Syntactic Labelled Tableaux for Łukasiewicz Fuzzy $\mathcal{ALC}$*

Agnieszka Kułacka
Imperial College London

Dirk Pattinson
Imperial College London
The Australian National University

Lutz Schröder
Friedrich-Alexander Universität
Erlangen-Nürnberg

Abstract

Fuzzy description logics (DLs) serve as a tool to handle vagueness in real-world knowledge. There is particular interest in logics implementing Łukasiewicz semantics, which has a number of favourable properties. Current decision procedures for Łukasiewicz fuzzy DLs work by reduction to exponentially large mixed integer programming problems. Here, we present a decision method that stays closer to logical syntax, a labelled tableau algorithm for Łukasiewicz Fuzzy $\mathcal{ALC}$ that calls only on (pure) linear programming, and this only to decide atomic clashes. The algorithm realizes the best known complexity bound, NEXPTIME. Our language features a novel style of fuzzy ABoxes that work with comparisons of truth degrees rather than explicit numerical bounds.

1 Introduction

Fuzzy logic has been introduced as a formalism for handling vagueness in real-world knowledge, which occurs, e.g., in everyday concepts such as 'tall person' and in modern-day notions such as 'good match with my service request'. Consequently, there has been increasing interest in description logics incorporating fuzzy truth values [Straccia, 2001; Lukasiewicz and Straccia, 2008]. A key feature of logics such as fuzzy $\mathcal{ALC}$ is that not only concepts but also roles are fuzzified, i.e. allowed to attain truth values in the unit interval in place of just binary truth values. Typical examples are vague relations such as 'likes', 'matches', or 'is preferred over'. For the underlying propositional logic, one has a wide variety of proposed semantics (see [Metcalfe *et al.*, 2008] for an overview). One widely accepted choice for the interpretation of propositional connectives that leads to a both mathematically well-behaved [Kundu and Chen, 1998] and logically expressive framework is to adopt *Łukasiewicz semantics* where propositional connectives are interpreted in the (universal) MV-algebra $[0, 1]$.

Semantic decision procedures for various Łukasiewicz Fuzzy DLs have been described in the literature [Straccia, 2005; Straccia and Bobillo, 2007]. The best known upper bound to date for Łukasiewicz Fuzzy $\mathcal{ALC}$ is NEXPTIME (it is known that *finite valued* Łukasiewicz modal logic is PSPACE-complete [Bou *et al.*, 2011] but this does not appear to extend easily to the case of infinitely many truth values); this bound has been established in [Schröder and Pattinson, 2011; Cerami and Straccia, 2013] and may also be derived by analysing the complexity of the algorithms presented in [Straccia, 2005; Straccia and Bobillo, 2007]. In all cases, the algorithms work by reducing reasoning problems to exponentially large mixed integer programming problems.

Here, we present an algorithm for Łukasiewicz Fuzzy $\mathcal{ALC}$ that works in a more familiar tableau style; in particular, it handles natural syntactic objects. Specifically, it implements a branching labelled tableau calculus that manipulates ABoxes consisting of linear inequalities over concept and role assertions. Only at the leaves of the tableau tree, we call linear programming (rather than mixed integer programming) to detect clashes among atomic assertions.

We envisage the following benefits of this approach:

- Presenting ABoxes in terms of inequalities among assertions is more appropriate to the goal of representing vague knowledge than the explicit numerical bounds employed previously. Essentially, our logic admits ABoxes saying things such as 'Bob (tragically) likes Alice more than Alice likes Bob' or 'the cumulative aptitude of all team members does not meet the requirements of the project' but does not directly support statements of the form 'John is tall to the degree at least 0.394'. Although truth constants would be easy to add to the calculus (and in fact can be emulated by McNaughton's theorem [McNaughton, 1951]), we explicitly prefer the former style of asserting individual knowledge on the grounds that it is unclear how precise numerical values such as 0.394 would be sensibly determined in an ontology. This corresponds to views expressed in the philosophical analysis of fuzzy truth degrees, see, e.g., [Smith, 2012] and references therein including [Goguen, 1968].
- Our calculus is simple and natural, and as such enables clear proofs of termination and correctness. By comparison, e.g. the blocking condition of [Straccia and Bobillo, 2007] causes incompleteness as noted in [Baader and Peñaloza, 2011] (as it must, since the algorithm works with general TBoxes, which actually cause undecidability [Cerami and Straccia, 2013]; note that the latter work does give a full com-

*Work of the third author forms part of the DFG project *Generic Algorithmic Methods in Modal and Hybrid Logics* (SCHR 1118/5-2)

pleteness proof for the case of acyclic TBoxes).

• While existing algorithms for Łukasiewicz Fuzzy $\mathcal{ALC}$ always fully expand the target concept, and, e.g., will insist on always expanding the existential restriction when checking satisfiability in $A \sqcup \exists R.\, C$, our calculus will, given the right heuristics (e.g. expand simple concepts first, see Example 4.7), be happy to close the tableau early by just satisfying the left disjunct. In other words, previous algorithms always exhibit worst-case behaviour, while our tableau method offers the prospect of feasible average-case behaviour.

Related Work. Besides the mentioned work [Straccia, 2005; Straccia and Bobillo, 2007; Schröder and Pattinson, 2011] on Łukasiewicz Fuzzy $\mathcal{ALC}$, e.g., Straccia [2001] and Stoilos et al. [2007] deal with modal logics of vagueness over other (fuzzy) base logics, in particular using the Zadeh interpretation of propositional connectives. The comparision expressions in [Kang *et al.*, 2006] do not permit general linear inequalities and also use Zadeh semantics for propositional connectives. Proof theory for Łukasiewicz Logic centres around the propositional and first-order setup [Ciabattoni *et al.*, 2005; Olivetti, 2003; Fermüller and Metcalfe, 2009] (despite the title, Montagna [2003] does not treat the modal analog of fuzzy $\mathcal{ALC}$). Our approach is inspired by Hähnle [1994] where satisfiability is reduced to feasibility of linear programming problems. Satisfiability in propositional Łukasiewicz logic is known to be in NP.

2 Syntax and Semantics of Fuzzy $\mathcal{ALC}$

We recall the definition of Łukasiewicz Fuzzy $\mathcal{ALC}$ following [Straccia, 2005] but with a more expressive notion of ABox. The syntax of concepts is the same as in standard $\mathcal{ALC}$, i.e. given sets $\mathsf{N_C}$ and $\mathsf{N_R}$ of *atomic concepts* and *roles*, respectively, *concepts* are defined by the grammar

$$C, D ::= \bot \mid A \mid \neg C \mid C \sqcap D \mid \exists R.\, C \quad (A \in \mathsf{N_C}, R \in \mathsf{N_R}).$$

The nesting depth of $\exists$ in a concept C is called the *quantifier rank* of C, denoted $\mathsf{rk}(C)$. As Łukasiewicz semantics validates the respective equivalences, we can put $C \sqcup D = \neg(\neg C \sqcap \neg D)$, $C \to D = \neg C \sqcup D$, and $\forall R.\, C = \neg\exists R.\, \neg C$ as usual. The semantics of the language is then defined over *fuzzy interpretations* $\mathcal{I} = (\Delta^{\mathcal{I}}, (A^{\mathcal{I}})_{A \in \mathsf{N_C}}, (R^{\mathcal{I}})_{R \in \mathsf{N_R}})$ that consist of

- a set $\Delta^{\mathcal{I}}$, the *domain*;
- a map $A^{\mathcal{I}} : \Delta^{\mathcal{I}} \to [0, 1]$ for each $A \in \mathsf{N_C}$;
- a map $R^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to [0, 1]$ for each $R \in \mathsf{N_R}$;

— i.e. $A^{\mathcal{I}}$ is a fuzzy subset of Δ and $R^{\mathcal{I}}$ a fuzzy binary relation over Δ, for each atomic concept $A \in \mathsf{N_C}$ and each role $R \in \mathsf{N_R}$. We extend the interpretation by fuzzy subsets to all concepts C, defining $C^{\mathcal{I}} : \Delta^{\mathcal{I}} \to [0, 1]$ by

$$(\exists C)^{\mathcal{I}}(d) = \sup_{e \in \Delta^{\mathcal{I}}} (R^{\mathcal{I}}(d, e) \otimes C^{\mathcal{I}}(e)) \qquad \bot^{\mathcal{I}}(d) = 0$$

$$(\neg C)^{\mathcal{I}}(d) = 1 - C^{\mathcal{I}}(d) \qquad (C \sqcap D)^{\mathcal{I}}(d) = C^{\mathcal{I}}(d) \otimes D^{\mathcal{I}}(d)$$

where $a \otimes b = \max(a + b - 1, 0)$. A concept C is *valid* if $C^{\mathcal{I}}(d) = 1$ for all fuzzy interpretations $\mathcal{I}$ and all $d \in \Delta^{\mathcal{I}}$, in which case we write $\models C$. Dually, C is *satisfiable* if $\neg C$ is not valid. Thus, C is satisfiable if $C^{\mathcal{I}}(d) > 0$ for some point $d \in \Delta^{\mathcal{I}}$ in some fuzzy interpretation $\mathcal{I}$.

Łukasiewicz semantics of fuzzy logic has several desirable properties that fail in alternative definitions, in particular residuatedness (i.e. conjunction and implication are mutually adjoint) and continuity of all operators, see e.g. [Kundu and Chen, 1998].

Example 2.1. Maybe somewhat surprisingly, the concept

$$\forall\, \mathsf{likes}.\, \mathsf{tall} \sqcap \exists\, \mathsf{likes}.\, \mathsf{blond} \sqcap \neg\exists\, \mathsf{likes}.\, (\mathsf{tall} \sqcap \mathsf{blond})$$

is satisfiable: take a fuzzy interpretation where $\Delta^{\mathcal{I}}$ contains d, e with $\mathsf{likes}^{\mathcal{I}}(d, e) = 0.5 = \mathsf{tall}^{\mathcal{I}}(e)$, $\mathsf{likes}^{\mathcal{I}}(d, f) = 0$ for $f \neq e$, and $\mathsf{blond}^{\mathcal{I}}(e) = 1$. What is ultimately behind this is the fact that Łukasiewicz Fuzzy logic is not idempotent; in this case: $\mathsf{likes}^{\mathcal{I}}(d, e) \otimes \mathsf{tall}^{\mathcal{I}}(e) = 0.5 \otimes 0.5 = 0$. In other words, Łukasiewicz semantics is 'resource-aware'. As shown in [Kundu and Chen, 1998], the failure of idempotence is unavoidable for logics satisfying certain desirable properties enabled by Łukasiewicz semantics. For many application scenarios, this is a feature rather than a bug; e.g. when casting a movie role for which we want a talented and popular actor ($\mathsf{T} \sqcap \mathsf{P}$), we might indeed be inclined to regard someone who displays only moderate talent ($\mathsf{T} = 0.5$) *and* is only mildly popular ($\mathsf{P} = 0.5$) as altogether unsuitable ($\mathsf{T} \sqcap \mathsf{P} = 0$).

We will reduce the problem of concept satisfiability to that of ABox satisfiability where we take ABoxes to be sets of linear inequalities relating membership of named individuals in concepts and roles. Formally, given a set $\mathsf{I_N}$ of individual names, an *ABox assertion* is a linear inequality

$$\sum_{i \in I} a_i : C_i + \sum_{j \in J} (a_j, b_j) : R_j >_z \sum_{k \in K} a_k : C_k + \sum_{l \in L} (a_l, b_k) : R_l$$

where subscripted occurrences of a, b, C and R are individual names, concepts and role names, respectively and $z \in \mathbb{Z}$ is an integer, with $n >_z m$ abbreviating $n + z > m$. An *ABox* is a finite set of ABox assertions. Terms of the form $a : C$ and $(a, b) : R$ as in ABox assertions are called *concept assertions* and *role assertions*, respectively. If $\Gamma = \sum_{i \in I} a_i{:}C_i + \sum_{j \in J} (a_j, b_j){:}R_j$ is a formal sum of concept and role assertions and $\mathcal{I}$ additionally assigns elements $i^{\mathcal{I}}$ to individuals, we put $\Gamma^{\mathcal{I}} = \sum_{i \in I} C_i^{\mathcal{I}}(a_i^{\mathcal{I}}) + \sum_{j \in J} R^{\mathcal{I}}(a_j^{\mathcal{I}}, b_j^{\mathcal{I}})$ and say that $\mathcal{I}$ *satisfies* an ABox assertion $\Gamma >_z \Delta$ if $z + \Gamma^{\mathcal{I}} > \Delta^{\mathcal{I}}$. That is, we require the linear inequality to hold if concepts and roles are replaced by their meaning. An interpretation $\mathcal{I}$ satisfies an ABox $\mathcal{A}$ if it satisfies all elements, this is written $\mathcal{I} \models \mathcal{A}$. Finally, an ABox $\mathcal{A}$ is *satisfiable* if $\mathcal{I} \models \mathcal{A}$ for some interpretation $\mathcal{I}$. We use standard notational conventions and denote unions of ABoxes in the form $\mathcal{A}_1 \mid \mathcal{A}_2$.

Remark 2.2. There are a number of alternative options regarding the particular shape of ABoxes: we could have used non-strict rather than strict inequality (or a mix of both), and we could have admitted scalars in linear inequalities. While McNaughton's Theorem [McNaughton, 1951] shows that rational scalars are redundant technically, our argument for not including them is philosophical, as statements like 'my hat is

1.27 times as red as yours' seem to be of only limited meaningfulness. The formulation of ABoxes in terms of strict inequalities dualizes the validity problem, and admitting non-strict inequalities in addition to strict inequalities poses no technical difficulties.

Example 2.3. While McNaughton's Theorem [McNaughton, 1951] implicitly allows us to formulate linear inequalities involving satisfaction of concepts at a single individual in the style used in previous frameworks, i.e. as simple lower bounds $a : C > \alpha$, the formulation of ABoxes as linear inequalities is strictly more powerful as it can also describe relations between role assertions and relations between concepts satisfied by different individuals. Our notion of ABox may thus be seen as a fuzzy variant of Boolean ABoxes [Areces *et al.*, 2003; Liu *et al.*, 2006]. Using linear inequalities in ABoxes, we can e.g. express Bob's tragic plight as in the introduction using the ABox assertion

$$(\mathsf{Bob}, \mathsf{Alice}) : \mathsf{likes} > (\mathsf{Alice}, \mathsf{Bob}) : \mathsf{likes}.$$

Similarly, the inadequacy of the team consisting of Alice, Bob, and Charlie to the task at hand can be expressed by

$$\begin{aligned}&\mathsf{Project} : \exists \mathsf{demands}.\, \mathsf{Skill} > \\ &\qquad \mathsf{Bob} : \mathsf{compt} + \mathsf{Alice} : \mathsf{compt} + \mathsf{Charlie} : \mathsf{compt}\end{aligned}$$

where compt abbreviates $\exists \mathsf{has}.\, \mathsf{Skill}$.

3 The Tableau Calculus

We now introduce the tableau calculus for Łukasiewicz Fuzzy $\mathcal{ALC}$. Semantically, the main difficulty is that although Fuzzy $\mathcal{ALC}$ does have the tree model property [Schröder and Pattinson, 2011], one has to deal with arithmetic dependencies between different branches in the model. Syntactically these dependencies are captured using labels in ABox assertions. The primitive objects handled in our calculus are not concepts but instead *linear inequalities* between concepts and roles, i.e. ABox assertions which play a role similar to that of concept assertions and role assertions in the classical case. In particular, satisfaction of an ABox assertion under an interpretation is two-valued in the same way as satisfaction of concept and role assertions in the classical case.

We introduce a Beth-style tableau calculus for determining satisfiability of ABoxes and prove its soundness and completeness, i.e. unsatisfiability of an ABox $\mathcal{A}$ is equivalent to existence of a closed tableau with root $\mathcal{A}$.

As axioms of our calculus, we take those ABoxes whose unsatisfiability can be seen purely by reasoning with linear inequalities. In particular, we can take those ABoxes to consist of atomic concepts and role assertions only.

Definition 3.1. An ABox assertion A is *atomic* if C is an atomic concept for every $a{:}C$ that occurs in A, and an ABox is atomic if it consists of atomic ABox assertions only.

In particular, the unsatisfiability of atomic ABoxes immediately reduces to a linear programming problem.

Lemma and Definition 3.2 (Clashes)**.** *For an atomic ABox $\mathcal{A}$, the following are equivalent:*

1. *$\mathcal{A}$ is unsatisfiable*
2. *the linear programming problem in the variables $a{:}A$ and $(a,b){:}R$ consisting of $\mathcal{A}$ and inequalities stating that all variables are in $[0,1]$ is infeasible.*

We say that $\mathcal{A}$ clashes *if it satisfies these conditions.*

The calculus that we are about to introduce consists of propositional rules that are invertible as in the case of classical propositional logic, together with left and right rules for the existential restriction operator. As in labelled modal calculi, the existential rule manipulates the labels whereas propositional rules leave labels intact.

Definition 3.3 (Linear Tableau Calculus)**.** The *linear tableau calculus* (LTC) for Łukasiewicz Fuzzy $\mathcal{ALC}$ consists of the rules displayed in Figure 1.

If (R) is a rule of LTC with premiss P and conclusions $C_1, \dots, C_n$, then

$$\frac{P}{P \mid C_1 \qquad \dots \qquad P \mid C_n}$$

is an *instance* of (R) provided each $P \mid C_i$ is distinct from P. We say that an ABox $\mathcal{A}$ is *inconsistent* if there exists a *closed tableau for* $\mathcal{A}$, i.e. a tree constructed using rule instances of LTC with root $\mathcal{A}$ and all leaves labelled (Ax), and $\mathcal{A}$ is *incomplete* if there is a rule instance with premiss $\mathcal{A}$, and *complete*, otherwise.

The fact that tableaux are defined by applying rule *instances* means that the rules in Figure 1 are to be read in the form 'if the ABox in the premiss has been reached, then add the conclusion to it'. As usual, a *branch* in the tableau is a sequence of ABoxes such that each successive ABox is a conclusion of a rule instance applied to the previous ABox.

In the propositional rules, one can see branching familiar from propositional tableau. The index-shift between premiss and conclusion of propositional rules absorbs the arithmetic that stems from the interpretation of the propositional connectives and is used here mainly for cosmetic reasons. The existential rules are more subtle, but they achieve the same as their counterparts in classical (crisp) $\mathcal{ALC}$. For the premiss of (∃L) to be satisfiable, the truth value of the existential restriction needs to be maximized. This maximal value (up to ϵ owing to the interpretation of $\exists$ as supremum) needs to be realized by a successor, and the value of $a : \exists R.C$ is either zero (leading to the left conclusion) or achieved (again up to ϵ) for some successor b as indicated in the right conclusion. The side condition for (∃L) ensures termination of the calculus and guarantees that it is only applied once to each concept assertion $a : \exists R.\, C$. The (∃R) rule plays the role of the universal restriction rule in classical $\mathcal{ALC}$, and we need to minimize the truth value of the existential restriction $a{:}\exists R.C$ on the right for the premise of (∃R) to be satisfied. This is achieved by equipping a with as few successors as possible, i.e., we only consider those successors the existence of which is stipulated by the ABox (and assign 0 as transition weight to all other successors). Again, in the conclusion we distinguish cases depending on whether a successor contributes positively to the truth value of the existential restriction. The last ABox assertion in the premiss of (∃R) serves only to restrict attention to labels b such that $(a,b) : R$ occurs on the left in some $\Gamma' + (a, b) : R >_z \Delta'$; Γ', Δ', and z' themselves are then irrelevant for the rule. Note that rule instances copy the premiss to the conclusion. In particular, $(\exists\mathsf{R})$ does *not* discard the existential restriction as further relational successors may emerge due to concept unfolding. We illustrate the rules before establishing soundness and completeness.

Example 3.4. Anticipating soundness, we show that $\forall R.p \sqcap q \rightarrow \forall R.p$ is universally valid in Łukasiewicz $\mathcal{ALC}$ by demonstrating that the ABox

$$a : \neg\exists R.\neg(p \sqcap q) \sqcap \exists R.\neg p >_0 \emptyset$$

is inconsistent. Using the propositional rules, this is readily reduced to inconsistency of

$$a{:}\exists R.\neg p >_0 a{:}\exists R.\neg(p \sqcap q)$$

and that of $\emptyset >_0 \emptyset$, which is immediate. Applying $(\exists\mathsf{L})$ we obtain two conclusions, $\emptyset >_0 a : \exists R.\neg(p \sqcap q)$, which eventually clashes, and

$$(a, b){:}R >_0 b{:}p + a{:}\exists R.\neg(p \sqcap q)$$

where we have already applied the $(\neg\mathsf{L})$-rule. Application of $(\exists\mathsf{R})$ leads to the above ABox together with

$$(a, b){:}R >_0 b{:}p \mid (a, b){:}R >_1 b{:}p + (a, b){:}R + b{:}\neg(p \sqcap q),$$

which clashes after propositional reasoning: the right hand assertion is equivalent to $b{:}p \sqcap q > b{:}p$ and an application of $(\sqcap\mathsf{L})$ introduces two inconsistent branches: $0 > b{:}p$ and $b{:}q > 1$.

Remark 3.5. Among the existing algorithms for Łukasiewicz Fuzzy $\mathcal{ALC}$, the most closely related to ours are the constraint set algorithms by Straccia [2005] and Straccia and Bobillo [2007]. These algorithms handle constraints of the form $\langle a : C, l\rangle$ to be read in our syntax as $a{:}C \geq l$, where l is either a mixed integer linear expression in so-called control variables or, in [Straccia and Bobillo, 2007], a more restricted form of expression, and additionally handle inequalities between such expressions introduced by the algorithm. Upon completion, they solve exponentially large mixed integer programming problems. It is unclear precisely how the algorithm of Straccia and Bobillo [2007], originally designed to decide satisfiability over arbitrary TBoxes (but in fact incomplete for this case [Baader and Peñaloza, 2011]) behaves in the case of the empty TBox. Similarly, the algorithm in [Straccia, 2005] does not provide an a priori bound on the number of integer linear constraints that can appear (we provide such a bound for our calculus in the shape of Lemma 5.2 below), so that the termination argument remains somewhat unclear. (Note however a recently published more detailed discussion [Cerami and Straccia, 2013].)

By comparison, our algorithm only needs to handle pure linear inequalities (i.e. without integer variables), which it integrates into the actual syntax of ABoxes, so that the objects handled by the calculus always remain within the original syntax. It only calls linear programming to detect clashes, handling the branching on its own (see also Example 4.7). Finally, it has a clear and simple completeness proof and complexity analysis, which in fact we are going to present essentially in full.

4 Soundness and Completeness

Soundness (and indeed completeness) of the propositional rules is straightforward by the following result that additionally asserts strong invertibility: the premiss of a LTC-rule is unsatisfiable *if and only if* this holds for all conclusions.

Lemma 4.1 (Propositional Invertibility)**.** *Let $\mathcal{A}_0/\mathcal{A}_1 \dots \mathcal{A}_n$ be a propositional rule of* LTC*. Then $\mathcal{A}_0$ is unsatisfiable iff all $\mathcal{A}_i$ are unsatisfiable, for all $1 \leq i \leq n$.*

The soundness of the existential rules is most easily seen by considering the contrapositive statement. Our proof is elementary and could be slightly simplified by assuming witnessed models [Hájek, 2005].

Lemma 4.2. *If the premiss of an instance of $(\exists\mathsf{L})$ or $(\exists\mathsf{R})$ is satisfiable, then so is (at least) one of the conclusions.*

$$(\mathsf{Ax})\frac{\mathcal{A} \mid \mathcal{A}'}{} \quad (\mathcal{A}\text{ clashes}) \qquad (\neg\mathsf{L})\frac{\mathcal{A} \mid a{:}\neg C + \Gamma >_z \Delta}{\Gamma >_{z+1} a{:}C + \Delta} \qquad (\neg\mathsf{R})\frac{\mathcal{A} \mid \Gamma >_z \neg a{:}C + \Delta}{\Gamma + a{:}C >_{z-1} \Delta}$$

$$(\sqcap\mathsf{L})\frac{\mathcal{A} \mid a{:}C \sqcap D + \Gamma >_z \Delta}{a{:}C + a{:}D + \Gamma >_{z-1} \Delta \qquad \Gamma >_z \Delta} \qquad (\sqcap\mathsf{R})\frac{\mathcal{A} \mid \Gamma >_z a{:}C \sqcap D + \Delta}{\Gamma >_z \Delta \mid \Gamma >_{z+1} a{:}C + a{:}D + \Delta}$$

$$(\exists\mathsf{L})\frac{\mathcal{A} \mid \Gamma + a{:}\exists R.C >_z \Delta}{\Gamma >_z \Delta \qquad \Gamma + (a, b){:}R + b{:}C >_{z-1} \Delta}(*)$$

$$(\exists\mathsf{R})\frac{\mathcal{A} \mid \Gamma >_z \Delta + a{:}\exists R.C \mid \Gamma' + (a, b){:}R >_{z'} \Delta'}{\Gamma >_z \Delta \mid \Gamma >_{z+1} \Delta + (a, b){:}R + b{:}C}$$

$(*)$ if $\{\Gamma + (a, c){:}R + c{:}C >_{z-1} \Delta \mid c \in \mathsf{I_N}\} \cap \mathcal{A} = \emptyset$ and b does not occur in $\mathcal{A}$, Γ or Δ

Figure 1: Rules of LTC

Proof. We first consider (∃L) and fix an interpretation $\mathcal{I}$ for which $z+\Gamma^{\mathcal{I}}+(a{:}\exists R.C)^{\mathcal{I}} \geq \epsilon+\Delta^{\mathcal{I}}$ for some $\epsilon > 0$. Clearly the left conclusion is satisfiable in case $(a{:}\exists R.C)^{\mathcal{I}} = 0$, so assume that $(a{:}\exists R.C)^{\mathcal{I}} > 0$. Then there exists $y \in \Delta$ such that both $R^{\mathcal{I}}(x,y) \otimes C^{\mathcal{I}}(y) > 0$ where we have written $x = a^{\mathcal{I}}$, and $R^{\mathcal{I}}(x,y) \otimes C^{\mathcal{I}}(y) > (\exists R.C)^{\mathcal{I}}(x) - \epsilon$. By definition of $\otimes$ this gives $R^{\mathcal{I}}(x,y) + C^{\mathcal{I}}(y) - 1 > (\exists R.C)^{\mathcal{I}}(x) - \epsilon$ so that in total $(z-1) + \Gamma^{\mathcal{I}} + R^{\mathcal{I}}(x,y) + C^{\mathcal{I}}(y) = z + \Gamma^{\mathcal{I}} + R^{\mathcal{I}}(x,y) \otimes C^{\mathcal{I}}(y) > z + \Gamma^{\mathcal{I}} + (\exists R.C)^{\mathcal{I}}(x) - \epsilon \geq \epsilon + \Delta^{\mathcal{I}} - \epsilon = \Delta^{\mathcal{I}}$ as we had to show.

For soundness of (∃R) we fix an interpretation $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{A}$, $z + \Gamma^{\mathcal{I}} > \Delta^{\mathcal{I}} + (a{:}\exists R.C)^{\mathcal{I}}$ and $z' + \Gamma'^{\mathcal{I}} + R^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}}) > \Delta'^{\mathcal{I}}$. It suffices to show that $z+1+\Gamma^{\mathcal{I}} > \Delta^{\mathcal{I}} + R^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}}) + C^{\mathcal{I}}(b^{\mathcal{I}})$. This follows from $(\exists R.C)^{\mathcal{I}}(a^{\mathcal{I}}) \geq R^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}}) \otimes C^{\mathcal{I}}(b^{\mathcal{I}}) \geq R^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}}) + C^{\mathcal{I}}(b^{\mathcal{I}}) - 1$. ☐

Soundness of LTC with respect to many-valued interpretations is now an easy corollary of the contrapositive of the previous lemma.

Proposition 4.3 (Soundness of LTC)**.** *Every inconsistent ABox is unsatisfiable.*

We establish completeness of LTC by showing that the existential rules are in fact invertible and that the calculus does not generate infinite branches in backwards proof search, essentially the same argument that is also used to show completeness and termination in classical $\mathcal{ALC}$. Completeness follows from invertibility once we demonstrate that every unsatisfiable sequent either clashes or is a rule premiss and the fact that all branches in tableaux are finite. As to the former, it is straightforward to show that

Lemma 4.4. *If $\mathcal{A}$ is clash-free and complete, then $\mathcal{A}$ is satisfiable.*

Termination of the calculus is a consequence of the fact that only finitely many fresh labels will be generated using the (∃L)-rule as every newly introduced individual can only appear in concept assertions of smaller quantifier rank, and each label is annotated with sub-concepts of concepts that occur in the initial ABox, an argument similar to that used for termination of hybrid tableaux in [Bolander and Blackburn, 2007].

Lemma 4.5. *Every path in* LTC *is of finite length only.*

Proof. As all individuals are only labelled with sub-concepts of concepts that occur in $\mathcal{A}_0$ and the size of ABox-assertions is non-increasing, every infinite branch $(\mathcal{A}_0, \mathcal{A}_1, \dots)$ must contain an infinite number of applications of (∃L). Let $\mathcal{A} = \bigcup_{i \geq 0} \mathcal{A}_i$. Consider the forest of individual-labelled trees induced by the role assertions $(a,b) : R$ generated by (∃L). The restriction on applicability of (∃L) guarantees that all these trees are finitely branching, so that by König's lemma, there must be at least one tree with an infinite branch. In other words, there is an infinite sequence of (fresh) individuals $(a_0, a_1, \dots)$ such that $(a_i, a_{i+1}){:}R \in \mathcal{A}$ is induced by an application of (∃L). On the other hand, for every role assertion (a,b) introduced by (∃L) and every $b{:}D \in \mathcal{A}$ we have that

$$\mathsf{rk}(D) < \max\{\mathsf{rk}(C) \mid a{:}C \in \mathcal{A}\}$$

(recall that $\mathsf{rk}(\cdot)$ denotes quantifier rank). That is, the maximal quantifier rank decreases along the infinite chain $(a_0, a_1, \dots)$ constructed above, contradiction. ☐

Completeness is now an immediate consequence of the previous observations.

Theorem 4.6 (Completeness of LTC)**.** *Every unsatisfiable ABox is inconsistent, and every satisfiable ABox is satisfiable in an interpretation with finite domain.*

Proof. Suppose that $\mathcal{A}$ is an unsatisfiable ABox that is *not* inconsistent. In particular, it does not contain a clash, so we can apply a rule by Lemma 4.4. By invertibility, at least one of the premisses is again an unsatisfiable, but not inconsistent, ABox. Iterating this, we obtain an infinite branch, contradicting Lemma 4.5. The small model property also follows from Lemma 4.5 which in particular guarantees that every satisfiable ABox is expanded to a finite ABox to which no more rules apply. ☐

One advantage of our calculus over previous algorithms, which always exhibit worst-case behaviour, is that it allows for heuristic optimization. A very simplistic example is the following.

Example 4.7. Consider the example from the introduction, $A \sqcup \exists R.\, C$, where A is atomic and C is a concept that generates exponentially many individuals (such concepts exist also in the fuzzy case, as already observed by Straccia [2005]). Satisfiability of this concept translates into satisfiability of the ABox $a : A \sqcup \exists R.\, C >_0 0$, which unfolds, upon decoding the disjunction and subsequent propositional reasoning, to satisfiability of $a : A + a : \exists R.\, C >_0 0$. Our calculus can then finish successfully on the spot by choosing the left branch in $(\exists L)$. By comparison, the algorithms of [Straccia, 2005; Straccia and Bobillo, 2007; Schröder and Pattinson, 2011] will always expand the existential restriction and hence generate exponentially many labels.

5 Complexity and the Small Model Property

It follows from Theorem 4.6 in conjunction with Lemma 4.5 that ABox-satisfiability is decidable in Łukasiewicz $\mathcal{ALC}$. We obtain an upper complexity bound by carefully dissecting the length of branches and the size of ABoxes. The crucial observation is that all rules of LTC do not increase the size of ABox-assertions except possibly for an increase of the size of the index. The measure that we use in the analysis of the length of branches, the *order* of an ABox assertion, compensates for this by weighing the size of concepts with a factor of two. Formally:

Definition 5.1 (Size and Order)**.** The *size* of a concept C, $\mathsf{size}(C)$, is the number of its sub-concepts, and $\mathsf{ord}(C) = 2 \cdot \mathsf{size}(C)$ is its *order*. Size and order of an ABox-assertions $A = \Gamma >_z \Delta$ are the sum of size (resp. order) of all concepts that occur in A, additionally counting 1 for every role assertion $(a,b){:}R$ in A plus $\lceil \log_2 |z| + 1 \rceil$ for the index. Size and order of an ABox are the sum of the size (resp. order) of all ABox assertions contained in $\mathcal{A}$.

Note that $\mathsf{size}(A) \leq \mathsf{ord}(A) \leq 2 \cdot \mathsf{size}(A)$. We start with the following easy observation concerning order:

Lemma 5.2. *If (R) is a rule instance of* LTC *with premiss P and conclusions $C_1, \ldots, C_n$ then $\max\{\mathsf{ord}(A) \mid A \in C_i\} \leq \max\{\mathsf{ord}(A) \mid A \in P\}$ for all $1 \leq i \leq n$.*

In other words, the order (but not the size) of ABox assertions is non-decreasing along every branch. As far as we can see, it is essentially this fact that is crucially missing from the argumentation in [Straccia, 2005; Straccia and Bobillo, 2007], where the accompanying integer linear constraints lead a somewhat hidden life.

Our second observation is that the number of fresh individuals introduced by ($\exists$L) is at most exponential in the size of the initial ABox.

Lemma 5.3. *Let $\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_k$ be a path in* LTC. *Then the number of individuals in any $\mathcal{A}_i$ is exponentially bounded by* $\mathsf{size}(\mathcal{A}_0)$.

Proof. As every new individual can only be labelled with sub-concepts of concepts occurring in $\mathcal{A}_0$, the side condition on ($\exists$L) ensures that every individual only produces a number of descendants that is linear in $\mathsf{size}(\mathcal{A}_0)$, i.e. the cardinality of the set $\{b \mid (a, b){:}R \in \mathcal{A}_k\}$ is linearly bounded by $\mathsf{size}(\mathcal{A}_0)$. As every individual that is created using ($\exists$L) is only annotated with formulae of smaller quantifier rank, this results in a set of trees, as in the proof of Lemma 4.5, with linear height and linear branching, giving an exponential number of individuals in the worst case. □

The complexity estimate is now a consequence of the observation that one can only produce exponentially many different ABoxes using exponentially many labels.

Lemma 5.4. *The length of every path $(\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_k)$ in* LTC *is at most exponential in* $\mathsf{ord}(\mathcal{A}_0)$.

Proof. Note that $\mathcal{A}_0 \subseteq \mathcal{A}_1 \subseteq \mathcal{A}_n$ by the definition of rule instance, and that all inclusions are strict (Definition 3.3). As the order of all ABox assertions in the path is bounded by $\max\{\mathsf{ord}(A) \mid A \in \mathcal{A}_0\}$, i.e. linear in the size of $\mathcal{A}_0$, the claim follows from the fact that there can be at most exponentially many ABox-assertions of this (maximal) size in the path: we form a linear-sized expression from an alphabet, consisting of logical symbols, concept and role names, and labels, that has at most exponential size by Lemma 5.3. □

The announced complexity bound for ABox-consistency is now a consequence of the length of the branches.

Theorem 5.5. *ABox-consistency in* LTC *is in* NEXPTIME.

Proof. By the above, consistency of an ABox $\mathcal{A}$ can be solved non-deterministically by applying a rule instance and then guessing a conclusion. After at most exponentially many steps, this leads to a linear programming problem of at most exponential size (exponentially many ABoxes, each of which satisfies $\mathsf{ord}(\mathcal{A}') \leq \mathsf{ord}(\mathcal{A}) \leq 2\mathsf{size}(\mathcal{A})$); we are done since linear programming is in P [Sontag, 1985]. □

The complexity of Łukasiewicz Fuzzy $\mathcal{ALC}$ now follows from completeness (Theorem 4.6).

Corollary 5.6. *ABox-satisfiability in Łukasiewicz $\mathcal{ALC}$ is decidable in* NEXPTIME. *In particular, universal validity in Łukasiewicz $\mathcal{ALC}$ is decidable in* CONEXPTIME.

As an aside, Lemma 5.3 also allows us to improve the finite model property stated in Theorem 4.6 to a small model property.

Corollary 5.7. *Every satisfiable ABox $\mathcal{A}$ is satisfied by an interpretation with domain bounded exponentially by* $\mathsf{size}(\mathcal{A})$.

Remark 5.8. While it would be desirable to support also TBox axioms such as the axiom

$$\mathsf{young} \sqcap \exists R.\, \mathsf{risk} \sqsubseteq \exists R.\, \mathsf{sportsCar},$$

which states that young people that like to take risks also tend to like sports cars (where we regard all concepts as vague, including the concept sportsCar, which would apply to a Ferrari to a higher degree than to a Vauxhall Tigra), it has been shown that unrestricted TBoxes lead to undecidability in Łukasiewicz Fuzzy $\mathcal{ALC}$ [Baader and Peñaloza, 2011; Cerami and Straccia, 2013] (while in the finite-valued case, decidability extends even to very expressive logics such Łukasiewucz fuzzy $\mathcal{SROIQ}$ [Bobillo and Straccia, 2011]). On the other hand, we expect no essential difficulties in accommodating so-called *acyclic* TBoxes by the standard technique of on-the-fly expansion [Lutz, 1999]. Here, a TBox is called *acyclic* if its axioms all have the form $A = C$ where A is atomic and C is called the *definition of A* such that every atomic concept has at most one definition and the relation 'appears in the definition of' on atomic concepts is acyclic.

6 Conclusions

We have introduced a labelled tableau calculus for Łukasiewicz Fuzzy $\mathcal{ALC}$, which we have shown to be sound (Proposition 4.3) and complete (Theorem 4.6). The calculus realizes the best known upper complexity bound NEXPTIME (Theorem 5.5). The calculus generalizes previous calculi by allowing *linear inequalities* in ABoxes which, for the first time, permit the comparison of truth degrees across individuals. As a by-product we obtain a new proof of a previously known result, the small model property of Łukasiewicz Fuzzy $\mathcal{ALC}$ (Corollary 5.7). It is known that reasoning with general TBoxes destroys both the finite model property [Bobillo *et al.*, 2011] and decidability [Cerami and Straccia, 2013; Baader and Peñaloza, 2011]. As a consequence, ABox reasoning, as treated here, together with acyclic TBoxes that are expanded on the fly, is the best we can hope for. Compared to previous decision procedures, our approach has the following main advantages:

- it is strictly more expressive (due to the use of linear inequalities) while retaining the same complexity as other algorithms as demonstrated in Example 2.3
- the complexity bound of our calculus is a true worst case bound that can be often be avoided in practical examples, whereas all other calculi known to date produce integer programming problems that are as large as the full tableau tree, i.e. typically exponential, see Remark 3.5.

The most pressing remaining open question is the actual optimality of the complexity bound, i.e. to show that concept satisfiability in Łukasiewicz Fuzzy $\mathcal{ALC}$ is NEXPTIME-hard.

References

[Areces *et al.*, 2003] Carlos Areces, Patrick Blackburn, Bernadette Martínez Hernández, and Maarten Marx. Handling Boolean ABoxes. In *Description Logics*, volume 81 of *CEUR Workshop Proceedings*, 2003.

[Baader and Peñaloza, 2011] Franz Baader and Rafael Peñaloza. Are fuzzy description logics with general concept inclusion axioms decidable? In *Fuzzy Systems, FUZZ-IEEE 2011*, pages 1735–1742. IEEE, 2011.

[Bobillo and Straccia, 2011] Fernando Bobillo and Umberto Straccia. Reasoning with the finitely many-valued Łukasiewicz fuzzy description logic $\mathcal{SROIQ}$. *Inform. Sci.*, 181:758–778, 2011.

[Bobillo *et al.*, 2011] Fernando Bobillo, Félix Bou, and Umberto Straccia. On the failure of the finite model property in some fuzzy description logics. *Fuzzy Sets and Systems*, 172(1):1–12, 2011.

[Bolander and Blackburn, 2007] Thomas Bolander and Patrick Blackburn. Termination for hybrid tableaus. *J. Log. Comput.*, 17(3):517–554, 2007.

[Bou *et al.*, 2011] Félix Bou, Marco Cerami, and Francesco Esteva. Finite-valued Łukasiewicz modal logic is PSPACE-complete. In *International Joint Conference on Artificial Intelligence, IJCAI 2011*, pages 774–779. IJCAI/AAAI, 2011.

[Cerami and Straccia, 2013] Marco Cerami and Umberto Straccia. On the (un)decidability of fuzzy description logics under Łukasiewicz t-norm. *Information Sciences*, 227:1–21, 2013.

[Ciabattoni *et al.*, 2005] Agata Ciabattoni, Christian G. Fermüller, and George Metcalfe. Uniform rules and dialogue games for fuzzy logics. In Franz Baader and Andrei Voronkov, editors, *Proc. LPAR 2004*, volume 3452 of *Lecture Notes in Computer Science*, pages 496–510. Springer, 2005.

[Fermüller and Metcalfe, 2009] Christian G. Fermüller and George Metcalfe. Giles's game and the proof theory of lukasiewicz logic. *Studia Logica*, 92(1):27–61, 2009.

[Goguen, 1968] Joseph Goguen. The logic of inexact concepts. *Synthese*, 19:325–373, 1968.

[Hähnle, 1994] Reiner Hähnle. Many-valued logic and mixed integer programming. *Annals of Mathematics and Artificial Intelligence*, 12:231–263, 1994.

[Hájek, 2005] Petr Hájek. Making fuzzy description logic more general. *Fuzzy Sets and Systems*, 154:1–15, 2005.

[Kang *et al.*, 2006] Dazhou Kang, Baowen Xu, Jianjiang Lu, and Yanhui Li. Reasoning for A fuzzy description logic with comparison expressions. In *Proc. Description Logic 2006*, volume 189 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2006.

[Kundu and Chen, 1998] Sukhamay Kundu and Jianhua Chen. Fuzzy logic or Łukasiewicz logic: A clarification. *Fuzzy Sets and Systems*, 95:369 – 379, 1998.

[Liu *et al.*, 2006] Hongkai Liu, Carsten Lutz, Maja Milicic, and Frank Wolter. Updating description logic aboxes. In *Principles of Knowledge Representation and Reasoning, KR 2006*, pages 46–56. AAAI Press, 2006.

[Lukasiewicz and Straccia, 2008] Thomas Lukasiewicz and Umberto Straccia. Managing uncertainty and vagueness in description logics for the semantic web. *J. Web Sem.*, 6(4):291–308, 2008.

[Lutz, 1999] Carsten Lutz. Complexity of terminological reasoning revisited. In *Logic Programming and Automated Reasoning, LPAR 99*, volume 1705 of *LNCS*, pages 181–200. Springer, 1999.

[McNaughton, 1951] R. McNaughton. A theorem about infinite-valued sentential logic. *Journal of Symbolic Logic*, 16:1–13, 1951.

[Metcalfe *et al.*, 2008] G. Metcalfe, N. Olivetti, and D. Gabbay. *Proof Theory for Fuzzy Logics*. Number 36 in Series in Applied Logic. Springer, 2008.

[Montagna and Sacchetti, 2003] Franco Montagna and Lorenzo Sacchetti. Kripke-style semantics for many-valued logics. *Math. Log. Q.*, 49(6):629–641, 2003.

[Olivetti, 2003] Nicola Olivetti. Tableaux for łukasiewicz infinite-valued logic. *Stud. Log.*, 73:81–111, 2003.

[Schröder and Pattinson, 2011] L. Schröder and D. Pattinson. Description logics and fuzzy probability. In *International Joint Conference on Artificial Intelligence, IJCAI 2011*, pages 1075–1080. AAAI Press, 2011.

[Smith, 2012] Nicholas Smith. Measuring and modeling truth. *Amer. Philos. Quarterly*, 49:345–56, 2012.

[Sontag, 1985] Eduardo Sontag. Real addition and the polynomial hierarchy. *Inf. Proc. Lett.*, 20:115–120, 1985.

[Stoilos *et al.*, 2007] Giorgos Stoilos, Giorgos Stamou, Jeff Pan, Vassilis Tzouvaras, and Ian Horrocks. Reasoning with very expressive fuzzy description logics. *J. Artif. Intell. Res.*, pages 273–320, 2007.

[Straccia and Bobillo, 2007] Umberto Straccia and Fernando Bobillo. Mixed integer programming, general concept inclusions and fuzzy description logics. In *New Dimensions in Fuzzy Logic and Related Technologies, EUSFLAT 2007*, pages 213–220. Universitas Ostraviensis, 2007.

[Straccia, 2001] U. Straccia. Reasoning within fuzzy description logics. *J. Artif. Intell. Res.*, 14:137–166, 2001.

[Straccia, 2005] Umberto Straccia. Description logics with fuzzy concrete domains. In *Uncertainty in Artificial Intelligence, UAI 2005*, pages 559–567. AUAI Press, 2005.