



TECHNISCHE
UNIVERSITÄT
DRESDEN

FOUNDATIONS OF SEMANTIC WEB TECHNOLOGIES

Tableau Procedures II

Sebastian Rudolph

Dresden, 17 May 2013



DRESDEN
UNIVERSITÄT
Fakultät für
Informatik
und Physik

Content

Overview & XML	9 APR DS2	Hypertableau II	7 JUN DS5
Introduction into RDF	9 APR DS3	Tutorial 5	11 JUN DS2
RDFS – Syntax & Intuition	12 APR DS5	SPARQL Syntax & Intuition	11 JUN DS3
RDFS – Semantics	16 APR DS2	SPARQL – Semantics	14 JUN DS2
RDFS Rule-based Reasoning	16 APR DS3	SPARQL 1.1	14 JUN DS3
Tutorial 1	19 APR DS5	SPARQL Entailment	14 JUN DS5
OWL – Syntax & Intuition	26 APR DS5	Tutorial 6	18 JUN DS2
Tutorial 2	3 MAY DS5	SPARQL Implementation	18 JUN DS3
OWL – Syntax & Intuition ctd.	7 MAY DS2	Ontology Editing	2 JUL DS2
OWL & Description Logics	7 MAY DS3	Ontology Engineering	2 JUL DS3
OWL 2	10 MAY DS5	Tutorial 7	9 JUL DS2
Tutorial 3	14 MAY DS2	Linked Data	9 JUL DS3
Tableau I	14 MAY DS3	Applications	12 JUL DS5
Tableau II	17 MAY DS5	Test Exam	16 JUL DS2
Tutorial 4	7 JUN DS2	Test Exam Evaluation	16 JUL DS3
Hypertableau I	7 JUN DS3	Q&A Session	19 JUL DS5

Tableau Calculus 2

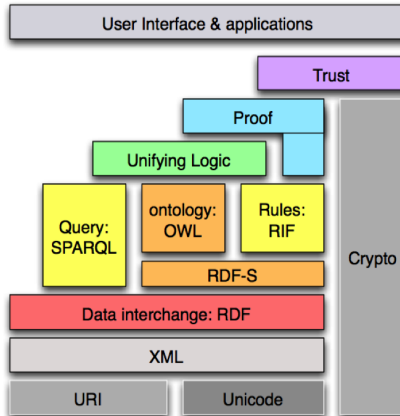
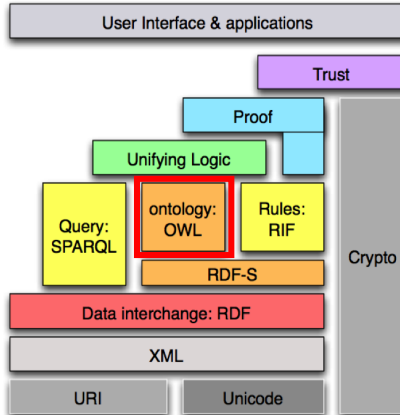


Tableau Calculus 2



Agenda

- Recap Tableau Calculus
- Tableau with \mathcal{ALC} TBoxes
- Tableau for \mathcal{ALC} Knowledge Bases
- Extension by Inverse Roles
- Extension by Functional Roles
- Model Construction with Unravelling
- Summary

Agenda

- [Recap Tableau Calculus](#)
- Tableau with \mathcal{ALC} TBoxes
- Tableau for \mathcal{ALC} Knowledge Bases
- Extension by Inverse Roles
- Extension by Functional Roles
- Model Construction with Unravelling
- Summary

Tableau Algorithm for \mathcal{ALC} Concepts and TBoxes

- check of satisfiability of C by construction of an abstraction of a model \mathcal{I} such that $C^{\mathcal{I}} \neq \emptyset$

Tableau Algorithm for \mathcal{ALC} Concepts and TBoxes

- check of satisfiability of C by construction of an abstraction of a model \mathcal{I} such that $C^{\mathcal{I}} \neq \emptyset$
- concepts in negation normal form (NNF) \rightsquigarrow easier rules

Tableau Algorithm for \mathcal{ALC} Concepts and TBoxes

- check of satisfiability of C by construction of an abstraction of a model \mathcal{I} such that $C^{\mathcal{I}} \neq \emptyset$
- concepts in negation normal form (NNF) \rightsquigarrow easier rules
- tableau (model abstraction) corresponds to a graph/tree $G = \langle V, E, L \rangle$

Tableau Algorithm for \mathcal{ALC} Concepts and TBoxes

- check of satisfiability of C by construction of an abstraction of a model \mathcal{I} such that $C^{\mathcal{I}} \neq \emptyset$
- concepts in negation normal form (NNF) \rightsquigarrow easier rules
- tableau (model abstraction) corresponds to a graph/tree $G = \langle V, E, L \rangle$
- initialize G with a node v such that $L(v) = \{C\}$

Tableau Algorithm for \mathcal{ALC} Concepts and TBoxes

- check of satisfiability of C by construction of an abstraction of a model \mathcal{I} such that $C^{\mathcal{I}} \neq \emptyset$
- concepts in negation normal form (NNF) \rightsquigarrow easier rules
- tableau (model abstraction) corresponds to a graph/tree $G = \langle V, E, L \rangle$
- initialize G with a node v such that $L(v) = \{C\}$
- extend G by applying **tableau rules**

Tableau Algorithm for \mathcal{ALC} Concepts and TBoxes

- check of satisfiability of C by construction of an abstraction of a model \mathcal{I} such that $C^{\mathcal{I}} \neq \emptyset$
- concepts in negation normal form (NNF) \rightsquigarrow easier rules
- tableau (model abstraction) corresponds to a graph/tree $G = \langle V, E, L \rangle$
- initialize G with a node v such that $L(v) = \{C\}$
- extend G by applying **tableau rules**
 - \sqcup rule is non-deterministic (we guess)
- tableau branch closed if G contains an atomic contradiction (aka **clash**)

Tableau Algorithm for \mathcal{ALC} Concepts and TBoxes

- check of satisfiability of C by construction of an abstraction of a model \mathcal{I} such that $C^{\mathcal{I}} \neq \emptyset$
- concepts in negation normal form (NNF) \rightsquigarrow easier rules
- tableau (model abstraction) corresponds to a graph/tree $G = \langle V, E, L \rangle$
- initialize G with a node v such that $L(v) = \{C\}$
- extend G by applying **tableau rules**
 - \sqcup rule is non-deterministic (we guess)
- tableau branch closed if G contains an atomic contradiction (aka **clash**)
- tableau construction successful if no rules applicable and no contradiction

Tableau Algorithm for \mathcal{ALC} Concepts and TBoxes

- check of satisfiability of C by construction of an abstraction of a model \mathcal{I} such that $C^{\mathcal{I}} \neq \emptyset$
- concepts in negation normal form (NNF) \rightsquigarrow easier rules
- tableau (model abstraction) corresponds to a graph/tree $G = \langle V, E, L \rangle$
- initialize G with a node v such that $L(v) = \{C\}$
- extend G by applying **tableau rules**
 - \sqcup rule is non-deterministic (we guess)
- tableau branch closed if G contains an atomic contradiction (aka **clash**)
- tableau construction successful if no rules applicable and no contradiction
- C is satisfiable iff there is a successful tableau construction

Tableau Rules for \mathcal{ALC} Concepts

- \sqcap -rule: For an $v \in V$ with $C \sqcap D \in L(v)$ and $\{C, D\} \not\subseteq L(v)$, let $L(v) := L(v) \cup \{C, D\}$.
- \sqcup -rule: For an $v \in V$ with $C \sqcup D \in L(v)$ and $\{C, D\} \cap L(v) = \emptyset$, choose $X \in \{C, D\}$ and let $L(v) := L(v) \cup \{X\}$.
- \exists -rule: For an $v \in V$ with $\exists r.C \in L(v)$ such that there is no r -successor v' of v with $C \in L(v')$, let $V = V \cup \{v'\}$, $E = E \cup \{(v, v')\}$, $L(v') := \{C\}$ and $L(v, v') := \{r\}$ for v' a new node.
- \forall -rule: For $v, v' \in V$, v' r -neighbor of v , $\forall r.C \in L(v)$ and $C \notin L(v')$, let $L(v') := L(v') \cup \{C\}$.

Tableau Algorithm Example

$$C = \exists r.(A \sqcup \exists r.B) \sqcap \exists r.\neg A \sqcap \forall r.(\neg A \sqcap \forall r.(\neg B \sqcup A))$$

u

$$L(u) = \{C\}$$

Tableau Algorithm Example

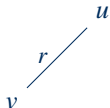
$$C = \exists r.(A \sqcup \exists r.B) \sqcap \exists r.\neg A \sqcap \forall r.(\neg A \sqcap \forall r.(\neg B \sqcup A))$$

u

$$L(u) = \{C, \exists r.(A \sqcup \exists r.B), \\ \exists r.\neg A, \forall r.(\neg A \sqcap \forall r.(\neg B \sqcup A))\}$$

Tableau Algorithm Example

$$C = \exists r.(A \sqcup \exists r.B) \sqcap \exists r.\neg A \sqcap \forall r.(\neg A \sqcap \forall r.(\neg B \sqcup A))$$

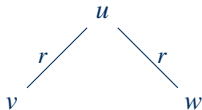


$$L(u) = \{C, \exists r.(A \sqcup \exists r.B), \\ \exists r.\neg A, \forall r.(\neg A \sqcap \forall r.(\neg B \sqcup A))\}$$

$$L(v) = \{A \sqcup \exists r.B\}$$

Tableau Algorithm Example

$$C = \exists r.(A \sqcup \exists r.B) \sqcap \exists r.\neg A \sqcap \forall r.(\neg A \sqcap \forall r.(\neg B \sqcup A))$$



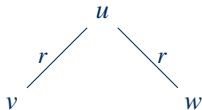
$$L(u) = \{C, \exists r.(A \sqcup \exists r.B), \\ \exists r.\neg A, \forall r.(\neg A \sqcap \forall r.(\neg B \sqcup A))\}$$

$$L(v) = \{A \sqcup \exists r.B\}$$

$$L(w) = \{\neg A\}$$

Tableau Algorithm Example

$$C = \exists r.(A \sqcup \exists r.B) \sqcap \exists r.\neg A \sqcap \forall r.(\neg A \sqcap \forall r.(\neg B \sqcup A))$$



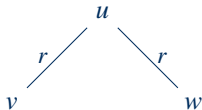
$$L(u) = \{C, \exists r.(A \sqcup \exists r.B), \\ \exists r.\neg A, \forall r.(\neg A \sqcap \forall r.(\neg B \sqcup A))\}$$

$$L(v) = \{A \sqcup \exists r.B, \neg A, \forall r.(\neg B \sqcup A)\}$$

$$L(w) = \{\neg A\}$$

Tableau Algorithm Example

$$C = \exists r.(A \sqcup \exists r.B) \sqcap \exists r.\neg A \sqcap \forall r.(\neg A \sqcap \forall r.(\neg B \sqcup A))$$



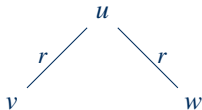
$$L(u) = \{C, \exists r.(A \sqcup \exists r.B), \\ \exists r.\neg A, \forall r.(\neg A \sqcap \forall r.(\neg B \sqcup A))\}$$

$$L(v) = \{A \sqcup \exists r.B, \neg A, \forall r.(\neg B \sqcup A)\}$$

$$L(w) = \{\neg A, \forall r.(\neg B \sqcup A)\}$$

Tableau Algorithm Example

$$C = \exists r.(A \sqcup \exists r.B) \sqcap \exists r.\neg A \sqcap \forall r.(\neg A \sqcap \forall r.(\neg B \sqcup A))$$



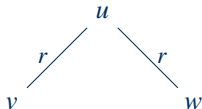
$$L(u) = \{C, \exists r.(A \sqcup \exists r.B), \\ \exists r.\neg A, \forall r.(\neg A \sqcap \forall r.(\neg B \sqcup A))\}$$

$$L(v) = \{A \sqcup \exists r.B, \neg A, \forall r.(\neg B \sqcup A), A\}$$

$$L(w) = \{\neg A, \forall r.(\neg B \sqcup A)\}$$

Tableau Algorithm Example

$$C = \exists r.(A \sqcup \exists r.B) \sqcap \exists r.\neg A \sqcap \forall r.(\neg A \sqcap \forall r.(\neg B \sqcup A))$$



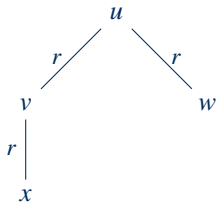
$$L(u) = \{C, \exists r.(A \sqcup \exists r.B), \\ \exists r.\neg A, \forall r.(\neg A \sqcap \forall r.(\neg B \sqcup A))\}$$

$$L(v) = \{A \sqcup \exists r.B, \neg A, \forall r.(\neg B \sqcup A), \text{X}\}$$

$$L(w) = \{\neg A, \forall r.(\neg B \sqcup A)\}$$

Tableau Algorithm Example

$$C = \exists r.(A \sqcup \exists r.B) \sqcap \exists r.\neg A \sqcap \forall r.(\neg A \sqcap \forall r.(\neg B \sqcup A))$$



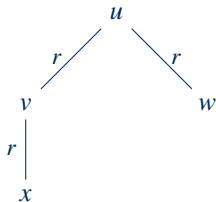
$$L(u) = \{C, \exists r.(A \sqcup \exists r.B), \\ \exists r.\neg A, \forall r.(\neg A \sqcap \forall r.(\neg B \sqcup A))\}$$

$$L(v) = \{A \sqcup \exists r.B, \neg A, \forall r.(\neg B \sqcup A), \cancel{\exists r.B}\}$$

$$L(w) = \{\neg A, \forall r.(\neg B \sqcup A)\}$$

Tableau Algorithm Example

$$C = \exists r.(A \sqcup \exists r.B) \sqcap \exists r.\neg A \sqcap \forall r.(\neg A \sqcap \forall r.(\neg B \sqcup A))$$



$$L(u) = \{C, \exists r.(A \sqcup \exists r.B), \\ \exists r.\neg A, \forall r.(\neg A \sqcap \forall r.(\neg B \sqcup A))\}$$

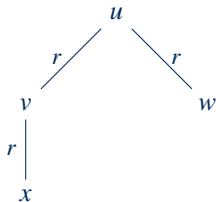
$$L(v) = \{A \sqcup \exists r.B, \neg A, \forall r.(\neg B \sqcup A), \cancel{\exists r.B}\}$$

$$L(w) = \{\neg A, \forall r.(\neg B \sqcup A)\}$$

$$L(x) = \{B\}$$

Tableau Algorithm Example

$$C = \exists r.(A \sqcup \exists r.B) \sqcap \exists r.\neg A \sqcap \forall r.(\neg A \sqcap \forall r.(\neg B \sqcup A))$$



$$L(u) = \{C, \exists r.(A \sqcup \exists r.B), \\ \exists r.\neg A, \forall r.(\neg A \sqcap \forall r.(\neg B \sqcup A))\}$$

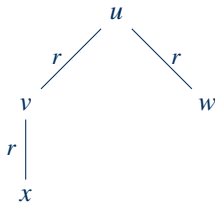
$$L(v) = \{A \sqcup \exists r.B, \neg A, \forall r.(\neg B \sqcup A), \cancel{\exists r.B}\}$$

$$L(w) = \{\neg A, \forall r.(\neg B \sqcup A)\}$$

$$L(x) = \{B, \neg B \sqcup A\}$$

Tableau Algorithm Example

$$C = \exists r.(A \sqcup \exists r.B) \sqcap \exists r.\neg A \sqcap \forall r.(\neg A \sqcap \forall r.(\neg B \sqcup A))$$



$$L(u) = \{C, \exists r.(A \sqcup \exists r.B), \\ \exists r.\neg A, \forall r.(\neg A \sqcap \forall r.(\neg B \sqcup A))\}$$

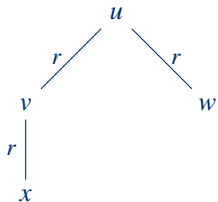
$$L(v) = \{A \sqcup \exists r.B, \neg A, \forall r.(\neg B \sqcup A), \cancel{\exists r.B}\}$$

$$L(w) = \{\neg A, \forall r.(\neg B \sqcup A)\}$$

$$L(x) = \{B, \neg B \sqcup A, \neg B\}$$

Tableau Algorithm Example

$$C = \exists r.(A \sqcup \exists r.B) \sqcap \exists r.\neg A \sqcap \forall r.(\neg A \sqcap \forall r.(\neg B \sqcup A))$$



$$L(u) = \{C, \exists r.(A \sqcup \exists r.B), \\ \exists r.\neg A, \forall r.(\neg A \sqcap \forall r.(\neg B \sqcup A))\}$$

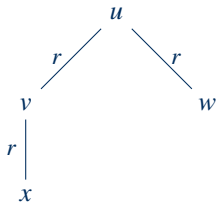
$$L(v) = \{A \sqcup \exists r.B, \neg A, \forall r.(\neg B \sqcup A), \exists r.B\}$$

$$L(w) = \{\neg A, \forall r.(\neg B \sqcup A)\}$$

$$L(x) = \{B, \neg B \sqcup A, B\}$$

Tableau Algorithm Example

$$C = \exists r.(A \sqcup \exists r.B) \sqcap \exists r.\neg A \sqcap \forall r.(\neg A \sqcap \forall r.(\neg B \sqcup A))$$



$$L(u) = \{C, \exists r.(A \sqcup \exists r.B), \\ \exists r.\neg A, \forall r.(\neg A \sqcap \forall r.(\neg B \sqcup A))\}$$

$$L(v) = \{A \sqcup \exists r.B, \neg A, \forall r.(\neg B \sqcup A), \exists r.B\}$$

$$L(w) = \{\neg A, \forall r.(\neg B \sqcup A)\}$$

$$L(x) = \{B, \neg B \sqcup A, B, A\}$$

Tableau Algorithm Example

the model \mathcal{I} constructed by the algorithm is the following:

$$\Delta^{\mathcal{I}} = \{u, v, w, x\}$$

$$A^{\mathcal{I}} = \{x\}$$

$$B^{\mathcal{I}} = \{x\}$$

$$r^{\mathcal{I}} = \{\langle u, v \rangle, \langle u, w \rangle, \langle v, x \rangle\}$$

Check that indeed $C^{\mathcal{I}} = \{u\}$, given the defined semantics of \mathcal{ALC}

Agenda

- Recap Tableau Calculus
- Tableau with \mathcal{ALC} TBoxes
- Tableau for \mathcal{ALC} Knowledge Bases
- Extension by Inverse Roles
- Extension by Functional Roles
- Model Construction with Unravelling
- Summary

Tableau Algorithm for TBoxes

We extend the tableau algorithm to capture \mathcal{ALC} TBoxes

- a TBox contains axioms (GCIs) of the form $C \sqsubseteq D$
- assumption: occurrences of $C \equiv D$ have been replaced by $C \sqsubseteq D$ and $D \sqsubseteq C$
- every GCI is equivalent to $\top \sqsubseteq \neg C \sqcup D$

We can compress the whole TBox into one axiom (we say we “internalize” it):

$$\mathcal{T} = \{C_i \sqsubseteq D_i \mid 1 \leq i \leq n\}$$

is equivalent to:

$$\mathcal{T}' = \{\top \sqsubseteq \prod_{1 \leq i \leq n} \neg C_i \sqcup D_i\}$$

Let $C_{\mathcal{T}}$ be the concept on the rhs of the GCI in NNF.

Tableau Algorithm for TBoxes

We extend the rules of the \mathcal{ALC} tableau algorithm with the rule:

\mathcal{T} rule: For an arbitrary $v \in V$ with $C_{\mathcal{T}} \notin L(v)$,
let $L(v) := L(v) \cup \{C_{\mathcal{T}}\}$.

Example: Let $\mathcal{T} = A \sqsubseteq \exists r.A$. Is A satisfiable given \mathcal{T} ?

Tableau Algorithm for TBoxes

We extend the rules of the \mathcal{ALC} tableau algorithm with the rule:

\mathcal{T} rule: For an arbitrary $v \in V$ with $C_{\mathcal{T}} \notin L(v)$,
let $L(v) := L(v) \cup \{C_{\mathcal{T}}\}$.

Example: Let $\mathcal{T} = A \sqsubseteq \exists r.A$. Is A satisfiable given \mathcal{T} ?

the tableau algorithm doesn't terminate any more!

Tableau Algorithm for TBoxes

We extend the rules of the \mathcal{ALC} tableau algorithm with the rule:

\mathcal{T} rule: For an arbitrary $v \in V$ with $C_{\mathcal{T}} \notin L(v)$,
let $L(v) := L(v) \cup \{C_{\mathcal{T}}\}$.

Example: Let $\mathcal{T} = A \sqsubseteq \exists r.A$. Is A satisfiable given \mathcal{T} ?

the tableau algorithm doesn't terminate any more!

the quantifier depth does not necessarily decrease for newly introduced child nodes

Tableau Algorithm for TBoxes

We extend the rules of the \mathcal{ALC} tableau algorithm with the rule:

\mathcal{T} rule: For an arbitrary $v \in V$ with $C_{\mathcal{T}} \notin L(v)$,
let $L(v) := L(v) \cup \{C_{\mathcal{T}}\}$.

Example: Let $\mathcal{T} = A \sqsubseteq \exists r.A$. Is A satisfiable given \mathcal{T} ?

the tableau algorithm doesn't terminate any more!

the quantifier depth does not necessarily decrease for newly introduced child nodes

solution: we will recognize cycles (that is, repeating node labellings)

Tableau Algorithm for TBoxes

Definition (Blocking)

A node $v \in V$ **blocks** a node $v' \in V$ **directly**, if:

- 1 v' is reachable from v ,
- 2 $L(v') \subseteq L(v)$; and
- 3 there is no directly blocking node v'' such that v' is reachable from v'' .

A node $v' \in V$ is **blocked** if either

- 1 v' is blocked directly or
- 2 there is a directly blocked node v , such that v' is reachable from v .

Tableau Algorithm for TBoxes

Definition (Blocking)

A node $v \in V$ **blocks** a node $v' \in V$ **directly**, if:

- 1 v' is reachable from v ,
- 2 $L(v') \subseteq L(v)$; and
- 3 there is no directly blocking node v'' such that v' is reachable from v'' .

A node $v' \in V$ is **blocked** if either

- 1 v' is blocked directly or
- 2 there is a directly blocked node v , such that v' is reachable from v .

The application of the \exists rule is restricted to nodes that are **not blocked**.

Tableau Algorithm with Blocking

Example: Let $\mathcal{T} = A \sqsubseteq \exists r.A$. Is A satisfiable w.r.t. \mathcal{T} ?

we obtain the following contradiction-free tableau:



$$L(v_0) = \{A, C_{\mathcal{T}}, \exists r.A\}$$

$$L(v_1) = \{A, C_{\mathcal{T}}, \exists r.A\}$$

wherein v_1 is **directly** blocked by v_0

Tableau Algorithm with Blocking

Example: Let $\mathcal{T} = A \sqsubseteq \exists r.A$. Is A satisfiable w.r.t. \mathcal{T} ?

we obtain the following contradiction-free tableau:



$$L(v_0) = \{A, C_{\mathcal{T}}, \exists r.A\}$$

$$L(v_1) = \{A, C_{\mathcal{T}}, \exists r.A\}$$

wherein v_1 is **directly** blocked by v_0

again, the algorithm constructs finite trees

- from a contradiction-free tableau, we can construct a model
- if there is no contradiction-free tableau, there is no model

From the Tableau to the Model

again, we can construct a finite model from a contradiction-free tableau:

$$\Delta^{\mathcal{I}} = \{v_0\}$$

$$A^{\mathcal{I}} = \Delta^{\mathcal{I}}$$

$$r^{\mathcal{I}} = \{\langle v_0, v_0 \rangle\}$$

- blocked nodes do not represent elements of the model
- when constructing the model, an edge from a node v to a directly blocked node v' will be “translated” into an “edge” from v to the node, that directly blocks v'

From the Tableau to the Model

again, we can construct a finite model from a contradiction-free tableau:

$$\Delta^{\mathcal{I}} = \{v_0\}$$

$$A^{\mathcal{I}} = \Delta^{\mathcal{I}}$$

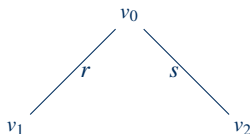
$$r^{\mathcal{I}} = \{\langle v_0, v_0 \rangle\}$$

- blocked nodes do not represent elements of the model
 - when constructing the model, an edge from a node v to a directly blocked node v' will be “translated” into an “edge” from v to the node, that directly blocks v'
- ↪ we have the **finite model property**
- ↪ constructed model is not necessarily tree-shaped

Tableau Algorithm with Blocking II

Example: Let $\mathcal{T} = A \sqsubseteq \exists r.A \sqcap \exists s.B$. Is A satisfiable w.r.t. \mathcal{T} ?

We obtain the following contradiction-free tableau:



$$L(v_0) = \{A, C_{\mathcal{T}}, \exists r.A \sqcap \exists s.B, \exists r.A, \exists s.B\}$$

$$L(v_1) = \{A, C_{\mathcal{T}}, \exists r.A \sqcap \exists s.B, \exists r.A, \exists s.B\}$$

$$L(v_2) = \{B, C_{\mathcal{T}}, \neg A\}$$

in which v_1 is again **directly** blocked by v_0

From the Tableau to a Model II

again, we can construct a finite model from a contradiction-free tableau:

$$\Delta^{\mathcal{I}} = \{v_0, v_2\}$$

$$A^{\mathcal{I}} = \{v_0\}$$

$$B^{\mathcal{I}} = \{v_2\}$$

$$r^{\mathcal{I}} = \{\langle v_0, v_0 \rangle\}$$

$$s^{\mathcal{I}} = \{\langle v_0, v_2 \rangle\}$$

Tableau Algorithm Example

Example: Let $\mathcal{T} = \{A \sqsubseteq B \sqcap \exists r.C, B \equiv C \sqcup D, C \sqsubseteq \exists r.D\}$. Is A satisfiable w.r.t. \mathcal{T} ?

Tableau Algorithm Example

Example: Let $\mathcal{T} = \{A \sqsubseteq B \sqcap \exists r.C, B \equiv C \sqcup D, C \sqsubseteq \exists r.D\}$. Is A satisfiable w.r.t. \mathcal{T} ?

Normalization I:

$\mathcal{T}' = \{A \sqsubseteq B, A \sqsubseteq \exists r.C, B \sqsubseteq C \sqcup D, C \sqcup D \sqsubseteq B, C \sqsubseteq \exists r.D\}$

Tableau Algorithm Example

Example: Let $\mathcal{T} = \{A \sqsubseteq B \sqcap \exists r.C, B \equiv C \sqcup D, C \sqsubseteq \exists r.D\}$. Is A satisfiable w.r.t. \mathcal{T} ?

Normalization I:

$$\mathcal{T}' = \{A \sqsubseteq B, A \sqsubseteq \exists r.C, B \sqsubseteq C \sqcup D, C \sqcup D \sqsubseteq B, C \sqsubseteq \exists r.D\}$$

Normalization II:

$$\mathcal{T}' = \{A \sqsubseteq B, A \sqsubseteq \exists r.C, B \sqsubseteq C \sqcup D, C \sqsubseteq B, D \sqsubseteq B, C \sqsubseteq \exists r.D\}$$

Tableau Algorithm Example

Example: Let $\mathcal{T} = \{A \sqsubseteq B \sqcap \exists r.C, B \equiv C \sqcup D, C \sqsubseteq \exists r.D\}$. Is A satisfiable w.r.t. \mathcal{T} ?

Normalization I:

$$\mathcal{T}' = \{A \sqsubseteq B, A \sqsubseteq \exists r.C, B \sqsubseteq C \sqcup D, C \sqcup D \sqsubseteq B, C \sqsubseteq \exists r.D\}$$

Normalization II:

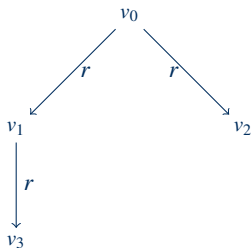
$$\mathcal{T}' = \{A \sqsubseteq B, A \sqsubseteq \exists r.C, B \sqsubseteq C \sqcup D, C \sqsubseteq B, D \sqsubseteq B, C \sqsubseteq \exists r.D\}$$

$$C_{\mathcal{T}} = (\neg A \sqcup B) \sqcap (\neg A \sqcup \exists r.C) \sqcap (\neg B \sqcup C \sqcup D) \sqcap (\neg C \sqcup B) \sqcap (\neg D \sqcup B) \sqcap (\neg C \sqcup \exists r.D)$$

Tableau Algorithm Example

$$C_{\mathcal{T}} = (\neg A \sqcup B) \sqcap (\neg A \sqcup \exists r.C) \sqcap (\neg B \sqcup C \sqcup D) \sqcap (\neg C \sqcup B) \sqcap (\neg D \sqcup B) \sqcap (\neg C \sqcup \exists r.D)$$

we obtain the following contradiction-free tableau:



$$L(v_0) = \{A, C_{\mathcal{T}}, \dots, B, \exists r.C, C, \neg D, \exists r.D\}$$

$$L(v_1) = \{C, C_{\mathcal{T}}, \dots, \neg A, B, \exists r.D\}$$

$$L(v_2) = \{D, C_{\mathcal{T}}, \dots, \neg A, \neg C, B\}$$

$$L(v_3) = \{D, C_{\mathcal{T}}, \dots, \neg A, \neg C, B\}$$

Agenda

- Recap Tableau Calculus
- Tableau with \mathcal{ALC} TBoxes
- Tableau for \mathcal{ALC} Knowledge Bases
- Extension by Inverse Roles
- Extension by Functional Roles
- Model Construction with Unravelling
- Summary

Treatment of ABoxes

to take an ABox \mathcal{A} into account, initialize G such that

- V contains a node v_a for each individual a occurring in \mathcal{A}

Treatment of ABoxes

to take an ABox \mathcal{A} into account, initialize G such that

- V contains a node v_a for each individual a occurring in \mathcal{A}
- $L(v_a) = \{C \mid C(a) \in \mathcal{A}\}$

Treatment of ABoxes

to take an ABox \mathcal{A} into account, initialize G such that

- V contains a node v_a for each individual a occurring in \mathcal{A}
- $L(v_a) = \{C \mid C(a) \in \mathcal{A}\}$
- $\langle v_a, v_b \rangle \in E$ and $r \in L(\langle v_a, v_b \rangle)$ iff $r(a, b) \in \mathcal{A}$

Treatment of ABoxes

to take an ABox \mathcal{A} into account, initialize G such that

- V contains a node v_a for each individual a occurring in \mathcal{A}
- $L(v_a) = \{C \mid C(a) \in \mathcal{A}\}$
- $\langle v_a, v_b \rangle \in E$ and $r \in L(\langle v_a, v_b \rangle)$ iff $r(a, b) \in \mathcal{A}$

the tableau rules can then be applied to this initialized graph

Agenda

- Recap Tableau Calculus
- Tableau with \mathcal{ALC} TBoxes
- Tableau for \mathcal{ALC} Knowledge Bases
- Extension by Inverse Roles
- Extension by Functional Roles
- Model Construction with Unravelling
- Summary

Tableau for \mathcal{ALC} with Inverse Roles

in order to take into account inverse roles, we have to make the following changes

- 1 edge labels may contain inverse roles (r^-),

Tableau for \mathcal{ALC} with Inverse Roles

in order to take into account inverse roles, we have to make the following changes

- 1 edge labels may contain inverse roles (r^-),
- 2 a node v' is an r -neighbor of a node v if either
 - v' is an r -successor of v or
 - v is an r^- -successor of v'

Tableau for \mathcal{ALC} with Inverse Roles

in order to take into account inverse roles, we have to make the following changes

- 1 edge labels may contain inverse roles (r^-),
- 2 a node v' is an r -neighbor of a node v if either
 - v' is an r -successor of v or
 - v is an r^- -successor of v'
- 3 replace the term “ r -successor” in the \forall - and the \exists -rule with “ r -neighbor”

the \exists -rule still generates

- an r -**successor** for a concept $\exists r.C$ (if no fitting neighbor exists yet)
- an r^- -**successor** for a concept $\exists r^-.C$ (if no fitting neighbor exists yet)

Tableau Example with Inverses

Example: is A satisfiable w.r.t. \mathcal{T} ?

$$\mathcal{T} = \{A \equiv \exists r^{-}.A \sqcap (\forall r.(\neg A \sqcup \exists s.B))\}$$

Tableau Example with Inverses

Example: is A satisfiable w.r.t. \mathcal{T} ?

$$\mathcal{T} = \{A \equiv \exists r^- . A \sqcap (\forall r. (\neg A \sqcup \exists s. B))\}$$

$$\begin{aligned} C_{\mathcal{T}} = & (\neg A \sqcup \exists r^- . A) \sqcap (\neg A \sqcup \forall r. (\neg A \sqcup \exists s. B)) \sqcap \\ & (\forall r^- . (\neg A) \sqcup \exists r. (A \sqcap \forall s. (\neg B)) \sqcup A) \end{aligned}$$

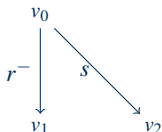
Tableau Example with Inverses

Example: is A satisfiable w.r.t. \mathcal{T} ?

$$\mathcal{T} = \{A \equiv \exists r^- .A \sqcap (\forall r.(\neg A \sqcup \exists s.B))\}$$

$$C_{\mathcal{T}} = (\neg A \sqcup \exists r^- .A) \sqcap (\neg A \sqcup \forall r.(\neg A \sqcup \exists s.B)) \sqcap$$

$$(\forall r^- .(\neg A) \sqcup \exists r.(A \sqcap \forall s.(\neg B)) \sqcup A)$$



$$L(v_0) = \{A, C_{\mathcal{T}}, \exists r^- .A, \forall r.(\neg A \sqcup \exists s.B),$$

$$\neg A \sqcup \exists s.B, \exists s.B\}$$

$$L(v_1) = \{A, C_{\mathcal{T}}, \exists r^- .A, \forall r.(\neg A \sqcup \exists s.B)\}$$

$$L(v_2) = \{B, C_{\mathcal{T}}, \neg A, \forall r^- .(\neg A)\}$$

v_0 blocks v_1

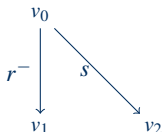
Tableau Example with Inverses

Example: is A satisfiable w.r.t. \mathcal{T} ?

$$\mathcal{T} = \{A \equiv \exists r^- .A \sqcap (\forall r.(\neg A \sqcup \exists s.B))\}$$

$$C_{\mathcal{T}} = (\neg A \sqcup \exists r^- .A) \sqcap (\neg A \sqcup \forall r.(\neg A \sqcup \exists s.B)) \sqcap$$

$$(\forall r^- .(\neg A) \sqcup \exists r.(A \sqcap \forall s.(\neg B)) \sqcup A)$$



$$L(v_0) = \{A, C_{\mathcal{T}}, \exists r^- .A, \forall r.(\neg A \sqcup \exists s.B),$$

$$\neg A \sqcup \exists s.B, \exists s.B\}$$

$$L(v_1) = \{A, C_{\mathcal{T}}, \exists r^- .A, \forall r.(\neg A \sqcup \exists s.B)\}$$

$$L(v_2) = \{B, C_{\mathcal{T}}, \neg A, \forall r^- .(\neg A)\}$$

v_0 blocks v_1

Is the algorithm thus correct?

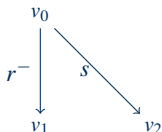
Tableau Example with Inverses

Example: is A satisfiable w.r.t. \mathcal{T} ?

$$\mathcal{T} = \{A \equiv \exists r^- .A \sqcap (\forall r.(\neg A \sqcup \exists s.B))\}$$

$$C_{\mathcal{T}} = (\neg A \sqcup \exists r^- .A) \sqcap (\neg A \sqcup \forall r.(\neg A \sqcup \exists s.B)) \sqcap$$

$$(\forall r^- .(\neg A) \sqcup \exists r.(A \sqcap \forall s.(\neg B)) \sqcup A)$$



$$L(v_0) = \{A, C_{\mathcal{T}}, \exists r^- .A, \forall r.(\neg A \sqcup \exists s.B),$$

$$\neg A \sqcup \exists s.B, \exists s.B\}$$

$$L(v_1) = \{A, C_{\mathcal{T}}, \exists r^- .A, \forall r.(\neg A \sqcup \exists s.B)\}$$

$$L(v_2) = \{B, C_{\mathcal{T}}, \neg A, \forall r^- .(\neg A)\}$$

v_0 blocks v_1

Is the algorithm thus correct? **No!**

Tableau Example with Inverses II

Example: Is $C \sqcap \exists s.C$ satisfiable w.r.t. \mathcal{T} ?

$$\mathcal{T} = \{T \sqsubseteq \forall r^-. (\forall s^-. (\neg C)) \sqcap \exists r.C\}$$

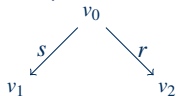
Tableau Example with Inverses II

Example: Is $C \sqcap \exists s.C$ satisfiable w.r.t. \mathcal{T} ?

$$\mathcal{T} = \{ \top \sqsubseteq \forall r^- . (\forall s^- . (\neg C)) \sqcap \exists r.C \}$$
$$C_{\mathcal{T}} = \forall r^- . (\forall s^- . (\neg C)) \sqcap \exists r.C$$

Tableau Example with Inverses II

Example: Is $C \sqcap \exists s.C$ satisfiable w.r.t. \mathcal{T} ?



$$\mathcal{T} = \{ \top \sqsubseteq \forall r^- . (\forall s^- . (\neg C)) \sqcap \exists r . C \}$$

$$C_{\mathcal{T}} = \forall r^- . (\forall s^- . (\neg C)) \sqcap \exists r . C$$

$$L(v_0) = \{ C, \exists s . C, C_{\mathcal{T}}, \forall r^- . (\forall s^- . (\neg C)), \exists r . C, \forall s^- . (\neg C) \}$$

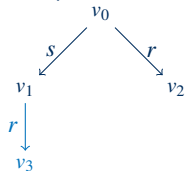
$$L(v_1) = \{ C, C_{\mathcal{T}}, \forall r^- . (\forall s^- . (\neg C)), \exists r . C \}$$

$$L(v_2) = \{ C, C_{\mathcal{T}}, \forall r^- . (\forall s^- . (\neg C)), \exists r . C \}$$

v_0 blocks v_1 and v_2

Tableau Example with Inverses II

Example: Is $C \sqcap \exists s.C$ satisfiable w.r.t. \mathcal{T} ?



$$\mathcal{T} = \{ \top \sqsubseteq \forall r^-. (\forall s^-. (\neg C)) \sqcap \exists r.C \}$$

$$C_{\mathcal{T}} = \forall r^-. (\forall s^-. (\neg C)) \sqcap \exists r.C$$

$$L(v_0) = \{ C, \exists s.C, C_{\mathcal{T}}, \forall r^-. (\forall s^-. (\neg C)), \exists r.C, \forall s^-. (\neg C) \}$$

$$L(v_1) = \{ C, C_{\mathcal{T}}, \forall r^-. (\forall s^-. (\neg C)), \exists r.C \}$$

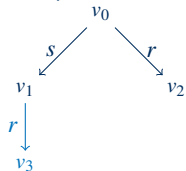
$$L(v_2) = \{ C, C_{\mathcal{T}}, \forall r^-. (\forall s^-. (\neg C)), \exists r.C \}$$

v_0 blocks v_1 and v_2 but

$$L(v_3) = \{ C, C_{\mathcal{T}}, \forall r^-. (\forall s^-. (\neg C)), \exists r.C \}$$

Tableau Example with Inverses II

Example: Is $C \sqcap \exists s.C$ satisfiable w.r.t. \mathcal{T} ?



$$\mathcal{T} = \{ \top \sqsubseteq \forall r^- . (\forall s^- . (\neg C)) \sqcap \exists r . C \}$$

$$C_{\mathcal{T}} = \forall r^- . (\forall s^- . (\neg C)) \sqcap \exists r . C$$

$$L(v_0) = \{ C, \exists s . C, C_{\mathcal{T}}, \forall r^- . (\forall s^- . (\neg C)), \exists r . C, \forall s^- . (\neg C) \}$$

$$L(v_1) = \{ C, C_{\mathcal{T}}, \forall r^- . (\forall s^- . (\neg C)), \exists r . C \} \cup \{ \forall s^- . (\neg C) \}$$

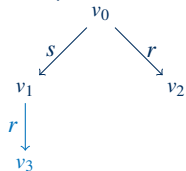
$$L(v_2) = \{ C, C_{\mathcal{T}}, \forall r^- . (\forall s^- . (\neg C)), \exists r . C \}$$

v_0 blocks v_1 and v_2 but

$$L(v_3) = \{ C, C_{\mathcal{T}}, \forall r^- . (\forall s^- . (\neg C)), \exists r . C \}$$

Tableau Example with Inverses II

Example: Is $C \sqcap \exists s.C$ satisfiable w.r.t. \mathcal{T} ?



$$\mathcal{T} = \{ \top \sqsubseteq \forall r^-. (\forall s^-. (\neg C)) \sqcap \exists r.C \}$$

$$C_{\mathcal{T}} = \forall r^-. (\forall s^-. (\neg C)) \sqcap \exists r.C$$

$$L(v_0) = \{ C, \exists s.C, C_{\mathcal{T}}, \forall r^-. (\forall s^-. (\neg C)), \exists r.C, \forall s^-. (\neg C) \} \cup \{ \neg C \}$$

$$L(v_1) = \{ C, C_{\mathcal{T}}, \forall r^-. (\forall s^-. (\neg C)), \exists r.C \} \cup \{ \forall s^-. (\neg C) \}$$

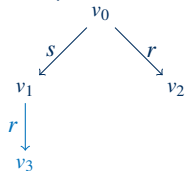
$$L(v_2) = \{ C, C_{\mathcal{T}}, \forall r^-. (\forall s^-. (\neg C)), \exists r.C \}$$

v_0 blocks v_1 and v_2 but

$$L(v_3) = \{ C, C_{\mathcal{T}}, \forall r^-. (\forall s^-. (\neg C)), \exists r.C \}$$

Tableau Example with Inverses II

Example: Is $C \sqcap \exists s.C$ satisfiable w.r.t. \mathcal{T} ?



$$\mathcal{T} = \{ \top \sqsubseteq \forall r^- . (\forall s^- . (\neg C)) \sqcap \exists r.C \}$$

$$C_{\mathcal{T}} = \forall r^- . (\forall s^- . (\neg C)) \sqcap \exists r.C$$

$$L(v_0) = \{ C, \exists s.C, C_{\mathcal{T}}, \forall r^- . (\forall s^- . (\neg C)), \exists r.C, \forall s^- . (\neg C) \} \cup \{ \neg C \}$$

$$L(v_1) = \{ C, C_{\mathcal{T}}, \forall r^- . (\forall s^- . (\neg C)), \exists r.C \} \cup \{ \forall s^- . (\neg C) \}$$

$$L(v_2) = \{ C, C_{\mathcal{T}}, \forall r^- . (\forall s^- . (\neg C)), \exists r.C \}$$

v_0 blocks v_1 and v_2 but

$$L(v_3) = \{ C, C_{\mathcal{T}}, \forall r^- . (\forall s^- . (\neg C)), \exists r.C \}$$

correctness can be retained by replacing subset blocking with equality blocking
i.e., replace $L(v') \subseteq L(v)$ by $L(v') = L(v)$ in the blocking condition

Model Construction for Tableau Example with Inverses II

We cannot build a cyclic model as we could up to now !

Example: Is $C \sqcap \exists s.C$ satisfiable w.r.t. \mathcal{T} ?



$$\mathcal{T} = \{ \top \sqsubseteq \forall r^- . (\forall s^- . (\neg C)) \sqcap \exists r . C \}$$

$$C_{\mathcal{T}} = \forall r^- . (\forall s^- . (\neg C)) \sqcap \exists r . C$$

$$L(v_0) = \{ C, \exists s . C, C_{\mathcal{T}}, \forall r^- . (\forall s^- . (\neg C)), \exists r . C, \forall s^- . (\neg C) \}$$

Model Construction for Tableau Example with Inverses II

We cannot build a cyclic model as we could up to now !

Example: Is $C \sqcap \exists s.C$ satisfiable w.r.t. \mathcal{T} ?



$$\mathcal{T} = \{ \top \sqsubseteq \forall r^- . (\forall s^- . (\neg C)) \sqcap \exists r . C \}$$

$$C_{\mathcal{T}} = \forall r^- . (\forall s^- . (\neg C)) \sqcap \exists r . C$$

$$L(v_0) = \{ C, \exists s . C, C_{\mathcal{T}}, \forall r^- . (\forall s^- . (\neg C)), \exists r . C, \forall s^- . (\neg C) \}$$

Model Construction for Tableau Example with Inverses II

We cannot build a cyclic model as we could up to now !

Example: Is $C \sqcap \exists s.C$ satisfiable w.r.t. \mathcal{T} ?



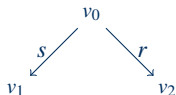
$$\mathcal{T} = \{ \top \sqsubseteq \forall r^- . (\forall s^- . (\neg C)) \sqcap \exists r . C \}$$

$$C_{\mathcal{T}} = \forall r^- . (\forall s^- . (\neg C)) \sqcap \exists r . C$$

$$L(v_0) = \{ C, \exists s . C, C_{\mathcal{T}}, \forall r^- . (\forall s^- . (\neg C)), \exists r . C, \forall s^- . (\neg C) \} \cup \{ \neg C \}$$

Example with Inverses & Equality Blocking

Example: Is $C \sqcap \exists s.C$ satisfiable w.r.t. \mathcal{T} ?



$$\mathcal{T} = \{ \top \sqsubseteq \forall r^- . (\forall s^- . (\neg C)) \sqcap \exists r.C \}$$

$$C_{\mathcal{T}} = \forall r^- . (\forall s^- . (\neg C)) \sqcap \exists r.C$$

$$L(v_0) = \{ C, \exists s.C, C_{\mathcal{T}}, \forall r^- . (\forall s^- . (\neg C)), \exists r.C, \forall s^- . (\neg C) \}$$

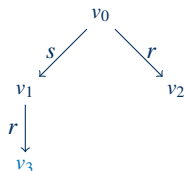
$$L(v_1) = \{ C, C_{\mathcal{T}}, \forall r^- . (\forall s^- . (\neg C)), \exists r.C \}$$

$$L(v_2) = \{ C, C_{\mathcal{T}}, \forall r^- . (\forall s^- . (\neg C)), \exists r.C \}$$

v_1 blocks v_2 (same labels)

Example with Inverses & Equality Blocking

Example: Is $C \sqcap \exists s.C$ satisfiable w.r.t. \mathcal{T} ?



$$\mathcal{T} = \{ \top \sqsubseteq \forall r^- . (\forall s^- . (\neg C)) \sqcap \exists r.C \}$$

$$C_{\mathcal{T}} = \forall r^- . (\forall s^- . (\neg C)) \sqcap \exists r.C$$

$$L(v_0) = \{ C, \exists s.C, C_{\mathcal{T}}, \forall r^- . (\forall s^- . (\neg C)), \exists r.C, \forall s^- . (\neg C) \}$$

$$L(v_1) = \{ C, C_{\mathcal{T}}, \forall r^- . (\forall s^- . (\neg C)), \exists r.C \}$$

$$L(v_2) = \{ C, C_{\mathcal{T}}, \forall r^- . (\forall s^- . (\neg C)), \exists r.C \}$$

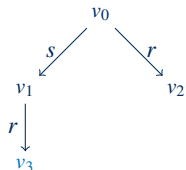
v_1 blocks v_2 (same labels)

$$L(v_3) = \{ C, C_{\mathcal{T}}, \forall r^- . (\forall s^- . (\neg C)), \exists r.C \}$$

v_1 blocks v_3 but \forall -rule applicable

Example with Inverses & Equality Blocking

Example: Is $C \sqcap \exists s.C$ satisfiable w.r.t. \mathcal{T} ?



$$\mathcal{T} = \{ \top \sqsubseteq \forall r^- . (\forall s^- . (\neg C)) \sqcap \exists r.C \}$$

$$C_{\mathcal{T}} = \forall r^- . (\forall s^- . (\neg C)) \sqcap \exists r.C$$

$$L(v_0) = \{ C, \exists s.C, C_{\mathcal{T}}, \forall r^- . (\forall s^- . (\neg C)), \exists r.C, \forall s^- . (\neg C) \}$$

$$L(v_1) = \{ C, C_{\mathcal{T}}, \forall r^- . (\forall s^- . (\neg C)), \exists r.C \} \cup \{ \forall s^- . (\neg C) \}$$

$$L(v_2) = \{ C, C_{\mathcal{T}}, \forall r^- . (\forall s^- . (\neg C)), \exists r.C \}$$

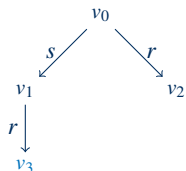
v_1 blocks v_2 (same labels)

$$L(v_3) = \{ C, C_{\mathcal{T}}, \forall r^- . (\forall s^- . (\neg C)), \exists r.C \}$$

~~v_1 blocks v_3 but \forall rule applicable~~

Example with Inverses & Equality Blocking

Example: Is $C \sqcap \exists s.C$ satisfiable w.r.t. \mathcal{T} ?



$$\mathcal{T} = \{ \top \sqsubseteq \forall r^- . (\forall s^- . (\neg C)) \sqcap \exists r.C \}$$

$$C_{\mathcal{T}} = \forall r^- . (\forall s^- . (\neg C)) \sqcap \exists r.C$$

$$L(v_0) = \{ C, \exists s.C, C_{\mathcal{T}}, \forall r^- . (\forall s^- . (\neg C)), \exists r.C, \forall s^- . (\neg C) \} \cup \{ \neg C \}$$

$$L(v_1) = \{ C, C_{\mathcal{T}}, \forall r^- . (\forall s^- . (\neg C)), \exists r.C \} \cup \{ \forall s^- . (\neg C) \}$$

$$L(v_2) = \{ C, C_{\mathcal{T}}, \forall r^- . (\forall s^- . (\neg C)), \exists r.C \}$$

v_1 blocks v_2 (same labels)

$$L(v_3) = \{ C, C_{\mathcal{T}}, \forall r^- . (\forall s^- . (\neg C)), \exists r.C \}$$

~~v_1 blocks v_3 but \forall rule applicable~~

Now unsatisfiability is recognized!

Agenda

- Recap Tableau Calculus
- Tableau with \mathcal{ALC} TBoxes
- Tableau for \mathcal{ALC} Knowledge Bases
- Extension by Inverse Roles
- [Extension by Functional Roles](#)
- Model Construction with Unravelling
- Summary

Tableau with Functional Roles

Example: is A satisfiable w.r.t. \mathcal{T} ?

Note: $\top \sqsubseteq \leq 1f$ is equivalent to $\text{Func}(f)$

$$\mathcal{T} = \{A \sqsubseteq \exists f.B \sqcap \exists f.(\neg B), \top \sqsubseteq \leq 1f\}$$

Tableau with Functional Roles

Example: is A satisfiable w.r.t. \mathcal{T} ?

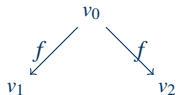
Note: $\top \sqsubseteq \leq 1f$ is equivalent to $\text{Func}(f)$

$$\begin{aligned}\mathcal{T} &= \{A \sqsubseteq \exists f.B \sqcap \exists f.(\neg B), \top \sqsubseteq \leq 1f\} \\ C_{\mathcal{T}} &= (\neg A \sqcup (\exists f.B \sqcap \exists f.(\neg B))) \sqcap \leq 1f\end{aligned}$$

Tableau with Functional Roles

Example: is A satisfiable w.r.t. \mathcal{T} ?

Note: $\top \sqsubseteq \leq 1f$ is equivalent to $\text{Func}(f)$



$$\mathcal{T} = \{A \sqsubseteq \exists f.B \sqcap \exists f.(\neg B), \top \sqsubseteq \leq 1f\}$$

$$C_{\mathcal{T}} = (\neg A \sqcup (\exists f.B \sqcap \exists f.(\neg B))) \sqcap \leq 1f$$

$$L(v_0) = \{A, C_{\mathcal{T}}, \dots, \exists f.B, \exists f.(\neg B), \leq 1f\}$$

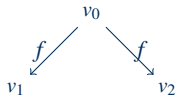
$$L(v_1) = \{B, C_{\mathcal{T}}, \dots, \neg A, \leq 1f\}$$

$$L(v_2) = \{\neg B, C_{\mathcal{T}}, \dots, \neg A, \leq 1f\}$$

Tableau with Functional Roles

Example: is A satisfiable w.r.t. \mathcal{T} ?

Note: $\top \sqsubseteq \leq 1f$ is equivalent to $\text{Func}(f)$



$$\mathcal{T} = \{A \sqsubseteq \exists f.B \sqcap \exists f.(\neg B), \top \sqsubseteq \leq 1f\}$$

$$C_{\mathcal{T}} = (\neg A \sqcup (\exists f.B \sqcap \exists f.(\neg B))) \sqcap \leq 1f$$

$$L(v_0) = \{A, C_{\mathcal{T}}, \dots, \exists f.B, \exists f.(\neg B), \leq 1f\}$$

$$L(v_1) = \{B, C_{\mathcal{T}}, \dots, \neg A, \leq 1f\}$$

$$L(v_2) = \{\neg B, C_{\mathcal{T}}, \dots, \neg A, \leq 1f\}$$

functionality requires $v_1 = v_2$!

\rightsquigarrow we need a new tableau rule for treating functional roles

Tableau Rules for \mathcal{ALCIF} Concepts and TBoxes

- \sqcap -rule: For an $v \in V$ with $C \sqcap D \in L(v)$ and $\{C, D\} \not\subseteq L(v)$, let $L(v) := L(v) \cup \{C, D\}$.
- \sqcup -rule: For an $v \in V$ with $C \sqcup D \in L(v)$ and $\{C, D\} \cap L(v) = \emptyset$, choose $X \in \{C, D\}$ and let $L(v) := L(v) \cup \{X\}$.
- \exists -rule: For a non-blocked $v \in V$ with $\exists r.C \in L(v)$ such that there is no r -neighbor v' of v with $C \in L(v')$, let $V = V \cup \{v'\}$, $E = E \cup \{(v, v')\}$, $L(v') := \{C\}$ and $L(v, v') := \{r\}$ for v' a new node.
- \forall -rule: For $v, v' \in V$, v' r -neighbor of v , $\forall r.C \in L(v)$ and $C \notin L(v')$, let $L(v') := L(v') \cup \{C\}$.
- ≤ 1 -rule: For a functional role f and a $v \in V$ with two f -neighbors v_1 and v_2 , execute $\text{merge}(v_1, v_2)$.
- \mathcal{T} -rule: For a $v \in V$ with $C_{\mathcal{T}} \notin L(v)$, let $L(v) := L(v) \cup \{C_{\mathcal{T}}\}$.

Merging Nodes

we define $\text{merge}(v_1, v_2)$ as follows:

- if v_1 is an ancestor of v_2 ,
let $v_i = v_1$ and $v_o = v_2$;
- otherwise let $v_i = v_2$ and $v_o = v_1$.

let $L(v_i) = L(v_i) \cup L(v_o)$ and execute $\text{prune}(v_o)$.

where $\text{prune}(v_o)$ is defined as:

- $V_o = \{v \mid v \text{ belongs to the subtree with root } v_o\}$,
- let $V = V \setminus V_o$ and $E = E \setminus \{\langle v, v_o \rangle \mid v_o \in V_o, \langle v, v_o \rangle \in E\}$.

Tableau with Functional Roles

Example: Is $\exists f.A$ satisfiable w.r.t. \mathcal{T} ?

$$\mathcal{T} = \{A \sqsubseteq \exists f.A, \top \sqsubseteq \leq 1f^-\}$$

Tableau with Functional Roles

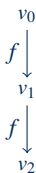
Example: Is $\exists f.A$ satisfiable w.r.t. \mathcal{T} ?

$$\mathcal{T} = \{A \sqsubseteq \exists f.A, \top \sqsubseteq \leq 1f^-\}$$

$$C_{\mathcal{T}} = (\neg A \sqcup \exists f.A) \sqcap \leq 1f^-$$

Tableau with Functional Roles

Example: Is $\exists f.A$ satisfiable w.r.t. \mathcal{T} ?



$$\mathcal{T} = \{A \sqsubseteq \exists f.A, \top \sqsubseteq \leq 1f^-\}$$

$$C_{\mathcal{T}} = (\neg A \sqcup \exists f.A) \sqcap \leq 1f^-$$

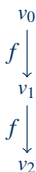
$$L(v_0) = \{\exists f.A, C_{\mathcal{T}}, \neg A, \leq 1f^-\}$$

$$L(v_1) = \{A, C_{\mathcal{T}}, \exists f.A, \leq 1f^-\}$$

$$L(v_2) = \{A, C_{\mathcal{T}}, \exists f.A, \leq 1f^-\}$$

Tableau with Functional Roles

Example: Is $\exists f.A$ satisfiable w.r.t. \mathcal{T} ?



$$\mathcal{T} = \{A \sqsubseteq \exists f.A, \top \sqsubseteq \leq 1f^-\}$$

$$C_{\mathcal{T}} = (\neg A \sqcup \exists f.A) \sqcap \leq 1f^-$$

$$L(v_0) = \{\exists f.A, C_{\mathcal{T}}, \neg A, \leq 1f^-\}$$

$$L(v_1) = \{A, C_{\mathcal{T}}, \exists f.A, \leq 1f^-\}$$

$$L(v_2) = \{A, C_{\mathcal{T}}, \exists f.A, \leq 1f^-\}$$

v_1 blocks v_2 , but cyclic model construction does not work (functionality violated)!



Agenda

- Recap Tableau Calculus
- Tableau with \mathcal{ALC} TBoxes
- Tableau for \mathcal{ALC} Knowledge Bases
- Extension by Inverse Roles
- Extension by Functional Roles
- Model Construction with Unravelling
- Summary

Unravelling

goal: we build an infinite model

How? Every blocked node is replaced by a subtree whose root is the corresponding blocking node.



$$L(v_0) = \{\exists f.A, C_{\mathcal{T}}, \neg A, \leq 1f^{-}\}$$

$$L(v_1) = \{A, C_{\mathcal{T}}, \exists f.A, \leq 1f^{-}\}$$

$$L(v_2) = \{A, C_{\mathcal{T}}, \exists f.A, \leq 1f^{-}\}$$

v_1 blocks v_2

Unravelling

goal: we build an infinite model

How? Every blocked node is replaced by a subtree whose root is the corresponding blocking node.



$$L(v_0) = \{\exists f.A, C_{\mathcal{T}}, \neg A, \leq 1f^-\}$$

$$L(v_1) = \{A, C_{\mathcal{T}}, \exists f.A, \leq 1f^-\}$$

$$L(v_2) = \{A, C_{\mathcal{T}}, \exists f.A, \leq 1f^-\}$$

v_1 blocks v_2

Unravelling

goal: we build an infinite model

How? Every blocked node is replaced by a subtree whose root is the corresponding blocking node.



$$L(v_0) = \{\exists f.A, C_{\mathcal{T}}, \neg A, \leq 1f^-\}$$

$$L(v_1) = \{A, C_{\mathcal{T}}, \exists f.A, \leq 1f^-\}$$

$$L(v_2) = \{A, C_{\mathcal{T}}, \exists f.A, \leq 1f^-\}$$

v_1 blocks v_2

Unravelling

goal: we build an infinite model

How? Every blocked node is replaced by a subtree whose root is the corresponding blocking node.



$$L(v_0) = \{\exists f.A, C_{\mathcal{T}}, \neg A, \leq 1f^-\}$$

$$L(v_1) = \{A, C_{\mathcal{T}}, \exists f.A, \leq 1f^-\}$$

$$L(v_2) = \{A, C_{\mathcal{T}}, \exists f.A, \leq 1f^-\}$$

v_1 blocks v_2

Blocking: Inverse and Functional Roles

Example: Is $\neg C \sqcap \exists f^{-}.D$ satisfiable w.r.t. \mathcal{T} ?

$$\mathcal{T} = \{D \sqsubseteq C \sqcap \exists f.(\neg C) \sqcap \exists f^{-}.D, \top \sqsubseteq \leq 1f\}$$

Blocking: Inverse and Functional Roles

Example: Is $\neg C \sqcap \exists f^{-}.D$ satisfiable w.r.t. \mathcal{T} ?

$$\mathcal{T} = \{D \sqsubseteq C \sqcap \exists f.(\neg C) \sqcap \exists f^{-}.D, \top \sqsubseteq \leq 1f\}$$

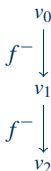
$$C_{\mathcal{T}} = (\neg D \sqcup (C \sqcap \exists f.(\neg C) \sqcap \exists f^{-}.D)) \sqcap \leq 1f$$

Blocking: Inverse and Functional Roles

Example: Is $\neg C \sqcap \exists f^- . D$ satisfiable w.r.t. \mathcal{T} ?

$$\mathcal{T} = \{D \sqsubseteq C \sqcap \exists f. (\neg C) \sqcap \exists f^- . D, \top \sqsubseteq \leq 1f\}$$

$$C_{\mathcal{T}} = (\neg D \sqcup (C \sqcap \exists f. (\neg C) \sqcap \exists f^- . D)) \sqcap \leq 1f$$



$$L(v_0) = \{\neg C, \exists f^- . D, C_{\mathcal{T}}, \dots, \neg D, \leq 1f\}$$

$$L(v_1) = \{D, C_{\mathcal{T}}, \dots, C, \exists f. (\neg C), \exists f^- . D, \leq 1f\}$$

$$L(v_2) = \{D, C_{\mathcal{T}}, \dots, C, \exists f. (\neg C), \exists f^- . D, \leq 1f\}$$

v_1 blocks v_2 (same label)

Blocking: Inverse and Functional Roles

Example: Is $\neg C \sqcap \exists f^- . D$ satisfiable w.r.t. \mathcal{T} ?

$$\mathcal{T} = \{D \sqsubseteq C \sqcap \exists f. (\neg C) \sqcap \exists f^- . D, \top \sqsubseteq \leq 1f\}$$

$$C_{\mathcal{T}} = (\neg D \sqcup (C \sqcap \exists f. (\neg C) \sqcap \exists f^- . D)) \sqcap \leq 1f$$



$$L(v_0) = \{\neg C, \exists f^- . D, C_{\mathcal{T}}, \dots, \neg D, \leq 1f\}$$

$$L(v_1) = \{D, C_{\mathcal{T}}, \dots, C, \exists f. (\neg C), \exists f^- . D, \leq 1f\}$$

$$L(v'_1) = \{D, C_{\mathcal{T}}, \dots, C, \exists f. (\neg C), \exists f^- . D, \leq 1f\}$$

v_1 blocks v_2 (same label) but

$$L(v''_1) = \{D, C_{\mathcal{T}}, \dots, C, \exists f. (\neg C), \exists f^- . D, \leq 1f\}$$

but we cannot build a model any more (neither cyclic nor infinite)!

Pairwise Blocking

A node x with predecessor x' blocks a node y with predecessor y' directly, if:

- 1 y is reachable from x ,
- 2 $L(x) = L(y)$, $L(x') = L(y')$ and $L(x', x) = L(y', y)$; and
- 3 there is no directly blocked node z such that y is reachable from z .

A node $y \in V$ is **blocked** if either

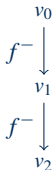
- 1 y is directly blocked or
- 2 there is a directly blocked node x , such that y can be reached from x .

Pairwise Blocking: Inverses and Functional Roles

Example: Is $\neg C \sqcap \exists f^- . D$ satisfiable w.r.t. \mathcal{T} ?

$$\mathcal{T} = \{D \sqsubseteq C \sqcap \exists f^- . (\neg C) \sqcap \exists f^- . D, \top \sqsubseteq \leq 1f\}$$

$$C_{\mathcal{T}} = (\neg D \sqcup (C \sqcap \exists f^- . (\neg C) \sqcap \exists f^- . D)) \sqcap \leq 1f$$



$$L(v_0) = \{\neg C, \exists f^- . D, C_{\mathcal{T}}, \dots, \neg D, \leq 1f\}$$

$$L(v_1) = \{D, C_{\mathcal{T}}, \dots, C, \exists f^- . (\neg C), \exists f^- . D, \leq 1f\}$$

$$L(v_2) = \{D, C_{\mathcal{T}}, \dots, C, \exists f^- . (\neg C), \exists f^- . D, \leq 1f\}$$

v_1 cannot block v_2 pairwise

Pairwise Blocking: Inverses and Functional Roles

Example: Is $\neg C \sqcap \exists f^- . D$ satisfiable w.r.t. \mathcal{T} ?

$$\mathcal{T} = \{D \sqsubseteq C \sqcap \exists f^- . (\neg C) \sqcap \exists f^- . D, \top \sqsubseteq \leq 1f\}$$

$$C_{\mathcal{T}} = (\neg D \sqcup (C \sqcap \exists f^- . (\neg C) \sqcap \exists f^- . D)) \sqcap \leq 1f$$



$$L(v_0) = \{\neg C, \exists f^- . D, C_{\mathcal{T}}, \dots, \neg D, \leq 1f\}$$

$$L(v_1) = \{D, C_{\mathcal{T}}, \dots, C, \exists f^- . (\neg C), \exists f^- . D, \leq 1f\}$$

$$L(v_2) = \{D, C_{\mathcal{T}}, \dots, C, \exists f^- . (\neg C), \exists f^- . D, \leq 1f\}$$

v_1 cannot block v_2 pairwise

$$L(v_3) = \{\neg C, C_{\mathcal{T}}, \dots, \neg D, \leq 1f\}$$

Pairwise Blocking: Inverses and Functional Roles

Example: Is $\neg C \sqcap \exists f^- . D$ satisfiable w.r.t. \mathcal{T} ?

$$\mathcal{T} = \{D \sqsubseteq C \sqcap \exists f^- . (\neg C) \sqcap \exists f^- . D, \top \sqsubseteq \leq 1f\}$$

$$C_{\mathcal{T}} = (\neg D \sqcup (C \sqcap \exists f^- . (\neg C) \sqcap \exists f^- . D)) \sqcap \leq 1f$$



$$L(v_0) = \{\neg C, \exists f^- . D, C_{\mathcal{T}}, \dots, \neg D, \leq 1f\}$$

$$L(v_1) = \{D, C_{\mathcal{T}}, \dots, C, \exists f^- . (\neg C), \exists f^- . D, \leq 1f\}$$

$$L(v_2) = \{D, C_{\mathcal{T}}, \dots, C, \exists f^- . (\neg C), \exists f^- . D, \leq 1f\}$$

v_1 cannot block v_2 pairwise

$$L(v_3) = \{\neg C, C_{\mathcal{T}}, \dots, \neg D, \leq 1f\}$$

Pairwise Blocking: Inverses and Functional Roles

Example: Is $\neg C \sqcap \exists f^- . D$ satisfiable w.r.t. \mathcal{T} ?

$$\mathcal{T} = \{D \sqsubseteq C \sqcap \exists f. (\neg C) \sqcap \exists f^- . D, \top \sqsubseteq \leq 1f\}$$

$$C_{\mathcal{T}} = (\neg D \sqcup (C \sqcap \exists f. (\neg C) \sqcap \exists f^- . D)) \sqcap \leq 1f$$



$$L(v_0) = \{\neg C, \exists f^- . D, C_{\mathcal{T}}, \dots, \neg D, \leq 1f\}$$

$$L(v_1) = \{D, C_{\mathcal{T}}, \dots, C, \exists f. (\neg C), \exists f^- . D, \leq 1f\}$$

$$L(v_2) = \{D, C_{\mathcal{T}}, \dots, C, \exists f. (\neg C), \exists f^- . D, \leq 1f\}$$

v_1 cannot block v_2 pairwise

$$L(v_3) = \{\neg C, C_{\mathcal{T}}, \dots, \neg D, \leq 1f\}$$

v_3 is merged into v_1

$$L(v_1) = L(v_1) \cup L(v_3) \supseteq \{\neg D, D\}$$

now the contradiction can be detected

Agenda

- Recap Tableau Calculus
- Tableau with \mathcal{ALC} TBoxes
- Tableau for \mathcal{ALC} Knowledge Bases
- Extension by Inverse Roles
- Extension by Functional Roles
- Model Construction with Unravelling
- Summary

Summary

- we now have a tableau algorithm for \mathcal{ALCIF} knowledge bases
 - treat the ABox like for \mathcal{ALC}
 - number restrictions can be handled similar to functional roles
- termination through cycle detection
 - becomes harder the more expressive the logic gets