal{I}}$ such that

- $(\exists r.C)^{\mathcal{I}}(\delta) = r^{\mathcal{I}}(\delta, \gamma) \cdot C^{\mathcal{I}}(\gamma)$ and
- $(\forall r.C)^{\mathcal{I}}(\delta) = r^{\mathcal{I}}(\delta, \gamma') \rightarrow C^{\mathcal{I}}(\gamma')$.

Notice that, in the literature (e.g. [11], [12], [15]), witnessed models are usually required to satisfy additionally that, for every two concepts C, D , there is a γ such that

$$\inf_{\eta \in \Delta^{\mathcal{I}}} (C^{\mathcal{I}}(\eta) \rightarrow D^{\mathcal{I}}(\eta)) = C^{\mathcal{I}}(\gamma) \rightarrow D^{\mathcal{I}}(\gamma).$$

We call those witnessed models that satisfy this additional restriction *strongly witnessed* models.

The reason why we do not include this condition is that, in our opinion, it does not capture the spirit of fuzzy concept

inclusions: the axiom $\langle C \sqsubseteq D \geq q \rangle$ is meant to constrain the interpretation of concepts and roles such that, for each individual $\delta \in \Delta$, we have $C^{\mathcal{I}}(\delta) \rightarrow D^{\mathcal{I}}(\delta) \geq q$. With this point of view, it is not really necessary that the infimum of the values for the residuum is indeed reached. In contrast, for the value restriction, this infimum is really the degree of this restriction, and thus it makes sense to require that this degree is witnessed by a role successor.

We say that an ontology $\mathcal{O}$ is *consistent* if it has a model, it is *witnessed consistent* if it has a witnessed model, and it is *strongly witnessed consistent* if it has a strongly witnessed model.

As it has been shown in [11], consistency does not imply witnessed consistency in fuzzy $\mathcal{ALC}$. However, it is standard in fuzzy DL to restrict reasoning to witnessed models only (see for instance [10], [11]).

Notation. We will often use the expression $\langle C = q \rangle$ to abbreviate the two axioms $\langle \top \sqsubseteq C \geq q \rangle, \langle \top \sqsubseteq \neg C \geq 1 - q \rangle$, and the expression $\langle C \equiv D \rangle$ to abbreviate $\langle C \sqsubseteq D \geq 1 \rangle, \langle D \sqsubseteq C \geq 1 \rangle$. Likewise, $\langle a : C = q \rangle$ is an abbreviation for $\langle a : C \geq q \rangle, \langle a : \neg C \geq 1 - q \rangle$.

Intuitively, the expressions $\langle C = q \rangle, \langle C \equiv D \rangle$, and $\langle a : C = q \rangle$ restrict every model $\mathcal{I}$ to be such that, for all $\delta \in \Delta^{\mathcal{I}}$, we have $C^{\mathcal{I}}(\delta) = q, C^{\mathcal{I}}(\delta) = D^{\mathcal{I}}(\delta)$, and $C^{\mathcal{I}}(a^{\mathcal{I}}) = q$, respectively.

B. The Original Algorithm of [10]

A tableau-based algorithm for deciding strongly witnessed consistency of a fuzzy $\mathcal{ALCF}(D)$ ontology was presented in [10].¹ However, as we will show, this algorithm is incorrect, even if restricted to fuzzy $\mathcal{ALC}$. We now briefly introduce the algorithm, leaving aside all the steps corresponding to expressivity beyond that of fuzzy $\mathcal{ALC}$. In the following, x (possibly with sub- or superindices) denotes a *continuous variable* taking values from $[0, 1]$, q denotes a *constant* in $[0, 1]$, l denotes a *literal*, that is, a continuous variable x , a *negated variable* $1 - x$ or a constant, and y denotes a *Boolean variable* taking values from $\{0, 1\}$.

The algorithm constructs a completion forest; that is, a collection of trees whose roots may be arbitrarily interconnected by edges. Every node v in this forest is labeled by a set $\mathcal{L}(v)$ of labeled concepts of the form $\langle C, l \rangle$ and a set $\mathcal{C}(v)$ of inequalities. Intuitively, $\langle C, l \rangle \in \mathcal{L}(v)$ means that v belongs to C with degree at least l and the inequalities in $\mathcal{C}(v)$ constrain the valuations of the different variables used. Additionally, every edge (v_1, v_2) is labeled with a set $\mathcal{L}(v_1, v_2)$ of labeled role names $\langle r, l \rangle$ with the meaning that (v_1, v_2) are in an r -relation with degree at least l .

The algorithm initializes the completion forest to contain one *root node* v_i for each individual name a_i appearing in the ABox $\mathcal{A}$ with $\mathcal{L}(v_i) = \{\langle C, l \rangle \mid \langle a_i : C \geq l \rangle \in \mathcal{A}\}$ and $\mathcal{C}(v_i, v_j) = \{\langle r, l \rangle \mid \langle (a_i, a_j) : r \geq l \rangle \in \mathcal{A}\}$. The sets $\mathcal{C}(v_i)$ are all initialized as empty. This completion forest is extended

¹Fuzzy $\mathcal{ALCF}(D)$ is fuzzy $\mathcal{ALC}$ extended with additional constructors that are not relevant for this paper.

- (A) if $\langle A, l \rangle \in \mathcal{L}(v)$ then add $x_{v:A} \geq l$ to $\mathcal{C}(v)$
- ($\bar{A}$) if $\langle \neg A, l \rangle \in \mathcal{L}(v)$ then add $x_{v:A} \leq 1 - l$ to $\mathcal{C}(v)$
- (r) if $\langle r, l \rangle \in \mathcal{L}(v, w)$ then add $x_{(v,w):r} \geq l$ to $\mathcal{C}(w)$
- ($\top$) if $\langle \top, l \rangle \in \mathcal{L}(v)$ or $\langle \neg \perp, l \rangle \in \mathcal{L}(v)$ then add $l = 1$ to $\mathcal{C}(v)$
- ($\perp$) if $\langle \perp, l \rangle \in \mathcal{L}(v)$ or $\langle \neg \top, l \rangle \in \mathcal{L}(v)$ then add $l = 0$ to $\mathcal{C}(v)$
- ($\neg$) if $\langle \neg \neg C, l \rangle \in \mathcal{L}(v)$ then add $\langle C, l \rangle$ to $\mathcal{L}(v)$
- ($\cap$) if $\langle C \sqcap D, l \rangle \in \mathcal{L}(v)$ then add $\langle C, x_1 \rangle, \langle D, x_2 \rangle$ to $\mathcal{L}(v)$ and $x_1 \cdot x_2 \geq l$ to $\mathcal{C}(v)$
- ($\bar{\cap}$) if $\langle \neg(C \sqcap D), l \rangle \in \mathcal{L}(v)$ then add $\langle \neg C \sqcup \neg D, l \rangle$ to $\mathcal{L}(v)$
- ($\sqcup$) if $\langle C \sqcup D, l \rangle \in \mathcal{L}(v)$ then add $\langle C, x_1 \rangle, \langle D, x_2 \rangle$ to $\mathcal{L}(v)$ and $x_1 + x_2 - x_1 \cdot x_2 \geq l$ to $\mathcal{C}(v)$
- ($\bar{\sqcup}$) if $\langle \neg(C \sqcup D), l \rangle \in \mathcal{L}(v)$ then add $\langle \neg C \sqcap \neg D, l \rangle$ to $\mathcal{L}(v)$
- ($\forall$) if $\langle \forall r.C, l_1 \rangle \in \mathcal{L}(v)$ and $\langle r, l_2 \rangle \in \mathcal{L}(v, w)$ then add $\langle C, x \rangle$ to $\mathcal{L}(w)$ and $x \geq l_1 \cdot l_2$ to $\mathcal{C}(w)$
- ($\bar{\forall}$) if $\langle \neg \exists r.C, l_1 \rangle \in \mathcal{L}(v)$ and $\langle r, l_2 \rangle \in \mathcal{L}(v, w)$ then add $\langle \neg C, 1 - x_1 \rangle$ to $\mathcal{L}(w)$ and $x_{(v,w):r} \leq x_2, x_1 \cdot x_2 = 1 - l_1$ to $\mathcal{C}(w)$
- ($\bar{\forall}$) if $\langle \neg \forall r.C, l \rangle \in \mathcal{L}(v)$ then create a new node w and add $\langle r, x_1 \rangle$ to $\mathcal{L}(v, w)$, $\langle C, x_2 \rangle$ to $\mathcal{L}(w)$ and $y \cdot x_1 + (1 - y) \cdot x_2 \geq y \cdot x_2 + (1 - y) \cdot x_1, l \leq y, y \cdot x_2 \leq x_1 - l \cdot x_1$ to $\mathcal{C}(w)$
- ($\exists$) if $\langle \exists r.C, l \rangle \in \mathcal{L}(v)$ then create a new node w and add $\langle r, x_1 \rangle$ to $\mathcal{L}(v, w)$, $\langle C, x_2 \rangle$ to $\mathcal{L}(w)$ and $x_1 \cdot x_2 \geq l$ to $\mathcal{C}(w)$
- ($\bar{\exists}$) if $\langle C \sqsubseteq D \geq q \rangle \in \mathcal{T}$ and v is a node in the forest, then add $\langle \neg C, 1 - x_1 \rangle, \langle D, x_2 \rangle$ to $\mathcal{L}(v)$ and $x_2 \geq x_1 \cdot q$ to $\mathcal{C}(v)$.

TABLE I
COMPLETION RULES FOR FUZZY $\mathcal{ALC}$ CONSISTENCY.
 x, x_1, x_2, y ARE ALWAYS NEW VARIABLES

by application of the completion rules from Table I. As is standard with tableau-based algorithms, these rules are only applied as long as something new is added to the completion forest.

The main idea of these rules is that they decompose complex concepts into their subconcepts, while preserving the fuzzy semantics through the restrictions in $\mathcal{C}$. Assume for the moment that the algorithm has terminated (i.e., no more completion rules are applicable) and that the constraint system $\mathcal{C} := \bigcup_v \mathcal{C}(v)$ generated by the algorithm has a solution in $[0, 1]$. The claim is then that the completion forest obtained by the algorithm describes a model of the ontology, where the membership degree of a node v to a concept name A corresponds to the value of the variable $x_{v:A}$, and accordingly for roles. Conversely, from any model of the ontology it is possible to build a satisfying valuation of the variables in $\mathcal{C}$.

However, as it is also the case for crisp $\mathcal{ALC}$, this completion forest construction may not terminate, due to the fact that concept inclusions can express cyclic relationships between concepts, which may enforce an infinite generation of new individuals. In order to ensure termination, a blocking condition is used, which disallows the application of “generating” rules; i.e., the ($\exists$) and ($\bar{\forall}$) rules cannot be applied in any node that is considered to be blocked.

Definition 1 (blocking [10]). *Two non-root nodes v, w are called equivalent, denoted as $\mathcal{L}(v) \approx \mathcal{L}(w)$, if $\mathcal{L}(v) = \{\langle C_1, l_1 \rangle, \dots, \langle C_k, l_k \rangle\}, \mathcal{L}(w) = \{\langle C_1, l'_1 \rangle, \dots, \langle C_k, l'_k \rangle\}$ and for every $1 \leq i \leq k$, l_i, l'_i are either both variables, or are*

both negated variables, or are both the same constant.

A node v is directly blocked iff it is not a root node and it has an ancestor w such that $\mathcal{L}(v) \approx \mathcal{L}(w)$; in this case we say that w is the blocking node of v . A node is blocked if it is directly blocked or its predecessor is blocked.

With this blocking condition, it is not hard to show that the algorithm always terminates. It then answers that the input ontology $\mathcal{O}$ is strongly witnessed consistent iff the system of inequalities $\mathcal{C}$ generated by this terminating run on input $\mathcal{O}$ has a solution.

The idea behind the proof of correctness of this algorithm, as presented in [10], is that the variables appearing in blocking and blocked nodes can be evaluated to the same numerical value in a solution of $\mathcal{C}$. If correct, this proof would show that an ontology that has a strongly witnessed model always has a finite model. Indeed, the interpretation constructed from the completion forest generated by a terminating run of the algorithm is finite. Unfortunately, as we will see below, it need not be a model. The first blow to the claimed correctness of the algorithm came when it was shown in [12] that the logic does not have the finite model property. More precisely, in [12] the authors present a simple ontology that is strongly witnessed consistent, but does not have a finite model. Thus, the approach through which the algorithm tries to build a (finite) model is wrong. However, this does not automatically imply that the algorithm itself is wrong. In fact, if one applies the algorithm to the ontology used in [12], then one obtains the correct answer, namely that the ontology is strongly witnessed consistent. The authors conjecture in [12] that the algorithm is still correct.²

Note that the algorithm tries to construct a finite portion of a model. If the input ontology is indeed consistent, and thus has a (possibly infinite) model, then it is easy to see that this construction will always succeed (i.e., the constraint system generated by the algorithm is satisfiable). This means that the algorithm is complete.

Remark 2. *The algorithm from [10] is complete for strongly witnessed consistency; that is, it yields the right answer whenever the input ontology is strongly witnessed consistent.*

Unfortunately, as the following example shows, it is not sound since it fails to correctly identify inconsistency.

Example 3. *Consider the ontology with the following axioms:*

- $ax_1 : \langle a : A \geq 0.1 \rangle$
- $ax_2 : \langle A \sqsubseteq (\forall r.A) \sqcap B \geq 1 \rangle$
- $ax_3 : \langle \top \sqsubseteq \neg B \geq 0.5 \rangle$
- $ax_4 : \langle \top \sqsubseteq \exists r.\top \geq 1 \rangle$.

We will prove first that this ontology is inconsistent, and thus also not (strongly) witnessed consistent. We then show that the algorithm described above yields the wrong answer, i.e., it claims that the ontology is strongly witnessed consistent.

²Independently from us, a counterexample to the correctness of the algorithm in [10] has now also been found by the authors of [12]. This counterexample, which is presented in [16], is different from the one we give here. The paper [16], which is an improved version of [12], was available online only after we had submitted the present paper.

Suppose that there is a model (not necessarily witnessed) $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ of this ontology, and let $v_0 \in \Delta^{\mathcal{I}}$ be any element of the domain. From axiom ax_4 , we obtain that

$$1 = (\exists r. \top)^{\mathcal{I}}(v_0) = \sup_{\delta \in \Delta^{\mathcal{I}}} r^{\mathcal{I}}(v_0, \delta).$$

This means that there must exist $v_1 \in \Delta^{\mathcal{I}}$ such that $r^{\mathcal{I}}(v_0, v_1) > 0.99$.³ From ax_3 it follows that $(\neg B)^{\mathcal{I}}(v_0) \geq 0.5$; i.e., $B^{\mathcal{I}}(v_0) \leq 0.5$; additionally, from ax_2 we know

$$A^{\mathcal{I}}(v_0) \leq (\forall r. A)^{\mathcal{I}}(v_0) \cdot B^{\mathcal{I}}(v_0) \leq (\forall r. A)^{\mathcal{I}}(v_0) \cdot 0.5,$$

or equivalently,

$$\begin{aligned} 2 \cdot A^{\mathcal{I}}(v_0) \leq (\forall r. A)^{\mathcal{I}}(v_0) &= \inf_{\delta \in \Delta^{\mathcal{I}}} r^{\mathcal{I}}(v_0, \delta) \rightarrow A^{\mathcal{I}}(\delta) \\ &\leq r^{\mathcal{I}}(v_0, v_1) \rightarrow A^{\mathcal{I}}(v_1) \\ &\leq 0.99 \rightarrow A^{\mathcal{I}}(v_1). \end{aligned} \quad (1)$$

Additionally, it holds that $0.99 \rightarrow A^{\mathcal{I}}(v_1) \leq 1$, and so $A^{\mathcal{I}}(v_0) \leq 0.5$. Recall that v_0 was an arbitrary element of the domain, and thus the previous holds also for v_1 ; in particular we have that $A^{\mathcal{I}}(v_1) \leq 0.5$ and $0.99 \rightarrow A^{\mathcal{I}}(v_1) = c \cdot A^{\mathcal{I}}(v_1)$, where $c = 100/99$. Since $2/c > 1$, it follows from the inequality (1) that

$$A^{\mathcal{I}}(v_0) < (2/c) \cdot A^{\mathcal{I}}(v_0) \leq A^{\mathcal{I}}(v_1) \leq 0.5.$$

Once again, since all the previous is true for any element of the domain, it follows that there must exist infinitely many elements $v_2, v_3, \dots$ such that for every $n \geq 0$ it holds

$$(2/c)^n \cdot A^{\mathcal{I}}(v_0) \leq A^{\mathcal{I}}(v_n) \leq 0.5$$

But since $2/c > 1$, it then follows that $A^{\mathcal{I}}(v_0) = 0$. In particular, this is true for $v_0 = a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. But this is a contradiction to the assumption that $\mathcal{I}$ is a model of the axiom ax_1 . Thus, the ontology is inconsistent.

Let us now analyse the execution of the algorithm on this input. The algorithm starts with a single root node v_0 with $\mathcal{L}(v_0) = \{\langle A, 0.1 \rangle\}$ and $\mathcal{C}(v_0) = \emptyset$. Application of the rule $(\sqsubseteq)$ with axioms ax_2 , ax_3 and ax_4 will add

$$\langle \neg A, 1 - x_1 \rangle, \langle (\forall r. A) \sqcap B, x_2 \rangle, \langle \neg \top, 1 - x_3 \rangle, \langle \neg B, x_4 \rangle, \\ \langle \neg \top, 1 - x_5 \rangle, \langle \exists r. \top, x_6 \rangle$$

to $\mathcal{L}(v_0)$ and $x_1 \leq x_2, 0.5 \cdot x_3 \leq x_4, x_5 \leq x_6$ to $\mathcal{C}(v_0)$. The rule $(\sqcap)$ then adds $\langle \forall r. A, x_7 \rangle, \langle B, x_8 \rangle$ to $\mathcal{L}(v_0)$ and $x_2 = x_7 \cdot x_8$ to $\mathcal{C}(v_0)$. The rules (A) , $(\bar{A})$, and $(\perp)$ then yield

$$\begin{aligned} 0.1 \leq x_{v_0:A} \leq x_1, \quad x_8 \leq x_{v_0:B} \leq 1 - x_4, \\ 1 - x_3 = 0, \quad 1 - x_5 = 0. \end{aligned}$$

Finally, an application of the rule $(\exists)$ produces a new node v_1 with labels $\mathcal{L}(v_1) = \{\langle \top, x_9 \rangle\}$, $\mathcal{L}(v_0, v_1) = \{\langle r, x_{10} \rangle\}$, and $\mathcal{C}(v_1) = \{x_9 \cdot x_{10} = x_6\}$. The rule $(\forall)$ then adds $\langle A, x_{11} \rangle$ to $\mathcal{L}(v_1)$ and $x_{11} \geq x_7 \cdot x_{10}$ to $\mathcal{C}(v_1)$.

Notice that all the concepts and constraints added to the labels of v_0 , except $\langle A, 0.1 \rangle$, were caused by the axioms ax_2

to ax_4 , and thus they will also be added to v_1 . The result of the rule applications to this node is then

$$\begin{aligned} \mathcal{L}(v_1) &= \{ \langle A, x_{11} \rangle, \langle \neg A, 1 - x'_1 \rangle, \langle (\forall r. A) \sqcap B, x'_2 \rangle, \\ &\quad \langle \neg \top, 1 - x'_3 \rangle, \langle \neg B, x'_4 \rangle, \\ &\quad \langle \neg \top, 1 - x'_5 \rangle, \langle \exists r. \top, x'_6 \rangle, \langle \forall r. A, x'_7 \rangle, \langle B, x'_8 \rangle \}, \\ \mathcal{C}(v_1) &= \{ x_9 \cdot x_{10} = x_6, x_{11} \geq x_7 \cdot x_{10}, x'_1 \leq x'_2, \\ &\quad 0.5 \cdot x'_3 \leq x'_4, x'_5 \leq x'_6, x'_2 = x'_7 \cdot x'_8, \\ &\quad x_{11} \leq x_{v_1:A} \leq x'_1, x'_8 \leq x_{v_1:B} \leq 1 - x'_4, \\ &\quad 1 - x'_3 = 0, 1 - x'_5 = 0 \}. \end{aligned}$$

Once again we can apply the $(\exists)$ rule to obtain a new individual v_2 . From the $(\forall)$ rule and again the application of $(\sqsubseteq)$ rule with axioms ax_2 to ax_4 we obtain

$$\begin{aligned} \mathcal{L}(v_2) &= \{ \langle A, x'_{11} \rangle, \langle \neg A, 1 - x''_1 \rangle, \langle (\forall r. A) \sqcap B, x''_2 \rangle, \\ &\quad \langle \neg \top, 1 - x''_3 \rangle, \langle \neg B, x''_4 \rangle, \\ &\quad \langle \neg \top, 1 - x''_5 \rangle, \langle \exists r. \top, x''_6 \rangle, \langle \forall r. A, x''_7 \rangle, \langle B, x''_8 \rangle \}, \\ \mathcal{C}(v_2) &= \{ x'_9 \cdot x'_{10} = x'_6, x'_{11} \geq x'_7 \cdot x'_{10}, x'_1 \leq x''_2, \\ &\quad 0.5 \cdot x''_3 \leq x''_4, x''_5 \leq x''_6, x''_2 = x''_7 \cdot x''_8, \\ &\quad x'_{11} \leq x'_{v_2:A} \leq x'_1, x''_8 \leq x_{v_2:B} \leq 1 - x''_4, \\ &\quad 1 - x''_3 = 0, 1 - x''_5 = 0 \}. \end{aligned}$$

At this point $\mathcal{L}(v_1) \approx \mathcal{L}(v_2)$, and hence node v_2 is blocked and the algorithm terminates.

It is easy to verify that the valuation that sets

$$\begin{aligned} 0.1 &= x_1 = x_2 = x_{v_0:A}, \\ 0.2 &= x_7 = x_{11} = x'_1 = x'_2 = x_{v_1:A}, \\ 0.4 &= x'_7 = x'_{11} = x''_1 = x''_2 = x_{v_2:A}, \\ 0.5 &= x_4 = x'_4 = x''_4 = x_8 = x'_8 = x''_8 \\ &= x_{v_0:B} = x_{v_1:B} = x_{v_2:B}, \\ 0.8 &= x''_7 \end{aligned}$$

and all other variables to 1 satisfies all the constraints in $\mathcal{C}$. Thus the algorithm returns “strongly witnessed consistent” as an answer, which we have shown to be incorrect.

Notice that the arguments used to show inconsistency in this example do not depend on the exact value of the constant 0.1 in ax_1 . In order to make the ontology inconsistent, any constant κ greater than 0 can be used. However, the algorithm multiplies this value with 2 three times before stopping (see the valuation of x''_7 above). To ensure that the algorithm finishes with a satisfiable system of constraints, the constant chosen has to be small enough to ensure that $8 \cdot \kappa \leq 1$.

We emphasize that the ontology from the previous example has no model at all, which means that the algorithm is incorrect even if we do not restrict to only strongly witnessed models. Thus, at this moment there is no known algorithm for deciding consistency, witnessed consistency, or strongly witnessed consistency of fuzzy $\mathcal{ALC}$ ontologies under the product t-norm and involutive negation. In fact, it is not even clear whether this is a decidable problem. In [13] it was shown that, in the absense of concept inclusion axioms,

³Notice that the choice of 0.99 is an arbitrary one; the same arguments can be used with any constant $\kappa > 0.5$.

consistency of fuzzy $\mathcal{ALC}$ with product t-norm and residual negation w.r.t. witnessed models is decidable; this result was extended to so-called quasi-witnessed models (but still without concept inclusion axioms) in [17]. However, there is so far no result regarding the use of involutive negation or concept inclusion axioms. In the following section, we show that for a small extension of this logic, witnessed consistency becomes undecidable.

III. AN UNDECIDABILITY RESULT

We now consider the logic fuzzy $\mathcal{ALC}^+$, which extends fuzzy $\mathcal{ALC}$ by *strict concept inclusions*.

Definition 4 (strict concept inclusion). A strict concept inclusion axiom is an expression of the form $\langle C \sqsubseteq D > q \rangle$ where C, D are fuzzy $\mathcal{ALC}$ concepts and $q \in [0, 1]$. An interpretation $\mathcal{I}$ satisfies this axiom if for every $\delta \in \Delta^{\mathcal{I}}$ it holds that

$$C^{\mathcal{I}}(\delta) \rightarrow D^{\mathcal{I}}(\delta) > q.$$

For models that are not strongly witnessed, the semantics for strict axioms introduced above may differ from the one given by

$$\inf_{\delta \in \Delta^{\mathcal{I}}} (C^{\mathcal{I}}(\delta) \rightarrow D^{\mathcal{I}}(\delta)) > q$$

since the infimum may be q although there is no witness δ actually yielding the value q . In contrast, for the case of non-strict axioms, saying that the infimum is $\geq q$ is the same as saying that the values for all individuals are $\geq q$, i.e., $C^{\mathcal{I}}(\delta) \rightarrow D^{\mathcal{I}}(\delta) \geq q$ for all $\delta \in \Delta$. As argued before, we think that the essence of a concept inclusion axiom is to enforce the inequality for the value of all individuals, and not to say something about the infimum of these values. This explains why we do not use the infimum in our definition of the semantics for strict concept inclusions.

We will show that witnessed consistency of fuzzy $\mathcal{ALC}^+$ ontologies is undecidable, by a reduction from the Post correspondence problem, which is well-known to be undecidable [18].

Definition 5 (PCP). Let $v_1, \dots, v_m$ and $w_1, \dots, w_m$ be two finite lists of words over an alphabet $\Sigma = \{1, \dots, s\}$, $s > 1$. The Post correspondence problem (PCP) asks whether there is a non-empty sequence $i_1, i_2, \dots, i_k$, $1 \leq i_j \leq m$ such that $v_{i_1} v_{i_2} \dots v_{i_k} = w_{i_1} w_{i_2} \dots w_{i_k}$. If such a sequence exists, it is called a solution of the problem.

We assume w.l.o.g. that $v_i \neq w_i$ for every i , $1 \leq i \leq m$ since otherwise the problem has a trivial solution. Note that we assume the alphabet Σ to consist of the first s positive integers. We can thus view every word in Σ^* as a number in base $s+1$ representation in which 0 never occurs. Using this intuition, we will represent the empty word as the number 0.

Before describing the reduction in detail, we introduce a useful abbreviation.

Notation. Let C, D be two concepts and r a role name. We use the expression $\langle D \overset{r}{\rightsquigarrow} C \rangle$ to abbreviate the two axioms $\langle D \sqsubseteq \forall r.C \geq 1 \rangle$, $\langle \neg D \sqsubseteq \forall r.\neg C \geq 1 \rangle$.

To understand this abbreviation, consider an interpretation $\mathcal{I}$ satisfying $\langle D \overset{r}{\rightsquigarrow} C \rangle$ and let $\delta, \gamma \in \Delta^{\mathcal{I}}$ with $r^{\mathcal{I}}(\delta, \gamma) = 1$. From the first axiom it follows that

$$\begin{aligned} D^{\mathcal{I}}(\delta) &\leq (\forall r.C)^{\mathcal{I}}(\delta) = \inf_{\eta \in \Delta^{\mathcal{I}}} r^{\mathcal{I}}(\delta, \eta) \rightarrow C^{\mathcal{I}}(\eta) \\ &\leq r^{\mathcal{I}}(\delta, \gamma) \rightarrow C^{\mathcal{I}}(\gamma) = 1 \rightarrow C^{\mathcal{I}}(\gamma) \\ &= C^{\mathcal{I}}(\gamma). \end{aligned}$$

Analogously, from the second axiom we can deduce that

$$1 - D^{\mathcal{I}}(\delta) \leq 1 - C^{\mathcal{I}}(\gamma),$$

and hence $D^{\mathcal{I}}(\delta) = C^{\mathcal{I}}(\gamma)$. Thus, $\langle D \overset{r}{\rightsquigarrow} C \rangle$ expresses that the value of $D^{\mathcal{I}}(\delta)$ is propagated to the valuation of the concept C on all r successors with degree 1 of δ .

Let now C, D be two concepts and $\mathcal{T}$ the TBox

$$\begin{aligned} \text{ax}_1 &: \langle C \sqcup Y_1 \equiv X_1 \rangle \\ \text{ax}_2 &: \langle X_1 \equiv S \sqcap \neg Y_1 \rangle \\ \text{ax}_3 &: \langle \neg Y_1 \sqcap D \equiv Y_1 \rangle \end{aligned}$$

For every model $\mathcal{I}$ of $\mathcal{T}$ and $\delta \in \Delta^{\mathcal{I}}$ it holds that

$$C^{\mathcal{I}}(\delta)(1 - Y_1^{\mathcal{I}}(\delta)) + Y_1^{\mathcal{I}}(\delta) = X_1^{\mathcal{I}}(\delta), \quad (2)$$

$$X_1^{\mathcal{I}}(\delta) = S^{\mathcal{I}}(\delta)(1 - Y_1^{\mathcal{I}}(\delta)), \quad (3)$$

$$D^{\mathcal{I}}(\delta)(1 - Y_1^{\mathcal{I}}(\delta)) = Y_1^{\mathcal{I}}(\delta), \quad (4)$$

where equalities (2), (3), and (4) follow from axioms ax_1 , ax_2 , and ax_3 , respectively. Additionally, ax_3 implies that $Y_1^{\mathcal{I}}(\delta) \neq 1$ since otherwise $Y_1^{\mathcal{I}}(\delta)$ must also be 0. It then holds that

$$C^{\mathcal{I}}(\delta) + D^{\mathcal{I}}(\delta) = S^{\mathcal{I}}(\delta).$$

We can thus introduce the concept constructor $\boxplus$ to denote this addition without extending the expressivity of the logic. Notice however that the TBox having the axioms ax_1 to ax_3 is only satisfiable by models where $C^{\mathcal{I}}(\delta) + D^{\mathcal{I}}(\delta) \leq 1$ for all $\delta \in \Delta^{\mathcal{I}}$. One must keep this in mind when using this constructor to avoid making the ontology inconsistent. Given $C^{\mathcal{I}}(\delta)$ and $D^{\mathcal{I}}(\delta)$, there is exactly one value for $X_1^{\mathcal{I}}(\delta)$ and $Y_1^{\mathcal{I}}(\delta)$ that will satisfy the three axioms above. Then, describing the valuation of the concepts C and D suffices for knowing also the valuation of the auxiliary concepts X_1, Y_1 .

Notation. Let C, D be two concepts, then $C \boxplus D$ is also a concept, with the semantics $(C \boxplus D)^{\mathcal{I}}(\delta) = C^{\mathcal{I}}(\delta) + D^{\mathcal{I}}(\delta)$.

Let $\mathcal{P} = (v_1, \dots, v_m, w_1, \dots, w_m)$ be an instance of the PCP. We will construct an ontology $\mathcal{O}_{\mathcal{P}}$ whose models represent a search tree for a solution for $\mathcal{P}$. More precisely, every model of $\mathcal{O}_{\mathcal{P}}$ will contain one node δ for each possible sequence of indices $i_1, \dots, i_k$, such that the interpretations of two concept names A and B in δ encode the words $v_{i_1} \dots v_{i_k}$ and $w_{i_1} \dots w_{i_k}$, respectively. Then $\mathcal{P}$ has a solution iff for every model $\mathcal{I}$ of $\mathcal{O}_{\mathcal{P}}$ there is a δ such that $A^{\mathcal{I}}(\delta)$ and $B^{\mathcal{I}}(\delta)$ encode the same word.

Let $\Sigma = \{1, \dots, s\}$ be the alphabet over which $\mathcal{P}$ is defined. We will encode words in Σ^* using numbers from the interval $[0, 1]$ in base $s+1$. For technical reasons that will become

clear later, the concepts A and B will encode words in a slightly different way. For every sequence of indices $i_1, \dots, i_k$, there will be a node δ such that $A^\mathcal{I}(\delta) = 0.1v_{i_1} \dots v_{i_k}$ and $B^\mathcal{I}(\delta) = 0.0w_{i_1} \dots w_{i_k}$ (see Figure 1).

If there is a node δ encoding the words v and w , then we want to ensure the existence of a node γ that encodes their concatenation with the i -th pair; i.e., vv_i and ww_i . To do this, we define, for every $i, 1 \leq i \leq m$, the TBox $\mathcal{T}_P^i$ as follows:

$$\begin{aligned} \mathcal{T}_P^i := & \{ \langle \exists r_i. \top = 1 \rangle, \langle V_i = 0.0v_i \rangle, \langle W_i = 0.0w_i \rangle, \\ & \langle L_{v_i} = (s+1)^{-|v_i|} \rangle, \langle L_{w_i} = (s+1)^{-|w_i|} \rangle, \\ & \langle A \boxplus (V_i \sqcap L_A) \rightsquigarrow^i A \rangle, \langle L_A \sqcap L_{v_i} \rightsquigarrow^i L_A \rangle, \\ & \langle B \boxplus (W_i \sqcap L_B) \rightsquigarrow^i B \rangle, \langle L_B \sqcap L_{w_i} \rightsquigarrow^i L_B \rangle \}. \end{aligned} \quad (5)$$

Let $\mathcal{I}$ be a witnessed model of $\mathcal{T}_P^i$ and $\delta \in \Delta^\mathcal{I}$ such that $A^\mathcal{I}(\delta) = 0.1v$ and $B^\mathcal{I}(\delta) = 0.0w$ for some words $v, w \in \Sigma^*$. Assume additionally that $L_A^\mathcal{I}(\delta) = (s+1)^{-|v|}$ and $L_B^\mathcal{I}(\delta) = (s+1)^{-|w|}$; that is, L_A, L_B yield information about the length of the words encoded by A and B in δ . From the first axiom we know that $(\exists r_i. \top)^\mathcal{I}(\delta) = 1$, and since $\mathcal{I}$ is witnessed, there must exist $\gamma \in \Delta^\mathcal{I}$ such that $r_i^\mathcal{I}(\delta, \gamma) = 1$. The axioms in line (5) ensure that

$$\begin{aligned} A^\mathcal{I}(\gamma) &= (A \boxplus (V_i \sqcap L_A))^\mathcal{I}(\delta) \\ &= A^\mathcal{I}(\delta) + (V_i^\mathcal{I}(\delta) \cdot L_A^\mathcal{I}(\delta)) \\ &= 0.1v + (0.0v_i \cdot (s+1)^{-|v|}) = 0.1vv_i \\ L_A^\mathcal{I}(\gamma) &= (L_A \sqcap L_{v_i})^\mathcal{I}(\delta) \\ &= (s+1)^{-|v|} \cdot (s+1)^{-|v_i|} = (s+1)^{-|vv_i|} \end{aligned}$$

and the axioms in (6) analogously entail

$$\begin{aligned} B^\mathcal{I}(\gamma) &= 0.0ww_i \\ L_B^\mathcal{I}(\gamma) &= (s+1)^{-|ww_i|}. \end{aligned}$$

Thus, the interpretation of A, B at node γ encodes the words vv_i, ww_i , and the concepts L_A, L_B yield information on the lengths of these words.

As we have seen, we can use the TBox $\mathcal{T}_P^i$ to ensure that the concatenation of some words with the i -th pair of $\mathcal{P}$ will be encoded in every model. The following ABox $\mathcal{A}_P$ introduces the root of the search tree, where A and B both encode the empty word, which obviously has length 0.

$$\mathcal{A}_P := \{ \langle a : A = 0.1 \rangle, \langle a : B = 0 \rangle, \\ \langle a : L_A = 1 \rangle, \langle a : L_B = 1 \rangle \}$$

Notice that every finite sequence of indices $i_1, \dots, i_k$, with $1 \leq i_j \leq m$ can be seen as a word $\nu \in \{1, \dots, m\}^*$. For a word $\nu = i_1 i_2 \dots i_k \in \{1, \dots, m\}^*$, let v_ν and w_ν represent the words $v_{i_1} \dots v_{i_k}, w_{i_1} \dots w_{i_k} \in \{1, \dots, s\}^*$, respectively. We define the tree-like interpretation $\mathcal{I}_P := (\Delta^{\mathcal{I}_P}, \cdot^{\mathcal{I}_P})$, where

- $\Delta^{\mathcal{I}_P} = \{1, \dots, m\}^*$,
- $a^{\mathcal{I}_P} = \varepsilon$,
- $A^{\mathcal{I}_P}(\nu) = 0.1v_\nu, B^{\mathcal{I}_P}(\nu) = 0.0w_\nu$,
- $L_A^{\mathcal{I}_P}(\nu) = (s+1)^{-|v_\nu|}, L_B^{\mathcal{I}_P}(\nu) = (s+1)^{-|w_\nu|}$, and

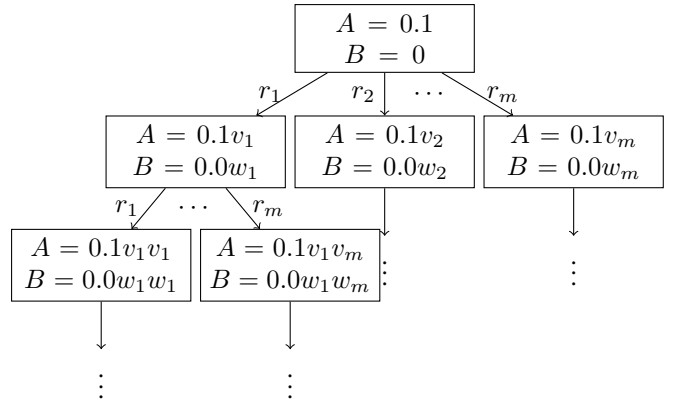


Fig. 1. The tree-shaped interpretation $\mathcal{I}_P$.

for all $i, 1 \leq i \leq m$

- $r_i^{\mathcal{I}_P}(\nu, \nu i) = 1$ and $r_i^{\mathcal{I}_P}(\nu, \nu') = 0$ if $\nu' \neq \nu i$,
- $V_i^{\mathcal{I}_P}(\nu) = 0.0v_i, W_i^{\mathcal{I}_P}(\nu) = 0.0w_i$,
 $L_{v_i}^{\mathcal{I}_P}(\nu) = (s+1)^{-|v_i|}$, and $L_{w_i}^{\mathcal{I}_P}(\nu) = (s+1)^{-|w_i|}$.

The main idea underlying this interpretation is depicted in Figure 1. It is easy to see that this interpretation is a model of the ontology $\mathcal{O}_P := (\mathcal{A}_P, \bigcup_{i=1}^m \mathcal{T}_P^i)$.

Lemma 6. $\mathcal{I}_P$ is a model of $\mathcal{O}_P$.

More interesting, however, is that every witnessed model of $\mathcal{O}_P$ must “include” $\mathcal{I}_P$, as stated by the following lemma.

Lemma 7. Let $\mathcal{I}$ be a witnessed model of $\mathcal{O}_P$. Then there exists a function $f : \Delta^{\mathcal{I}_P} \rightarrow \Delta^\mathcal{I}$ such that $C^{\mathcal{I}_P}(\nu) = C^\mathcal{I}(f(\nu))$ holds for every concept name C occurring in $\mathcal{O}_P$ and every $\nu \in \Delta^{\mathcal{I}_P}$.

Proof: We first remark that every model of $\mathcal{O}_P$ must interpret the concept names V_i, W_i, L_{v_i} , and L_{w_i} with the same constant function defined by the TBox; e.g. $V_i^\mathcal{I}(\delta) = 0.0v_i$ for every model $\mathcal{I}$ and $\delta \in \Delta^\mathcal{I}$. Thus, the interpretation function of any model is fully determined by the interpretation of the concept names A, B, L_A, L_B .

We build the function f inductively on the length of ν . First, since $\mathcal{I}$ is a model of $\mathcal{A}_P$, there must be a $\delta \in \Delta^\mathcal{I}$ such that $a^\mathcal{I} = \delta$. We fix $f(\varepsilon) := \delta$, and verify that

$$\begin{aligned} A^\mathcal{I}(\delta) &= 0.1 = A^{\mathcal{I}_P}(\varepsilon) \\ B^\mathcal{I}(\delta) &= 0 = B^{\mathcal{I}_P}(\varepsilon) \\ L_A^\mathcal{I}(\delta) &= L_B^\mathcal{I}(\delta) = 1 = L_A^{\mathcal{I}_P}(\varepsilon) = L_B^{\mathcal{I}_P}(\varepsilon). \end{aligned}$$

Let now ν be such that $f(\nu)$ has been defined. Then by induction we assume $A^\mathcal{I}(f(\nu)) = 0.1v_\nu, B^\mathcal{I}(f(\nu)) = 0.0w_\nu$, and L_A, L_B represent the lengths of these words. Since $\mathcal{I}$ is a witnessed model, for every $i, 1 \leq i \leq m$ there must exist a $\gamma \in \Delta^\mathcal{I}$ with $r_i^\mathcal{I}(f(\nu), \gamma) = 1, A^\mathcal{I}(\gamma) = 0.1v_\nu v_i, B^\mathcal{I}(\gamma) = 0.0w_\nu w_i$, and L_A, L_B encode the respective lengths. We set $f(\nu i) := \gamma$, which as we have seen satisfies the required property. ■

From this lemma it follows that, if the PCP $\mathcal{P}$ has a solution

ν for some $\nu \in \{1, \dots, m\}^+$, then every witnessed model $\mathcal{I}$ must contain a node $\delta = f(\nu)$ such that $A^\mathcal{I}(\delta) = 0.1 + B^\mathcal{I}(\delta)$; that is, where A and B encode the same word. Thus, to decide whether $\mathcal{P}$ has a solution, we will search for a node δ such that $((\neg A) \boxplus B)^\mathcal{I}(\delta) = 1 - 0.1 = s/(s+1)$. Conversely, if every witnessed model contains such a node, then in particular $\mathcal{I}_\mathcal{P}$ has such a node, and thus $\mathcal{P}$ has a solution.

We now show how to detect whether a node where A and B encode the same non-empty word exists in every interpretation $\mathcal{I}$. To do this, we will introduce *flag* concepts that are interpreted as 1 whenever a given condition is satisfied. Let C, D be two concepts and $q \in [0, 1]$. The axiom

$$\langle \neg F \sqsubseteq \neg F \sqcap \neg F \geq 1 - q \rangle$$

restricts every model $\mathcal{I}$ to satisfy, for every $\delta \in \Delta^\mathcal{I}$

$$((\neg F)^\mathcal{I}(\delta))^2 \geq (\neg F)^\mathcal{I}(\delta) \cdot (1 - q);$$

that is, for every $\delta \in \Delta^\mathcal{I}$, it holds that either $(\neg F)^\mathcal{I}(\delta) = 0$ or $(\neg F)^\mathcal{I}(\delta) \geq 1 - q$. If we add the axiom $\langle C \sqsubseteq F \geq 1 \rangle$, then it follows that $C^\mathcal{I}(\delta) \leq F^\mathcal{I}(\delta)$, and hence, $F^\mathcal{I}(\delta) = 1$ if $C^\mathcal{I}(\delta) > q$. We will denote this concept F as $\mathbb{F}_{[C > q]}$. In addition, if we have the axioms

$$\langle \neg F \sqsubseteq \neg F \sqcap \neg F > q \rangle, \langle \neg F \sqsubseteq C \geq 1 \rangle$$

then it follows that for every $\delta \in \Delta^\mathcal{I}$, $F^\mathcal{I}(\delta) < 1 - q$ or $F^\mathcal{I}(\delta) = 1$, and that $1 - C^\mathcal{I}(\delta) \leq F^\mathcal{I}(\delta)$, and hence $F^\mathcal{I}(\delta) = 1$ if $C^\mathcal{I}(\delta) \leq q$. Such concept F will be denoted as $\mathbb{F}_{[C \leq q]}$.⁴ In a similar way, we can define flags $\mathbb{F}_{[C < q]}$ and $\mathbb{F}_{[C \geq q]}$. Finally, the flag $\mathbb{F}_{[C = q]}$ is defined by $\langle \mathbb{F}_{[C = q]} \equiv \mathbb{F}_{[C \leq q]} \sqcap \mathbb{F}_{[C \geq q]} \rangle$.

Notation. For a concept C and $q \in [0, 1]$, the expression $\mathbb{F}_{[C \boxtimes q]}$, where $\boxtimes$ is one of $>, <, \geq, \leq, =$, denotes a concept such that for every model $\mathcal{I}$ and $\delta \in \Delta^\mathcal{I}$, $\mathbb{F}_{[C \boxtimes q]}^\mathcal{I}(\delta) = 1$ if $C^\mathcal{I}(\delta) \boxtimes q$.

Using these flags, we can detect whether A and B encode the same word, that is $((\neg A) \boxplus B)^\mathcal{I}(\delta) = s/(s+1)$, by checking whether the flag $\mathbb{F}_{[(\neg A) \boxplus B = s/(s+1)]}$ is interpreted as 1, and test that this word is not the empty word using the flag $\mathbb{F}_{[A \sqcup B > 0.1]}$. Notice, additionally, that $A^{\mathcal{I}_\mathcal{P}}(\delta) \geq B^{\mathcal{I}_\mathcal{P}}(\delta)$ for all δ , and thus the concept $(\neg A) \boxplus B$ is well-defined; that is, the axioms that are used to encode it are satisfied by $\mathcal{I}_\mathcal{P}$.

Thus, for a node δ , if $A^\mathcal{I}(\delta)$ and $B^\mathcal{I}(\delta)$ encode the same non-empty word, then

$$(\mathbb{F}_{[(\neg A) \boxplus B = s/(s+1)]} \sqcap \mathbb{F}_{[A \sqcup B > 0.1]})^\mathcal{I}(\delta) = 1.$$

We now define the ontology $\mathcal{O}'_\mathcal{P} := (\mathcal{A}_\mathcal{P}, \mathcal{T}_\mathcal{P})$ where

$$\begin{aligned} \mathcal{T}_\mathcal{P} := & \bigcup_{i=1}^m \mathcal{T}_\mathcal{P}^i \cup \\ & \{ \top \sqsubseteq \neg(\mathbb{F}_{[(\neg A) \boxplus B = s/(s+1)]} \sqcap \mathbb{F}_{[A \sqcup B > 0.1]}) > 0 \}. \end{aligned}$$

Theorem 8. *The instance $\mathcal{P}$ of the PCP has a solution iff the ontology $\mathcal{O}'_\mathcal{P}$ is not witnessed consistent.*

⁴Notice that this is the first instance of a strict axiom used in our construction so far.

Proof: Assume first that $\mathcal{P}$ has a solution $i_1, \dots, i_k$ and let $\nu = i_1 i_2 \dots i_k \in \{1, \dots, m\}^*$ and $u = v_\nu = w_\nu$. Suppose there is a witnessed model $\mathcal{I}$ of $\mathcal{O}'_\mathcal{P}$. Since $\mathcal{O}_\mathcal{P} \subseteq \mathcal{O}'_\mathcal{P}$, $\mathcal{I}$ must also be a model of $\mathcal{O}_\mathcal{P}$. From Lemma 7 it follows that there is a node $\delta \in \Delta^\mathcal{I}$ such that

$$\begin{aligned} A^\mathcal{I}(\delta) &= A^{\mathcal{I}_\mathcal{P}}(\nu) = 0.1u, \\ B^\mathcal{I}(\delta) &= B^{\mathcal{I}_\mathcal{P}}(\nu) = 0.0u, \end{aligned}$$

We thus have $((\neg A) \boxplus B)^\mathcal{I}(\delta) = s/(s+1)$ and hence $\mathbb{F}_{[(\neg A) \boxplus B = s/(s+1)]}^\mathcal{I}(\delta) = 1$. Additionally, since u is not the empty word, it follows that $(A \sqcup B)^\mathcal{I}(\delta) > 0.1$ and thus $\mathbb{F}_{[A \sqcup B > 0.1]}^\mathcal{I}(\delta) = 1$. But this violates the last axiom in $\mathcal{T}_\mathcal{P}$, and hence $\mathcal{I}$ cannot be a model of $\mathcal{O}'_\mathcal{P}$.

To show the converse, we will extend $\mathcal{I}_\mathcal{P}$ to interpret the three flag concepts $\mathbb{F}_{[(\neg A) \boxplus B \geq s/(s+1)]}$, $\mathbb{F}_{[(\neg A) \boxplus B \leq s/(s+1)]}$, and $\mathbb{F}_{[A \sqcup B > 0.1]}$ as follows for every $\delta \in \Delta^{\mathcal{I}_\mathcal{P}}$:

$$\mathbb{F}_{[C \boxtimes q]}^{\mathcal{I}_\mathcal{P}}(\delta) = \begin{cases} 1 & \text{if } C^{\mathcal{I}_\mathcal{P}}(\delta) \boxtimes q \\ C^{\mathcal{I}_\mathcal{P}}(\delta) & \text{otherwise} \end{cases}$$

if $\boxtimes$ is either $\geq$ or $>$, and

$$\mathbb{F}_{[C \leq q]}^{\mathcal{I}_\mathcal{P}}(\delta) = \begin{cases} 1 & \text{if } C^{\mathcal{I}_\mathcal{P}}(\delta) \leq q \\ 1 - C^{\mathcal{I}_\mathcal{P}}(\delta) & \text{otherwise.} \end{cases}$$

It is easy to see that this valuation satisfies the two axioms that define each flag concept.

Additionally, every element $\nu \in \Delta^{\mathcal{I}_\mathcal{P}}$ has exactly one r_i -successor for every $i, 1 \leq i \leq m$, i.e. there is exactly one element $\nu' \in \Delta^{\mathcal{I}_\mathcal{P}}$ such that $r_i^{\mathcal{I}_\mathcal{P}}(\nu, \nu') > 0$. Thus, the interpretation $\mathcal{I}_\mathcal{P}$ trivially satisfies the two conditions of witnessed models.

Suppose now that $\mathcal{O}'_\mathcal{P}$ is not witnessed consistent. Then $\mathcal{I}_\mathcal{P}$ is not a model of $\mathcal{O}'_\mathcal{P}$. Since it is a model of $\mathcal{O}_\mathcal{P}$ and satisfies the axioms defining each of the three flag concepts, it must violate the last axiom added to $\mathcal{T}_\mathcal{P}$. Thus, there is a node $\nu \in \Delta^{\mathcal{I}_\mathcal{P}}$ such that $\mathbb{F}_{[(\neg A) \boxplus B = s/(s+1)]}^{\mathcal{I}_\mathcal{P}}(\nu) = 1$ and $\mathbb{F}_{[A \sqcup B > 0.1]}^{\mathcal{I}_\mathcal{P}}(\nu) = 1$. By our interpretation of these flag concepts, it then follows that $1 - A^{\mathcal{I}_\mathcal{P}}(\nu) + B^{\mathcal{I}_\mathcal{P}}(\nu) = s/(s+1)$, or equivalently, $A^{\mathcal{I}_\mathcal{P}}(\nu) = 0.1 + B^{\mathcal{I}_\mathcal{P}}(\nu)$ which means that $A^{\mathcal{I}_\mathcal{P}}(\nu)$ and $B^{\mathcal{I}_\mathcal{P}}(\nu)$ encode the same word. From the second flag, it follows that they both encode a non-empty word. But then, this same ν must be a solution of $\mathcal{P}$, which finishes the proof. ■

IV. CONCLUSIONS

We have shown that the algorithm from [10] is incorrect, and given an undecidability result for a small extension of fuzzy $\mathcal{ALC}$ with product t-norm and involutive negation. Our results do not show that the logic used in [10] is undecidable since we (i) added expressivity by means of the strict axioms, and (ii) used a weaker notion of witnessed models, which we believe to be more closely related to the idea of concept inclusions in DL. As future work, we will analyse the decidability status of consistency (witnessed consistency, strongly witnessed consistency) for the logic considered in [10], [11] with general concept inclusions. On the one hand, this involves

trying to correct the algorithm from [10] by introducing a more sophisticated blocking condition. On the other hand, we will also try to modify our undecidability proof such that it does not use strict inclusion axioms, or such that it can also deal with pure consistency or strongly witnessed consistency.

Another area for future work includes studying the impact of concept inclusion axioms when reasoning w.r.t. other t-norm based semantics, in particular with the Łukasiewicz t-norm. We will also analyse the applicability of other reasoning techniques to fuzzy DLs. It is important to notice that under the product t-norm semantics, the models of an ontology may, in general, require infinitely many different interpretation values. Hence, the methods for transforming fuzzy DL ontologies into crisp DL ontologies [19], [20], [21] cannot be applied in this setting.

REFERENCES

- [1] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, Eds., *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
- [2] F. Baader and U. Sattler, "An overview of tableau algorithms for description logics," *Studia Logica*, vol. 69, pp. 5–40, 2001.
- [3] T. Lukasiewicz and U. Straccia, "Managing uncertainty and vagueness in description logics for the semantic web," *Journal of Web Semantics*, vol. 6, no. 4, pp. 291–308, 2008.
- [4] U. Straccia, "Description logics with fuzzy concrete domains," in *Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence (UAI'05)*. AUAI Press, 2005, pp. 559–567.
- [5] G. Stoilos, G. B. Stamou, V. Tzouvaras, J. Z. Pan, and I. Horrocks, "The fuzzy description logic f-SHIN," in *Proceedings of Uncertainty Reasoning for the Semantic Web (URSW'05)*, 2005, pp. 67–76.
- [6] U. Straccia, "Reasoning within fuzzy description logics," *Journal of Artificial Intelligence Research (JAIR)*, vol. 14, pp. 137–166, 2001.
- [7] G. Stoilos, U. Straccia, G. B. Stamou, and J. Z. Pan, "General concept inclusions in fuzzy description logics," in *Proceedings of the 17th European Conference on Artificial Intelligence (ECAI'06)*, ser. Frontiers in Artificial Intelligence and Applications, vol. 141. IOS Press, 2006, pp. 457–461.
- [8] U. Straccia and F. Bobillo, "Mixed integer programming, general concept inclusions and fuzzy description logics," in *Proceedings of the 5th EUSFLAT Conference*. Universitas Ostraviensis, 2007, pp. 213–220.
- [9] F. Bobillo and U. Straccia, "On qualified cardinality restrictions in fuzzy description logics under lukasiewicz semantics," in *Proceedings of the 12th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, (IPMU-08)*, 2008, pp. 1008–1015.
- [10] —, "A fuzzy description logic with product t-norm," in *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2007)*. IEEE Press, 2007, pp. 1–6.
- [11] —, "Fuzzy description logics with general t-norms and datatypes," *Fuzzy Sets and Systems*, vol. 160, no. 23, pp. 3382–3402, 2009.
- [12] F. Bobillo, F. Bou, and U. Straccia, "On the failure of the finite model property in some fuzzy description logics," *CoRR*, vol. abs/1003.1588, 2010.
- [13] P. Hájek, "Making fuzzy description logic more general," *Fuzzy Sets and Systems*, vol. 154, no. 1, pp. 1–15, 2005.
- [14] —, "On witnessed models in fuzzy logic," *Mathematical Logic Quarterly*, vol. 53, no. 1, pp. 66–77, 2007. [Online]. Available: <http://dx.doi.org/10.1002/malq.200610027>
- [15] A. García-Cerdáña, E. Armengol, and F. Esteva, "Fuzzy description logics and t-norm based fuzzy logics," *International Journal of Approximate Reasoning*, vol. 51, pp. 632 – 655, July/2010 2010.
- [16] F. Bobillo, F. Bou, and U. Straccia, "On the failure of the finite model property in some fuzzy description logics," *Fuzzy Sets and Systems*, vol. 172, no. 1, pp. 1–12, 2011.
- [17] M. Cerami, F. Esteva, and F. Bou, "Decidability of a description logic over infinite-valued product logic," in *Proceedings of the 12th International Conference on the Principles of Knowledge Representation and Reasoning (KR 2010)*. AAAI Press, 2010, pp. 203–213.
- [18] E. Post, "A variant of a recursively unsolvable problem," *Bulletin of the American Mathematical Society*, vol. 52, pp. 264–268, 1946.
- [19] U. Straccia, "Transforming fuzzy description logics into classical description logics," in *Proceedings of the 9th European Conference on Logics in Artificial Intelligence (JELIA-04)*, ser. Lecture Notes in Computer Science, no. 3229. Lisbon, Portugal: Springer Verlag, 2004, pp. 385–399.
- [20] F. Bobillo, M. Delgado, J. Gómez-Romero, and U. Straccia, "Fuzzy description logics under Gödel semantics," *International Journal of Approximate Reasoning*, vol. 50, no. 3, pp. 494–514, 2009.
- [21] F. Bobillo and U. Straccia, "Reasoning with the finitely many-valued lukasiewicz fuzzy description logic *SRQLQ*," *Information Sciences*, vol. 181, pp. 758–778, 2011.