Non-expansive Fuzzy $\mathcal{ALC}$

28th January 2025

Stefan Gebhart, Lutz Schröder, Paul Wild and Philipp Hermes

Oberseminar WS2024/25

Lehrstuhl für Theoretische Informatik 8
Friedrich-Alexander-Universität Erlangen-Nürnberg

T.CS

FAU Friedrich-Alexander-Universität
Faculty of Engineering

Fuzzy description languages

Friedrich-Alexander-Universität
Faculty of Engineering

Classical description languages

- A *signature* consists of a set N_C of atomic concepts and a set N_R of role names

Classical description languages

- A *signature* consists of a set $\mathsf{N_C}$ of atomic concepts and a set $\mathsf{N_R}$ of role names
- *Concepts* are generated by:

$$C, D ::= \bot \mid p \mid \neg C \mid C \wedge D \mid \exists R.C$$

where $p \in \mathsf{N_C}$ and $R \in \mathsf{N_R}$

Classical description languages

- A *signature* consists of a set $\mathsf{N_C}$ of atomic concepts and a set $\mathsf{N_R}$ of role names
- *Concepts* are generated by:

$$C, D ::= \bot \mid p \mid \neg C \mid C \wedge D \mid \exists R.C$$

  where $p \in \mathsf{N_C}$ and $R \in \mathsf{N_R}$
- An *interpretation* $\mathcal{I}$ consists of a set $\Delta^{\mathcal{I}}$ of *individuals*, a set $p^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ for all $p \in \mathsf{N_C}$ and a set $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ for all $R \in \mathsf{N_R}$

Classical description languages

- The *extension* $(\cdot)^{\mathcal{I}}$ of concepts is defined by:

$$\bot^{\mathcal{I}} = \emptyset \qquad (\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \qquad (C \wedge D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$$

$$(\exists R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \exists y \in C^{\mathcal{I}}.(x, y) \in R^{\mathcal{I}}\}$$

Classical description languages

- The *extension* $(\cdot)^{\mathcal{I}}$ of concepts is defined by:

$$\bot^{\mathcal{I}} = \emptyset \qquad (\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \qquad (C \wedge D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$$

$$(\exists R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \exists y \in C^{\mathcal{I}}.(x, y) \in R^{\mathcal{I}}\}$$

- A concept C is called *satisfiable* if there exists an interpretation $\mathcal{I}$ such that $C^{\mathcal{I}} \neq \emptyset$

Classical description languages

- The *extension* $(\cdot)^{\mathcal{I}}$ of concepts is defined by:

$$\bot^{\mathcal{I}} = \emptyset \qquad (\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \qquad (C \wedge D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$$

$$(\exists R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \exists y \in C^{\mathcal{I}}.(x, y) \in R^{\mathcal{I}}\}$$

- A concept C is called *satisfiable* if there exists an interpretation $\mathcal{I}$ such that $C^{\mathcal{I}} \neq \emptyset$

$\Rightarrow$ Equivalent to modal logic, where concepts correspond to formulas, atomic concepts to atoms, relations to modalities and interpretations to models

TBoxes

- A *general concept inclusion* (gci) is a global axiom of the form $C \sqsubseteq D$

TBoxes

- A *general concept inclusion* (gci) is a global axiom of the form $C \subseteq D$
- An interpretation satisfies $C \subseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$

TBoxes

- A *general concept inclusion* (gci) is a global axiom of the form $C \sqsubseteq D$
- An interpretation satisfies $C \sqsubseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
- A *TBox* is a set of general concept inclusions

TBoxes

- A *general concept inclusion* (gci) is a global axiom of the form $C \sqsubseteq D$
- An interpretation satisfies $C \sqsubseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
- A *TBox* is a set of general concept inclusions
- Reasoning under TBoxes becomes more involved as the complexity jumps from PSpace-completeness to ExpTime-completeness

Fuzzy description languages

- Instead of just $\bot$ and $\top$ we have rational truth values in the unit interval

Fuzzy description languages

- Instead of just $\bot$ and $\top$ we have rational truth values in the unit interval
- Signatures are the same as in the classical case

Fuzzy description languages

- Instead of just $\bot$ and $\top$ we have rational truth values in the unit interval
- Signatures are the same as in the classical case
- Concepts are now generated by:

$$C, D ::= c \mid p \mid \neg C \mid C \sqcap D \mid \exists R.C$$

  where $c \in [0, 1]$, $p \in \mathsf{N_C}$ and $R \in \mathsf{N_R}$

Fuzzy description languages

- Instead of just $\bot$ and $\top$ we have rational truth values in the unit interval
- Signatures are the same as in the classical case
- Concepts are now generated by:

$$C, D ::= c \mid p \mid \neg C \mid C \sqcap D \mid \exists R.C$$

  where $c \in [0,1]$, $p \in \mathsf{N_C}$ and $R \in \mathsf{N_R}$
- Interpretations now have maps $p^{\mathcal{I}} : \Delta^{\mathcal{I}} \to [0,1]$ and $R^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to [0,1]$ for all $p \in \mathsf{N_C}$ and $R \in \mathsf{N_R}$

Fuzzy description languages

- Instead of just $\bot$ and $\top$ we have rational truth values in the unit interval
- Signatures are the same as in the classical case
- Concepts are now generated by:

$$C, D ::= c \mid p \mid \neg C \mid C \sqcap D \mid \exists R.C$$

  where $c \in [0,1]$, $p \in \mathsf{N_C}$ and $R \in \mathsf{N_R}$
- Interpretations now have maps $p^{\mathcal{I}} : \Delta^{\mathcal{I}} \to [0,1]$ and $R^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to [0,1]$ for all $p \in \mathsf{N_C}$ and $R \in \mathsf{N_R}$
- The *valuation* $(\cdot)^{\mathcal{I}} : \Delta^{\mathcal{I}} \to [0,1]$ of concepts is defined by:

Fuzzy description languages

- Instead of just $\bot$ and $\top$ we have rational truth values in the unit interval
- Signatures are the same as in the classical case
- Concepts are now generated by:

$$C, D ::= c \mid p \mid \neg C \mid C \sqcap D \mid \exists R.C$$

  where $c \in [0,1]$, $p \in \mathsf{N_C}$ and $R \in \mathsf{N_R}$
- Interpretations now have maps $p^{\mathcal{I}} : \Delta^{\mathcal{I}} \to [0,1]$ and $R^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to [0,1]$ for all $p \in \mathsf{N_C}$ and $R \in \mathsf{N_R}$
- The *valuation* $(\cdot)^{\mathcal{I}} : \Delta^{\mathcal{I}} \to [0,1]$ of concepts is defined by:

$$c^{\mathcal{I}}(x) = c \qquad (\neg C)^{\mathcal{I}}(x) = 1 - C^{\mathcal{I}}(x)$$

Fuzzy description languages

- Instead of just $\perp$ and $\top$ we have rational truth values in the unit interval
- Signatures are the same as in the classical case
- Concepts are now generated by:

$$C, D ::= c \mid p \mid \neg C \mid C \sqcap D \mid \exists R.C$$

  where $c \in [0,1]$, $p \in \mathsf{N_C}$ and $R \in \mathsf{N_R}$
- Interpretations now have maps $p^{\mathcal{I}} : \Delta^{\mathcal{I}} \to [0,1]$ and $R^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to [0,1]$ for all $p \in \mathsf{N_C}$ and $R \in \mathsf{N_R}$
- The *valuation* $(\cdot)^{\mathcal{I}} : \Delta^{\mathcal{I}} \to [0,1]$ of concepts is defined by:

$$c^{\mathcal{I}}(x) = c \qquad (\neg C)^{\mathcal{I}}(x) = 1 - C^{\mathcal{I}}(x)$$

$$(\exists R.C)^{\mathcal{I}}(x) = \max\{\min\{R^{\mathcal{I}}(x,y), C^{\mathcal{I}}(y)\} \mid y \in \Delta^{\mathcal{I}}\}$$

Fuzzy description languages

- Instead of just $\bot$ and $\top$ we have rational truth values in the unit interval
- Signatures are the same as in the classical case
- Concepts are now generated by:

$$C, D ::= c \mid p \mid \neg C \mid C \sqcap D \mid \exists R.C$$

  where $c \in [0, 1]$, $p \in \mathsf{N_C}$ and $R \in \mathsf{N_R}$
- Interpretations now have maps $p^{\mathcal{I}} : \Delta^{\mathcal{I}} \to [0, 1]$ and $R^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to [0, 1]$ for all $p \in \mathsf{N_C}$ and $R \in \mathsf{N_R}$
- The *valuation* $(\cdot)^{\mathcal{I}} : \Delta^{\mathcal{I}} \to [0, 1]$ of concepts is defined by:

$$c^{\mathcal{I}}(x) = c \qquad (\neg C)^{\mathcal{I}}(x) = 1 - C^{\mathcal{I}}(x)$$

$$(\exists R.C)^{\mathcal{I}}(x) = \max\{\min\{R^{\mathcal{I}}(x, y), C^{\mathcal{I}}(y)\} \mid y \in \Delta^{\mathcal{I}}\}$$

$$(C \sqcap D)^{\mathcal{I}}(x) = ?$$

Fuzzy description languages

- Instead of just $\bot$ and $\top$ we have rational truth values in the unit interval
- Signatures are the same as in the classical case
- Concepts are now generated by:

$$C, D ::= c \mid p \mid \neg C \mid C \sqcap D \mid \exists R.C$$

  where $c \in [0,1]$, $p \in \mathsf{N_C}$ and $R \in \mathsf{N_R}$
- Interpretations now have maps $p^{\mathcal{I}} : \Delta^{\mathcal{I}} \to [0,1]$ and $R^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to [0,1]$ for all $p \in \mathsf{N_C}$ and $R \in \mathsf{N_R}$
- The *valuation* $(\cdot)^{\mathcal{I}} : \Delta^{\mathcal{I}} \to [0,1]$ of concepts is defined by:

$$c^{\mathcal{I}}(x) = c \qquad (\neg C)^{\mathcal{I}}(x) = 1 - C^{\mathcal{I}}(x)$$

$$(\exists R.C)^{\mathcal{I}}(x) = \max\{\min\{R^{\mathcal{I}}(x,y), C^{\mathcal{I}}(y)\} \mid y \in \Delta^{\mathcal{I}}\}$$

$$(C \sqcap D)^{\mathcal{I}}(x) = ?$$

$\Rightarrow$ To define the semantics of $\sqcap$ we need to choose a connective

Fuzzy connectives

- $(C \sqcap D)^{\mathcal{I}}(x) = \min\{C^{\mathcal{I}}(x), D^{\mathcal{I}}(x)\}$ is called the *Zadeh* connective

Fuzzy connectives

- $(C \sqcap D)^{\mathcal{I}}(x) = \min\{C^{\mathcal{I}}(x), D^{\mathcal{I}}(x)\}$ is called the *Zadeh* connective
- Most obvious choice, however little to no change to the classical case

Fuzzy connectives

- $(C \sqcap D)^{\mathcal{I}}(x) = \min\{C^{\mathcal{I}}(x), D^{\mathcal{I}}(x)\}$ is called the *Zadeh* connective
- Most obvious choice, however little to no change to the classical case
- $(C \sqcap D)^{\mathcal{I}}(x) = \max\{0, C^{\mathcal{I}}(x) + D^{\mathcal{I}}(x) - 1\}$ is called the *Łukasiewicz* connective

Fuzzy connectives

- $(C \sqcap D)^{\mathcal{I}}(x) = \min\{C^{\mathcal{I}}(x), D^{\mathcal{I}}(x)\}$ is called the *Zadeh* connective
- Most obvious choice, however little to no change to the classical case
- $(C \sqcap D)^{\mathcal{I}}(x) = \max\{0, C^{\mathcal{I}}(x) + D^{\mathcal{I}}(x) - 1\}$ is called the *Łukasiewicz* connective
- Preferable properties and more expressive, however computationally hard or even undecidable

Fuzzy connectives

- $(C \sqcap D)^{\mathcal{I}}(x) = \min\{C^{\mathcal{I}}(x), D^{\mathcal{I}}(x)\}$ is called the *Zadeh* connective
- Most obvious choice, however little to no change to the classical case
- $(C \sqcap D)^{\mathcal{I}}(x) = \max\{0, C^{\mathcal{I}}(x) + D^{\mathcal{I}}(x) - 1\}$ is called the *Łukasiewicz* connective
- Preferable properties and more expressive, however computationally hard or even undecidable

$\Rightarrow$ We want a middle ground between these extremes

Fuzzy Reasoning

- Satisfiability is now expressed as concept assertions $C \bowtie c$

Fuzzy Reasoning

- Satisfiability is now expressed as concept assertions $C \bowtie c$
- We call $C \bowtie c$ satisfiable if there exists an interpretation $\mathcal{I}$ and individual $x \in \Delta^{\mathcal{I}}$ with $C^{\mathcal{I}}(x) \bowtie c$

Fuzzy Reasoning

- Satisfiability is now expressed as concept assertions $C \bowtie c$
- We call $C \bowtie c$ satisfiable if there exists an interpretation $\mathcal{I}$ and individual $x \in \Delta^{\mathcal{I}}$ with $C^{\mathcal{I}}(x) \bowtie c$
- General concept inclusions are now of the form $C \sqsubseteq D$

Fuzzy Reasoning

- Satisfiability is now expressed as concept assertions $C \bowtie c$
- We call $C \bowtie c$ satisfiable if there exists an interpretation $\mathcal{I}$ and individual $x \in \Delta^{\mathcal{I}}$ with $C^{\mathcal{I}}(x) \bowtie c$
- General concept inclusions are now of the form $C \sqsubseteq D$
- $C \sqsubseteq D$ is satisfied in an interpretation iff $C^{\mathcal{I}}(x) \leq D^{\mathcal{I}}(x)$ for all $x \in \Delta^{\mathcal{I}}$

Fuzzy Reasoning

- Satisfiability is now expressed as concept assertions $C \bowtie c$
- We call $C \bowtie c$ satisfiable if there exists an interpretation $\mathcal{I}$ and individual $x \in \Delta^{\mathcal{I}}$ with $C^{\mathcal{I}}(x) \bowtie c$
- General concept inclusions are now of the form $C \sqsubseteq D$
- $C \sqsubseteq D$ is satisfied in an interpretation iff $C^{\mathcal{I}}(x) \leq D^{\mathcal{I}}(x)$ for all $x \in \Delta^{\mathcal{I}}$
- Reasoning under TBoxes in Zadeh $\mathcal{ALC}$ is ExpTime-complete

Fuzzy Reasoning

- Satisfiability is now expressed as concept assertions $C \bowtie c$
- We call $C \bowtie c$ satisfiable if there exists an interpretation $\mathcal{I}$ and individual $x \in \Delta^{\mathcal{I}}$ with $C^{\mathcal{I}}(x) \bowtie c$
- General concept inclusions are now of the form $C \sqsubseteq D$
- $C \sqsubseteq D$ is satisfied in an interpretation iff $C^{\mathcal{I}}(x) \leq D^{\mathcal{I}}(x)$ for all $x \in \Delta^{\mathcal{I}}$
- Reasoning under TBoxes in Zadeh $\mathcal{ALC}$ is $\textsc{ExpTime}$-complete
- Reasoning under TBoxes in Łukasiewicz $\mathcal{ALC}$ is undecidable

Non-expansive fuzzy $\mathcal{ALC}$

Friedrich-Alexander-Universität
Faculty of Engineering

Idea

- One problem of Łukasiewicz connectives is that a small change in the interpretation can lead to massive changes in behaviour

Idea

- One problem of Łukasiewicz connectives is that a small change in the interpretation can lead to massive changes in behaviour
- This is because there are concepts with an expansive evaluation with regards to logical and behavioural distance

Idea

- One problem of Łukasiewicz connectives is that a small change in the interpretation can lead to massive changes in behaviour
- This is because there are concepts with an expansive evaluation with regards to logical and behavioural distance
- For instance $p \sqcup p = \neg(\neg p \sqcap \neg p)$ would evaluate to $\min\{1, 2p\}$

Idea

- One problem of Łukasiewicz connectives is that a small change in the interpretation can lead to massive changes in behaviour
- This is because there are concepts with an expansive evaluation with regards to logical and behavioural distance
- For instance $p \sqcup p = \neg(\neg p \sqcap \neg p)$ would evaluate to $\min\{1, 2p\}$
- We try to eliminate this problem by restricting to non-expansive operators

Idea

- One problem of Łukasiewicz connectives is that a small change in the interpretation can lead to massive changes in behaviour
- This is because there are concepts with an expansive evaluation with regards to logical and behavioural distance
- For instance $p \sqcup p = \neg(\neg p \sqcap \neg p)$ would evaluate to $\min\{1, 2p\}$
- We try to eliminate this problem by restricting to non-expansive operators
- Namely, we restrict one side of the Łukasiewicz connectives to constants

Idea

T.CS

- One problem of Łukasiewicz connectives is that a small change in the interpretation can lead to massive changes in behaviour
- This is because there are concepts with an expansive evaluation with regards to logical and behavioural distance
- For instance $p \sqcup p = \neg(\neg p \sqcap \neg p)$ would evaluate to $\min\{1, 2p\}$
- We try to eliminate this problem by restricting to non-expansive operators
- Namely, we restrict one side of the Łukasiewicz connectives to constants
- As connective we take the minimum and maximum operators

Definition

T.CS

- The concepts of *non-expansive fuzzy* $\mathcal{ALC}$ are given by

$$C, D ::= p \mid c \mid \neg C \mid C \ominus c \mid C \sqcap D \mid \exists R.C$$

where $p \in \mathsf{N_C}$, $c \in [0,1]$ and $R \in \mathsf{N_R}$

Definition

- The concepts of *non-expansive fuzzy* $\mathcal{ALC}$ are given by

$$C, D ::= p \mid c \mid \neg C \mid C \ominus c \mid C \sqcap D \mid \exists R.C$$

  where $p \in \mathsf{N_C}$, $c \in [0,1]$ and $R \in \mathsf{N_R}$
- The valuation is given by:

$$c^{\mathcal{I}}(x) = c \qquad (\neg C)^{\mathcal{I}}(x) = 1 - C^{\mathcal{I}}(x)$$

$$(\exists R.C)^{\mathcal{I}}(x) = \max\{\min\{R^{\mathcal{I}}(x,y), C^{\mathcal{I}}(y)\} \mid y \in \Delta^{\mathcal{I}}\}$$

$$(C \sqcap D)^{\mathcal{I}}(x) = \min\{C^{\mathcal{I}}(x), D^{\mathcal{I}}(x)\} \qquad (C \ominus c)^{\mathcal{I}}(x) = \max\{0, C^{\mathcal{I}}(x) - c\}$$

Definition

- The concepts of *non-expansive fuzzy* $\mathcal{ALC}$ are given by

$$C, D ::= p \mid c \mid \neg C \mid C \ominus c \mid C \sqcap D \mid \exists R.C$$

  where $p \in \mathsf{N_C}$, $c \in [0,1]$ and $R \in \mathsf{N_R}$
- The valuation is given by:

$$c^{\mathcal{I}}(x) = c \qquad (\neg C)^{\mathcal{I}}(x) = 1 - C^{\mathcal{I}}(x)$$

$$(\exists R.C)^{\mathcal{I}}(x) = \max\{\min\{R^{\mathcal{I}}(x,y), C^{\mathcal{I}}(y)\} \mid y \in \Delta^{\mathcal{I}}\}$$

$$(C \sqcap D)^{\mathcal{I}}(x) = \min\{C^{\mathcal{I}}(x), D^{\mathcal{I}}(x)\} \qquad (C \ominus c)^{\mathcal{I}}(x) = \max\{0, C^{\mathcal{I}}(x) - c\}$$

- Valuations of concepts are now non-expansive

Difference to Zadeh $\mathcal{ALC}$

- Non-expansive fuzzy $\mathcal{ALC}$ is strictly more expressive than Zadeh $\mathcal{ALC}$

Difference to Zadeh $\mathcal{ALC}$

- Non-expansive fuzzy $\mathcal{ALC}$ is strictly more expressive than Zadeh $\mathcal{ALC}$
- For example we can model dampened inheritance of properties:

$$\mathsf{Rich} \sqsubseteq \forall \mathsf{hasChild}.\mathsf{Rich} \oplus 0.1$$

where we define $C \oplus c = \neg(\neg C \ominus c)$

Difference to Zadeh $\mathcal{ALC}$

- Non-expansive fuzzy $\mathcal{ALC}$ is strictly more expressive than Zadeh $\mathcal{ALC}$
- For example we can model dampened inheritance of properties:

$$\mathsf{Rich} \sqsubseteq \forall \mathsf{hasChild}.\mathsf{Rich} \oplus 0.1$$

  where we define $C \oplus c = \neg(\neg C \ominus c)$
- Every non-expansive operation on the unit interval can be approximated by the valuation of a concept of non-expansive fuzzy $\mathcal{ALC}$

Tableau calculi

- *Tableau calculi* are used to decide satisfiability in logical systems

Tableau calculi

- *Tableau calculi* are used to decide satisfiability in logical systems
- The idea is to attempt to build an interpretation for the given formulas by breaking them into simpler components

Tableau calculi

- *Tableau calculi* are used to decide satisfiability in logical systems
- The idea is to attempt to build an interpretation for the given formulas by breaking them into simpler components
- We start with the formulas we are trying to prove or disprove satisfiability of and apply *rules* to break them into simpler formulas

Tableau calculi

- *Tableau calculi* are used to decide satisfiability in logical systems
- The idea is to attempt to build an interpretation for the given formulas by breaking them into simpler components
- We start with the formulas we are trying to prove or disprove satisfiability of and apply *rules* to break them into simpler formulas
- For disjunctions or non-deterministic rules, we create branches to explore all possible cases

Tableau calculi

- *Tableau calculi* are used to decide satisfiability in logical systems
- The idea is to attempt to build an interpretation for the given formulas by breaking them into simpler components
- We start with the formulas we are trying to prove or disprove satisfiability of and apply *rules* to break them into simpler formulas
- For disjunctions or non-deterministic rules, we create branches to explore all possible cases
- We enforce consistency via axiom rules, i.e. we close a branch if we can determine that it is unsatisfiable

Tableau calculi

- *Tableau calculi* are used to decide satisfiability in logical systems
- The idea is to attempt to build an interpretation for the given formulas by breaking them into simpler components
- We start with the formulas we are trying to prove or disprove satisfiability of and apply *rules* to break them into simpler formulas
- For disjunctions or non-deterministic rules, we create branches to explore all possible cases
- We enforce consistency via axiom rules, i.e. we close a branch if we can determine that it is unsatisfiable
- We stop if no rules can be applied to leaves

Tableau calculi

- *Tableau calculi* are used to decide satisfiability in logical systems
- The idea is to attempt to build an interpretation for the given formulas by breaking them into simpler components
- We start with the formulas we are trying to prove or disprove satisfiability of and apply *rules* to break them into simpler formulas
- For disjunctions or non-deterministic rules, we create branches to explore all possible cases
- We enforce consistency via axiom rules, i.e. we close a branch if we can determine that it is unsatisfiable
- We stop if no rules can be applied to leaves
- If we still have an open branch, this branch corresponds to a model that satisfies the formulas

Tableau calculi

- *Tableau calculi* are used to decide satisfiability in logical systems
- The idea is to attempt to build an interpretation for the given formulas by breaking them into simpler components
- We start with the formulas we are trying to prove or disprove satisfiability of and apply *rules* to break them into simpler formulas
- For disjunctions or non-deterministic rules, we create branches to explore all possible cases
- We enforce consistency via axiom rules, i.e. we close a branch if we can determine that it is unsatisfiable
- We stop if no rules can be applied to leaves
- If we still have an open branch, this branch corresponds to a model that satisfies the formulas
- In our case, we need to deal with concept assertions instead of formulas and with existential and universal restrictions, i.e. concept assertions of the form $\exists R.C \triangleright c$ and $\exists R.C \triangleleft c$ respectively

A tableau calculus for non-expansive fuzzy $\mathcal{ALC}$

$$(\mathsf{Ax}\ 1)\ \frac{S, p \triangleright c}{\bot} \quad (\text{if } c \mathrel{\overline{\triangleright}} 1, p \in \mathsf{N_C}) \qquad (\mathsf{Ax}\ 0)\ \frac{S, p \triangleleft c}{\bot} \quad (\text{if } c \mathrel{\overline{\triangleleft}} 0, p \in \mathsf{N_C})$$

A tableau calculus for non-expansive fuzzy $\mathcal{ALC}$

$$(\mathsf{Ax}\ 1)\ \frac{S, p \rhd c}{\bot} \quad (\text{if } c \,\overline{\rhd}\, 1, p \in \mathsf{N_C}) \qquad (\mathsf{Ax}\ 0)\ \frac{S, p \lhd c}{\bot} \quad (\text{if } c \,\overline{\lhd}\, 0, p \in \mathsf{N_C})$$

$$(\mathsf{Ax}\ p)\ \frac{S, p \lhd c, p \rhd d}{\bot} \quad (\text{if } d \lhd_{(\lhd,\rhd)} c, p \in \mathsf{N_C}) \qquad (\mathsf{Ax}\ c)\ \frac{S, c \lhd d}{\bot} \quad (\text{if } c \,\overline{\lhd}^{\circ}\, d)$$

A tableau calculus for non-expansive fuzzy $\mathcal{ALC}$

$$(\text{Ax } 1)\ \frac{S, p \rhd c}{\bot} \quad (\text{if } c \,\overline{\rhd}\, 1, p \in \mathsf{N_C}) \qquad (\text{Ax } 0)\ \frac{S, p \lhd c}{\bot} \quad (\text{if } c \,\overline{\lhd}\, 0, p \in \mathsf{N_C})$$

$$(\text{Ax } p)\ \frac{S, p \lhd c, p \rhd d}{\bot} \quad (\text{if } d \lhd_{(\lhd,\rhd)} c, p \in \mathsf{N_C}) \qquad (\text{Ax } c)\ \frac{S, c \lhd d}{\bot} \quad (\text{if } c \,\overline{\lhd}^{\circ}\, d)$$

$$(\sqcap\rhd)\ \frac{S, C \sqcap D \rhd c}{S, C \rhd c, D \rhd c} \qquad (\sqcap\lhd)\ \frac{S, C \sqcap D \lhd c}{S, C \lhd c \quad S, D \lhd c} \qquad (\neg \bowtie)\ \frac{S, \neg C \bowtie c}{S, C \bowtie^{\circ} 1 - c}$$

A tableau calculus for non-expansive fuzzy $\mathcal{ALC}$

$$(\mathsf{Ax}\ 1)\ \frac{S, p \triangleright c}{\bot} \quad (\text{if } c\,\overline{\triangleright}\,1, p \in \mathsf{N_C}) \qquad (\mathsf{Ax}\ 0)\ \frac{S, p \triangleleft c}{\bot} \quad (\text{if } c\,\overline{\triangleleft}\,0, p \in \mathsf{N_C})$$

$$(\mathsf{Ax}\ p)\ \frac{S, p \triangleleft c, p \triangleright d}{\bot} \quad (\text{if } d \triangleleft_{(\triangleleft,\triangleright)} c, p \in \mathsf{N_C}) \qquad (\mathsf{Ax}\ c)\ \frac{S, c \triangleleft d}{\bot} \quad (\text{if } c\,\overline{\triangleleft}^{\circ}\, d)$$

$$(\sqcap\triangleright)\ \frac{S, C \sqcap D \triangleright c}{S, C \triangleright c, D \triangleright c} \qquad (\sqcap\triangleleft)\ \frac{S, C \sqcap D \triangleleft c}{S, C \triangleleft c \quad S, D \triangleleft c} \qquad (\neg \bowtie)\ \frac{S, \neg C \bowtie c}{S, C \bowtie^{\circ} 1 - c}$$

$$(\ominus\triangleleft)\ \frac{S, C \ominus c \triangleleft d}{S, C \triangleleft d + c, d \triangleleft^{\circ} 0} \qquad (\ominus\triangleright)\ \frac{S, C \ominus c \triangleright d}{S, C \triangleright d + c} \quad (\text{if } d\,\overline{\triangleright}\,0)$$

A tableau calculus for non-expansive fuzzy $\mathcal{ALC}$

$$(\mathsf{Ax}\ 1)\ \frac{S, p \rhd c}{\bot} \quad (\text{if } c \,\overline{\rhd}\, 1, p \in \mathsf{N_C}) \qquad (\mathsf{Ax}\ 0)\ \frac{S, p \lhd c}{\bot} \quad (\text{if } c \,\overline{\lhd}\, 0, p \in \mathsf{N_C})$$

$$(\mathsf{Ax}\ p)\ \frac{S, p \lhd c, p \rhd d}{\bot} \quad (\text{if } d \lhd_{(\lhd,\rhd)} c, p \in \mathsf{N_C}) \qquad (\mathsf{Ax}\ c)\ \frac{S, c \lhd d}{\bot} \quad (\text{if } c \,\overline{\lhd}^{\circ}\, d)$$

$$(\sqcap\rhd)\ \frac{S, C \sqcap D \rhd c}{S, C \rhd c, D \rhd c} \qquad (\sqcap\lhd)\ \frac{S, C \sqcap D \lhd c}{S, C \lhd c \quad S, D \lhd c} \qquad (\neg \bowtie)\ \frac{S, \neg C \bowtie c}{S, C \bowtie^{\circ} 1 - c}$$

$$(\ominus\lhd)\ \frac{S, C \ominus c \lhd d}{S, C \lhd d + c, d \lhd^{\circ} 0} \qquad (\ominus\rhd)\ \frac{S, C \ominus c \rhd d}{S, C \rhd d + c} \quad (\text{if } d \,\overline{\rhd}\, 0)$$

$$(\exists R)\ \frac{S, \{\exists R.D_j \lhd_j d_j \mid 1 \le j \le n\}, \exists R.C \rhd c}{\{D_j \lhd_j d_j \mid d_j \lhd_{(\lhd_j,\rhd)} c, j \in \{1, \dots, n\}\}, C \rhd c}$$

$$(\text{if } c \,\overline{\rhd}\, 0 \text{ and } S \text{ does not contain any } \exists R.D \lhd d)$$

A tableau calculus for non-expansive fuzzy $\mathcal{ALC}$

- We additionally require that the exists rule $(\exists R)$ may only be applied if no other rule is applicable

A tableau calculus for non-expansive fuzzy $\mathcal{ALC}$

- We additionally require that the exists rule $(\exists R)$ may only be applied if no other rule is applicable
- We label the nodes of the tableau as open if they are a leaf and do not have $\bot$ as its label or if the node has a (or all if the applied rule was the exists rule) child node labelled as open

A tableau calculus for non-expansive fuzzy $\mathcal{ALC}$

- We additionally require that the exists rule $(\exists R)$ may only be applied if no other rule is applicable
- We label the nodes of the tableau as open if they are a leaf and do not have $\bot$ as its label or if the node has a (or all if the applied rule was the exists rule) child node labelled as open
- This tableau calculus is sound and complete, i.e. the root node of a tableau can be labelled as open iff the concept assertions in its label are satisfiable

A tableau calculus for non-expansive fuzzy $\mathcal{ALC}$

- We additionally require that the exists rule $(\exists R)$ may only be applied if no other rule is applicable
- We label the nodes of the tableau as open if they are a leaf and do not have $\bot$ as its label or if the node has a (or all if the applied rule was the exists rule) child node labelled as open
- This tableau calculus is sound and complete, i.e. the root node of a tableau can be labelled as open iff the concept assertions in its label are satisfiable
- The algorithm to construct a tableau terminates and is PSPACE-complete

A tableau calculus for non-expansive fuzzy $\mathcal{ALC}$

- We additionally require that the exists rule $(\exists R)$ may only be applied if no other rule is applicable
- We label the nodes of the tableau as open if they are a leaf and do not have $\bot$ as its label or if the node has a (or all if the applied rule was the exists rule) child node labelled as open
- This tableau calculus is sound and complete, i.e. the root node of a tableau can be labelled as open iff the concept assertions in its label are satisfiable
- The algorithm to construct a tableau terminates and is PSpace-complete
- However this tableau algorithm cannot deal with TBoxes

A tableau calculus for non-expansive fuzzy $\mathcal{ALC}$

- We additionally require that the exists rule $(\exists R)$ may only be applied if no other rule is applicable
- We label the nodes of the tableau as open if they are a leaf and do not have $\bot$ as its label or if the node has a (or all if the applied rule was the exists rule) child node labelled as open
- This tableau calculus is sound and complete, i.e. the root node of a tableau can be labelled as open iff the concept assertions in its label are satisfiable
- The algorithm to construct a tableau terminates and is PSPACE-complete
- However this tableau algorithm cannot deal with TBoxes

$\Rightarrow$ We need a suitable characterization of a TBox as a concept assertion to check and stopping conditions to ensure termination

Reducing TBoxes to concept assertions

- Instead of checking $C^{\mathcal{I}}(x) \leq D^{\mathcal{I}}(x)$ for a gci $C \sqsubseteq D$ for every individual, we equivalently check if there exists a constant c such that $C^{\mathcal{I}}(x) \leq c$ and $c \leq D^{\mathcal{I}}(x)$

Reducing TBoxes to concept assertions

- Instead of checking $C^{\mathcal{I}}(x) \leq D^{\mathcal{I}}(x)$ for a gci $C \sqsubseteq D$ for every individual, we equivalently check if there exists a constant c such that $C^{\mathcal{I}}(x) \leq c$ and $c \leq D^{\mathcal{I}}(x)$
- If we want to prove or disprove satisfiability of a set of concept assertions Γ under a TBox $\mathcal{T}$, we can reduce the choices for c to only finitely many values that can be obtained from Γ and $\mathcal{T}$

Reducing TBoxes to concept assertions

- Instead of checking $C^{\mathcal{I}}(x) \leq D^{\mathcal{I}}(x)$ for a gci $C \sqsubseteq D$ for every individual, we equivalently check if there exists a constant c such that $C^{\mathcal{I}}(x) \leq c$ and $c \leq D^{\mathcal{I}}(x)$
- If we want to prove or disprove satisfiability of a set of concept assertions Γ under a TBox $\mathcal{T}$, we can reduce the choices for c to only finitely many values that can be obtained from Γ and $\mathcal{T}$
- More specifically if Z' is the additive group of the constants in Γ and $\mathcal{T}$ intersected with the unit interval, then checking the values of $Z := Z' \cup \{z + \varepsilon \mid z \in Z' \setminus \{1\}\}$ for small enough ε suffices

Reduking TBoxes to concept assertions

- Instead of checking $C^{\mathcal{I}}(x) \leq D^{\mathcal{I}}(x)$ for a gci $C \sqsubseteq D$ for every individual, we equivalently check if there exists a constant c such that $C^{\mathcal{I}}(x) \leq c$ and $c \leq D^{\mathcal{I}}(x)$
- If we want to prove or disprove satisfiability of a set of concept assertions Γ under a TBox $\mathcal{T}$, we can reduce the choices for c to only finitely many values that can be obtained from Γ and $\mathcal{T}$
- More specificially if Z' is the additive group of the constants in Γ and $\mathcal{T}$ intersected with the unit interval, then checking the values of $Z := Z' \cup \{z + \varepsilon \mid z \in Z' \setminus \{1\}\}$ for small enough ε suffices
- The idea is that any interpretation can be transformed into an interpretation with only values of Z that is indistuingishable by concept assertions that can be obtained from Γ and $\mathcal{T}$ from the original interpretation

Reducing TBoxes to concept assertions

- The satisfiability of $\mathcal{T}$ is then equivalent to checking if there is $z \in Z$ such that $C^{\mathcal{I}}(x) \leq z$ and $z \leq D^{\mathcal{I}}(x)$ for every individual x

Reducing TBoxes to concept assertions

- The satisfiability of $\mathcal{T}$ is then equivalent to checking if there is $z \in Z$ such that $C^{\mathcal{I}}(x) \leq z$ and $z \leq D^{\mathcal{I}}(x)$ for every individual x
- We can express this as a concept in the following way:

$$T := \bigsqcap_{C \sqsubseteq D \in \mathcal{T}} \bigsqcup_{z \in Z} (\neg C \oplus z) \sqcap (D \oplus (1 - z))$$

Reducing TBoxes to concept assertions

- The satisfiability of $\mathcal{T}$ is then equivalent to checking if there is $z \in Z$ such that $C^{\mathcal{I}}(x) \leq z$ and $z \leq D^{\mathcal{I}}(x)$ for every individual x
- We can express this as a concept in the following way:

$$T := \bigsqcap_{C \sqsubseteq D \in \mathcal{T}} \bigsqcup_{z \in Z} (\neg C \oplus z) \sqcap (D \oplus (1 - z))$$

- Then any interpretation that satisfies Γ satisfies $\mathcal{T}$ iff $T \geq 1$ is satisfied in every individual

Reducing TBoxes to concept assertions

- The satisfiability of $\mathcal{T}$ is then equivalent to checking if there is $z \in Z$ such that $C^{\mathcal{I}}(x) \leq z$ and $z \leq D^{\mathcal{I}}(x)$ for every individual x
- We can express this as a concept in the following way:

$$T := \bigsqcap_{C \sqsubseteq D \in \mathcal{T}} \bigsqcup_{z \in Z} (\neg C \oplus z) \sqcap (D \oplus (1 - z))$$

- Then any interpretation that satisfies Γ satisfies $\mathcal{T}$ iff $T \geq 1$ is satisfied in every individual

$\Rightarrow$ Adding $T \geq 1$ to the root node and every child node of the exists rule allows us to check this condition

Global caching

- However the algorithm to construct a tableau may not terminate

Global caching

- However the algorithm to construct a tableau may not terminate
- For instance the gci $1 \sqsubseteq \exists R.1$ leads to an infinite path

Global caching

- However the algorithm to construct a tableau may not terminate
- For instance the gci $1 \sqsubseteq \exists R.1$ leads to an infinite path
- To fix this, we require that nodes have unique labels

Global caching

- However the algorithm to construct a tableau may not terminate
- For instance the gci $1 \sqsubseteq \exists R.1$ leads to an infinite path
- To fix this, we require that nodes have unique labels
- More specifically, we cache labels and only create new nodes with labels that are not already cached yet

ExpTime-completeness

- The tableau calculus is complete and sound

ExpTime-completeness

- The tableau calculus is complete and sound
- Completeness is established by constructing an interpretation from an open tableau

ExpTime-completeness

- The tableau calculus is complete and sound
- Completeness is established by constructing an interpretation from an open tableau
- Soundness is established by showing that the rules preserve satsfiability under the TBox

ExpTime-completeness

- The tableau calculus is complete and sound
- Completeness is established by constructing an interpretation from an open tableau
- Soundness is established by showing that the rules preserve satsfiability under the TBox
- The algorithm constructing a tableau and checking if it is open or not is ExpTime

ExpTime-completeness

- The tableau calculus is complete and sound
- Completeness is established by constructing an interpretation from an open tableau
- Soundness is established by showing that the rules preserve satsfiability under the TBox
- The algorithm constructing a tableau and checking if it is open or not is ExpTime
- The ExpTime proof works by finding a suitable bound for the labels

ExpTime-completeness

- The tableau calculus is complete and sound
- Completeness is established by constructing an interpretation from an open tableau
- Soundness is established by showing that the rules preserve satsfiability under the TBox
- The algorithm constructing a tableau and checking if it is open or not is EXPTIME
- The EXPTIME proof works by finding a suitable bound for the labels
- One can prove hardness of the problem by reduction to classical $\mathcal{ALC}$

Conclusion & Future Work

- We provided a complete and sound tableau calculus for satisfiability under a TBox in non-expansive fuzzy $\mathcal{ALC}$

Conclusion & Future Work

- We provided a complete and sound tableau calculus for satisfiability under a TBox in non-expansive fuzzy $\mathcal{ALC}$
- We provided an $\textsc{ExpTime}$ algorithm for deciding this problem and proved $\textsc{ExpTime}$-completeness

Conclusion & Future Work

- We provided a complete and sound tableau calculus for satisfiability under a TBox in non-expansive fuzzy $\mathcal{ALC}$
- We provided an ExpTime algorithm for deciding this problem and proved ExpTime-completeness
- There are still a few notable extensions to non-expansive fuzzy $\mathcal{ALC}$ which warrant attention, namely transitive roles, role inclusions and nominals

Questions?

Friedrich-Alexander-Universität
Faculty of Engineering