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) \(\Rightarrow\) 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) $\Rightarrow$ easier rules
- tableau (model abstraction) corresponds to a graph/tree $G = \langle V, E, L \rangle$

TU Dresden, 14 May 2018 Deduction Systems slide 6 of 80
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) $\leadsto$ 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) $\Rightarrow$ 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 ALC Concepts and TBoxes

- check of satisfiability of $C$ by construction of an abstraction of a model $I$ such that $C^I \neq \emptyset$
- concepts in negation normal form (NNF) $\leadsto$ 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) \( \Rightarrow \) 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) $\leadsto$ 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
  - $\sqcap$ 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 $v, v' \in V$, $v'$ $r$-successor of $v$,
$\forall r. C \in L(v)$ and $C \notin L(v')$, let $L(v') := L(v') \cup \{C\}$.
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):

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

is equivalent to:

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

Let $C_T$ 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 $C_T \notin L(v)$, let $L(v) := L(v) \cup \{C_T\}$.

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 $C_T \notin L(v)$, let $L(v) := L(v) \cup \{C_T\}$.

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 $C_T \notin L(v)$,
let $L(v) := L(v) \cup \{C_T\}$.

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

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

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

we obtain the following contradiction-free tableau:

\[
\begin{align*}
L(v_0) &= \{A, C_\mathcal{T}, \exists r.A\} \\
L(v_1) &= \{A, C_\mathcal{T}, \exists r.A\}
\end{align*}
\]

wherein $v_1$ is directly blocked by $v_0$
Tableau Algorithm with Blocking

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

we obtain the following contradiction-free tableau:

\[
\begin{align*}
v_0 & \quad L(v_0) = \{A, C_T, \exists r.A\} \\
r & \\
v_1 & \quad L(v_1) = \{A, C_T, \exists r.A\}
\end{align*}
\]

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:

\[ \Delta^I = \{ v_0 \} \]
\[ A^I = \Delta^I \]
\[ r^I = \{ \langle v_0, v_0 \rangle \} \]

- 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 \( v' \)

\[ \sim \sim \text{ we have the finite model property} \]
\[ \sim \sim \text{ constructed model is not necessarily tree-shaped} \]
Example: Let $\mathcal{T} = A \subseteq \exists r.A \sqcap \exists s.B$. Is $A$ satisfiable w.r.t. $\mathcal{T}$?

We obtain the following contradiction-free tableau:

$$
\begin{align*}
L(v_0) &= \{A, C_\mathcal{T}, \exists r.A \sqcap \exists s.B, \exists r.A, \exists s.B\} \\
L(v_1) &= \{A, C_\mathcal{T}, \exists r.A \sqcap \exists s.B, \exists r.A, \exists s.B\} \\
L(v_2) &= \{B, C_\mathcal{T}, \neg A\}
\end{align*}
$$

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:

\[ \Delta^I = \{v_0, v_2\} \]
\[ A^I = \{v_0\} \]
\[ B^I = \{v_2\} \]
\[ r^I = \{\langle v_0, v_0 \rangle\} \]
\[ s^I = \{\langle v_0, v_2 \rangle\} \]
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 $\mathcal{A}$

the tableau rules can then be applied to this initialized graph
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)
Example: is $A$ satisfiable w.r.t. $\mathcal{T}$?

$$\mathcal{T} = \{ A \equiv \exists r . \neg A \land (\forall r . (\neg A \lor \exists s . B)) \}$$

Is the algorithm thus correct? No!
Tableau Example with Inverses

Example: is \( A \) satisfiable w.r.t. \( T \)?

\[
T = \{ A \equiv \exists r^- . A \land (\forall r . (\neg A \lor \exists s . B)) \}
\]

\[
C_T = (\neg A \lor \exists r^- . A) \land (\neg A \lor \forall r . (\neg A \lor \exists s . B)) \land
(\forall r^- . (\neg A) \lor \exists r . (A \land \forall s . (\neg B)) \lor A)
\]
Example: is $A$ satisfiable w.r.t. $\mathcal{T}$?

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

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

$$\land (\forall s^- . (\neg A) \lor \exists r . (A \land \forall s . (\neg B)) \lor A)$$

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

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

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

$v_0$ blocks $v_1$
Tableau Example with Inverses

Example: is \( A \) satisfiable w.r.t. \( T \)?

\[
T = \{ A \equiv \exists r^-. A \cap (\forall r.(\neg A \cup \exists s.B)) \}
\]

\[
C_T = (\neg A \cup \exists r^- . A) \cap (\neg A \cup \forall r.(\neg A \cup \exists s.B)) \cap
\]

\[
(\forall r^-.(\neg A) \cup \exists r.(A \cap \forall s.(\neg B)) \cup A)
\]

\[
L(v_0) = \{ A, C_T, \exists r^- . A, \forall r.(\neg A \cup \exists s.B),
\]

\[
\neg A \cup \exists s.B, \exists s.B \}
\]

\[
L(v_1) = \{ A, C_T, \exists r^- . A, \forall r.(\neg A \cup \exists s.B) \}
\]

\[
L(v_2) = \{ B, C_T, \neg A, \forall r^-.(\neg A) \}
\]

Is the algorithm thus correct?

\( v_0 \) blocks \( v_1 \)
Tableau Example with Inverses

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

\[
\mathcal{T} = \{ A \equiv \exists r^- . A \land (\forall r. (\neg A \lor \exists s . B)) \}
\]

\[
C_{\mathcal{T}} = (\neg A \lor \exists r^- . A) \land (\neg A \lor \forall r. (\neg A \lor \exists s . B)) \land \\
(\forall r^- . (\neg A) \lor \exists r . (A \lor \forall s . (\neg B)) \lor A)
\]

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

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

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

\( v_0 \) blocks \( v_1 \)

Is the algorithm thus correct? No!
Example: Is $C \cap \exists s. C$ satisfiable w.r.t. $\mathcal{T}$?

$$\mathcal{T} = \{ \top \subseteq \forall r^-.(\forall s^-.(\neg C)) \cap \exists r.C \}$$
Tableau Example with Inverses II

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

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

$$C_{\mathcal{T}} = \forall r^-.(\forall s^-.(\neg C)) \cap \exists r.C$$
Tableau Example with Inverses II

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

Example:

\[ T = \{ \top \sqsubseteq \forall r^-. (\forall s^-. (\neg C)) \sqcap \exists r. C \} \]
\[ C_T = \forall r^-.(\forall s^-.(\neg C)) \sqcap \exists r. C \]

\[ L(v_0) = \{ C, \exists s. C, C_T, \forall r^-. (\forall s^-.(\neg C)), \exists r. C, \forall s^-.(\neg C) \} \]
\[ L(v_1) = \{ C, C_T, \forall r^-. (\forall s^-.(\neg C)), \exists r. C \} \]
\[ L(v_2) = \{ C, C_T, \forall r^-.(\forall s^-.(\neg C)), \exists r. C \} \]

\[ v_0 \text{ blocks } v_1 \text{ and } v_2 \leadsto \text{tableau complete} \]
Tableau Example with Inverses II

Example: Is $C \cap \exists s. C$ satisfiable w.r.t. $T$?

$T = \{ \top \subseteq \forall r^-.(\forall s^-.(\neg C)) \cap \exists r.C \}$

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

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

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

$v_0$ blocks $v_1$ and $v_2 \leadsto$ tableau complete but $v_0 \leadsto$ tableau incomplete

$L(v_3) = \{ C, C_T, \forall r^-.(\forall s^-.(\neg C)), \exists r.C \}$
Example: Is \( C \cap \exists s.C \) satisfiable w.r.t. \( T \)?

\[
T = \{ \top \subseteq \forall r^-.(\forall s^-.(\neg C)) \cap \exists r.C \}
\]

\[
C_T = \forall r^-.(\forall s^-.(\neg C)) \cap \exists r.C
\]

\[
L(v_0) = \{ C, \exists s.C, C_T, \forall r^-.(\forall s^-.(\neg C)), \exists r.C, \forall s^-.(\neg C) \}
\]

\[
L(v_1) = \{ C, C_T, \forall r^-.(\forall s^-.(\neg C)), \exists r.C \} \cup \{ \forall s^-.(\neg C) \}
\]

\[
L(v_2) = \{ C, C_T, \forall r^-.(\forall s^-.(\neg C)), \exists r.C \}
\]

\[v_0 \text{ blocks } v_1 \text{ and } v_2 \leadsto \text{tableau complete but}\]

\[
L(v_3) = \{ C, C_T, \forall r^-.(\forall s^-.(\neg C)), \exists r.C \}
\]
Tableau Example with Inverses II

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

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

$$C_T = \forall r^-.(\forall s^-.(\neg C)) \sqcap \exists r. C$$

$$L(v_0) = \{ C, \exists s.C, C_T, \forall r^-.(\forall s^-.(\neg C)), \exists r. C, \forall s^-.(\neg C) \} \sqcup \{ \neg C \}$$

$$L(v_1) = \{ C, C_T, \forall r^-.(\forall s^-.(\neg C)), \exists r. C \} \sqcup \{ \forall s^-.(\neg C) \}$$

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

$v_0$ blocks $v_1$ and $v_2 \leadsto$ tableau complete but

$$L(v_3) = \{ C, C_T, \forall r^-.(\forall s^-.(\neg C)), \exists r. C \}$$
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 $v_2 \leadsto$ tableau complete but

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

We have blocked too early!
Example: Is $C \sqcap \exists s.C$ satisfiable w.r.t. $T$?

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

$C_T = \forall r^-.(\forall s^-.(\neg C)) \sqcap \exists r.C$

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

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

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

$v_0$ blocks $v_1$ and $v_2 \leadsto$ tableau complete  but

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

We have blocked too early! 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.
Why does subset blocking not work anymore?
We cannot build a cyclic model as we could up to now!

Example: early blocked tableau from previous example would yield:

\[
\begin{array}{c}
\neg r, s \\
\exists v_0 \ C
\end{array}
\]

However, this is not a model of \( T \subseteq \forall r^{-}.(\forall s^{-}.(\neg C)) \cap \exists r.C \).
Example with Inverses & Equality Blocking

Example: Is $C \cap \exists s.C$ satisfiable w.r.t. $T$?

Example:

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

$$
C_T = \forall r^-.(\forall s^-.(\neg C)) \cap \exists r.C
$$

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

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

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

$v_1$ blocks $v_3$ but $\forall$-rule applicable

Now unsatisfiability is recognized!
Example with Inverses & Equality Blocking

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

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

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

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

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

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

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

$v_1$ blocks $v_3$ but $\forall$-rule applicable
Example with Inverses & Equality Blocking

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

Let $
\begin{align*}
\mathcal{T} &= \{ \top \subseteq \forall r^-.(\forall s^-.(\neg C)) \cap \exists r.C \} \\
C_{\mathcal{T}} &= \forall r^-.(\forall s^-.(\neg C)) \cap \exists r.C
\end{align*}
$

Let
\begin{align*}
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 \} \\
L(v_3) &= \{ C, C_{\mathcal{T}}, \forall r^-.(\forall s^-.(\neg C)), \exists r.C \}
\end{align*}

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

Now unsatisfiability is recognized!
Example with Inverses & Equality Blocking

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

\[
\mathcal{T} = \{ \top \sqsubseteq \forall r^-.\forall s^-.\neg C \} \cap \exists r.C
\]

\[
C_{\mathcal{T}} = \forall r^-.\forall s^-.\neg C \cap \exists r.C
\]

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

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

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

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

$v_1$ blocks $v_2$ but $\forall$ rule applicable

Now unsatisfiability is recognized!
Agenda

- Recap Tableau Calculus
- Tableau with \( ALC \) TBoxes
- Tableau for \( ALC \) 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. $T$?

Note: $\top \sqsubseteq 1_f$ expresses functionality of the role $f$

$$T = \{ A \sqsubseteq \exists f.B \cap \exists f.(\neg B), \top \sqsubseteq 1_f \}$$
Tableau with Functional Roles

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

Note: \( \top \sqsubseteq \leq_1 f \) expresses functionality of the role \( f \)

\[
\mathcal{T} = \{ A \sqsubseteq \exists f.B \cap \exists f.(-B), \top \sqsubseteq \leq_1 f \}
\]

\[
C_\mathcal{T} = (\neg A \cup (\exists f.B \cap \exists f.(-B))) \cap \leq_1 f
\]
Tableau with Functional Roles

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

Note: $\top \sqsubseteq 1f$ expresses functionality of the role $f$

\[
\mathcal{T} = \{ A \sqsubseteq \exists f. B \sqcap \exists f. (\neg B), \top \sqsubseteq 1f \} \\
C_{\mathcal{T}} = (\neg A \sqcup (\exists f. B \sqcap \exists f. (\neg B))) \sqcap 1f
\]

\[
L(v_0) = \{ A, C_{\mathcal{T}}, \ldots, \exists f. B, \exists f. (\neg B), 1f \} \\
L(v_1) = \{ B, C_{\mathcal{T}}, \ldots, \neg A, 1f \} \\
L(v_2) = \{ \neg B, C_{\mathcal{T}}, \ldots, \neg A, 1f \}
\]
Tableau with Functional Roles

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

Note: $\top \sqsubseteq f \leq 1$ expresses functionality of the role $f$

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

$$C_T = \neg A \sqcup (\exists f.B \cap \exists f.(-B)) \sqcap \leq 1f$$

$$L(v_0) = \{A, C_T, \ldots, \exists f.B, \exists f.(-B), \leq 1f\}$$

$$L(v_1) = \{B, C_T, \ldots, \neg A, \leq 1f\}$$

$$L(v_2) = \{\neg B, C_T, \ldots, \neg A, \leq 1f\}$$

Functionality requires $v_1 = v_2$!

$\leadsto$ we need a new tableau rule for treating functional roles
Tableau Rules for $\textit{ALCIF}$ Concepts and TBoxes

\[\begin{array}{ll}
\sqcap\text{-rule:} & \text{For an } v \in V \text{ with } C \sqcap D \in L(v) \text{ and } \\
& \{C, D\} \not\subseteq L(v), \text{ let } L(v) := L(v) \cup \{C, D\}.
\\
\sqcup\text{-rule:} & \text{For an } v \in V \text{ with } C \sqcup D \in L(v) \text{ and } \\
& \{C, D\} \cap L(v) = \emptyset, \text{ choose } X \in \{C, D\} \text{ and let } \\
& L(v) := L(v) \cup \{X\}.
\\
\exists\text{-rule:} & \text{For a non-blocked } v \in V \text{ with } \exists_r C \in L(v) \text{ such that } \\
& \text{there is no } r\text{-neighbor } v' \text{ of } v \text{ with } C \in L(v'), \\
& \text{let } V = V \cup \{v'\}, E = E \cup \{(v, v')\}, L(v') := \{C\} \text{ and } \\
& L(v, v') := \{r\} \text{ for } v' \text{ a new node.}
\\
\forall\text{-rule:} & \text{For } v, v' \in V, v' \text{ } r\text{-neighbor of } v, \\
& \forall_r C \in L(v) \text{ and } C \notin L(v'), \text{ let } L(v') := L(v') \cup \{C\}.
\\
\leq 1\text{-rule:} & \text{For a functional role } f \text{ and a } v \in V \text{ with two } \\
& f\text{-neighbors } v_1 \text{ and } v_2, \text{ execute } \text{merge}(v_1, v_2).
\\
T\text{-rule:} & \text{For a } v \in V \text{ with } C_T \notin L(v), \\
& \text{let } L(v) := L(v) \cup \{C_T\}.
\end{array}\]
Merging Nodes

we define 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 prune($v_o$).

where 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. $\mathcal{T}$?

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

$v_1$ blocks $v_2$, but cyclic model construction does not work (functionality violated)!
Tableau with Functional Roles

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

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

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

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

\[\begin{array}{c}
 v_0 \\
 f \\
 v_1 \\
 f \\
 v_2
\end{array}\]

\[\begin{align*}
 \mathcal{T} &= \{ A \sqsubseteq \exists f.A, \top \sqsubseteq 1 f^- \} \\
 C_{\mathcal{T}} &= (\neg A \sqcup \exists f.A) \sqcap 1 f^- \\
 L(v_0) &= \{ \exists f.A, C_{\mathcal{T}}, \neg A, \leq 1 f^- \} \\
 L(v_1) &= \{ A, C_{\mathcal{T}}, \exists f.A, \leq 1 f^- \} \\
 L(v_2) &= \{ A, C_{\mathcal{T}}, \exists f.A, \leq 1 f^- \}
\end{align*}\]
Example: Is $\exists f.A$ satisfiable w.r.t. $T$?

\[
\begin{align*}
T = \{ & A \sqsubseteq \exists f.A, \top \sqsubseteq 1 f^- \} \\
C_T = (\neg A \sqcup \exists f.A) \sqcap 1 f^- \\
L(v_0) = \{ & \exists f.A, C_T, \neg A, 1 f^- \} \\
L(v_1) = \{ & A, C_T, \exists f.A, 1 f^- \} \\
L(v_2) = \{ & A, C_T, \exists f.A, 1 f^- \}
\end{align*}
\]

$v_1$ 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.

\[
\begin{align*}
L(v_0) &= \{\exists f.A, C_T, \neg A, \leq 1 f^- \} \\
L(v_1) &= \{A, C_T, \exists f.A, \leq 1 f^- \} \\
L(v_2) &= \{A, C_T, \exists f.A, \leq 1 f^- \} \\
\end{align*}
\]

\(v_1\) 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.

\[
\begin{align*}
L(v_0) &= \{ \exists f. A, C_T, \neg A, \leq 1f^- \} \\
L(v_1) &= \{ A, C_T, \exists f. A, \leq 1f^- \} \\
L(v_2) &= \{ A, C_T, \exists f. A, \leq 1f^- \}
\end{align*}
\]

\[
v_1 \text{ blocks } v_2
\]
Unravelling

goal: we build an infinite model
How? Every blocked node is replaced by a subtree whose root is the corresponding blocking node.

\[ L(v_0) = \{ \exists f. A, C_T, \neg A, \leq 1 f^- \} \]
\[ L(v_1) = \{ A, C_T, \exists f. A, \leq 1 f^- \} \]
\[ L(v_2) = \{ A, C_T, \exists f. A, \leq 1 f^- \} \]

\( v_1 \) 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.

\[ L(v_0) = \{ \exists f.A, C_T, \neg A, \leq 1f^- \} \]

\[ L(v_1) = \{ A, C_T, \exists f.A, \leq 1f^- \} \]

\[ L(v_2) = \{ A, C_T, \exists f.A, \leq 1f^- \} \]

\( v_1 \) blocks \( v_2 \)
Blocking: Inverse and Functional Roles

Example: Is $\neg C \cap \exists f.\ D$ satisfiable w.r.t. $T$?

$$T = \{ D \subseteq C \cap \exists f. (\neg C) \cap \exists f.\ D , \top \subseteq 1f \}$$
Blocking: Inverse and Functional Roles

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

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

$$C_{\mathcal{T}} = (\neg D \sqcup (C \sqcap \exists f.(\neg C) \sqcap \exists f^{-}.D)) \sqcap 1f$$
Example: Is $\neg C \sqcap \exists f^- . D$ satisfiable w.r.t. $\mathcal{T}$?

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

$$C_T = (\neg D \sqcup (C \sqcap \exists f . (\neg C) \sqcap \exists f^- . D)) \sqcap \leq f$$

$$L(v_0) = \{ \neg C, \exists f^- . D, C_T, \ldots, \neg D, \leq f \}$$

$$L(v_1) = \{ D, C_T, \ldots, C, \exists f . (\neg C), \exists f^- . D, \leq f \}$$

$$L(v_2) = \{ D, C_T, \ldots, C, \exists f . (\neg C), \exists f^- . D, \leq f \}$$

$v_1$ blocks $v_2$ (same label)
Blocking: Inverse and Functional Roles

Example: Is \( \neg C \sqcap \exists f^- . D \) satisfiable w.r.t. \( T \)?

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

\[
C_T = (\neg D \sqcup (C \sqcap \exists f . (\neg C) \sqcap \exists f^- . D)) \sqsubseteq 1 f
\]

\[
L(v_0) = \{ \neg C, \exists f^- . D, C_T, \ldots, \neg D, \leq 1 f \}
\]

\[
L(v_1) = \{ D, C_T, \ldots, C, \exists f . (\neg C), \exists f^- . D, \leq 1 f \}
\]

\[
L(v'_1) = \{ D, C_T, \ldots, C, \exists f . (\neg C), \exists f^- . D, \leq 1 f \}
\]

\[
v_1 \text{ blocks } v_2 \text{ (same label) but}
\]

\[
L(v''_1) = \{ D, C_T, \ldots, C, \exists f . (\neg C), \exists f^- . D, \leq 1 f \}
\]

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. \( \mathcal{T} \)?

\[
\mathcal{T} = \{ D \sqsubseteq C \sqcap \exists f. (\neg C) \sqcap \exists f^-.D, \top \sqsubseteq \leq f \} \\
C_\mathcal{T} = (\neg D \sqcup (C \sqcap \exists f. (\neg C) \sqcap \exists f^-.D)) \sqcap \leq f
\]

\[
L(v_0) = \{ \neg C, \exists f^-.D, C_\mathcal{T}, \ldots, \neg D, \leq f \} \\
L(v_1) = \{ D, C_\mathcal{T}, \ldots, C, \exists f. (\neg C), \exists f^-.D, \leq f \} \\
L(v_2) = \{ D, C_\mathcal{T}, \ldots, C, \exists f. (\neg C), \exists f^-.D, \leq f \}
\]

\( v_1 \) cannot block \( v_2 \) pairwise
Pairwise Blocking: Inverses and Functional Roles

Example: Is \( \neg C \sqcap \exists f^- . D \) satisfiable w.r.t. \( T \)?

\[ T = \{ D \sqsubseteq C \sqcap \exists f . (\neg C) \sqcap \exists f^- . D, \top \sqsubseteq 1 f \} \]
\[ C_T = (\neg D \sqcup (C \sqcap \exists f . (\neg C) \sqcap \exists f^- . D)) \sqcap 1 f \]

\begin{align*}
L(v_0) &= \{ \neg C, \exists f^- . D, C_T, \ldots, \neg D, \leq 1 f \} \\
L(v_1) &= \{ D, C_T, \ldots, C, \exists f . (\neg C), \exists f^- . D, \leq 1 f \} \\
L(v_2) &= \{ D, C_T, \ldots, C, \exists f . (\neg C), \exists f^- . D, \leq 1 f \}
\end{align*}

\( v_1 \) cannot block \( v_2 \) pairwise

\begin{align*}
L(v_3) &= \{ \neg C \}
\end{align*}
Example: Is $\neg C \cap \exists f^- . D$ satisfiable w.r.t. $T$?

$$T = \{ D \subseteq C \cap \exists f . (\neg C) \cap \exists f^- . D, \top \subseteq 1 f \}$$

$$C_T = (\neg D \sqcup (C \cap \exists f . (\neg C) \cap \exists f^- . D)) \sqcup 1 f$$

\[
\begin{array}{c}
v_0 \\
\downarrow f^- \\
v_1 \\
\downarrow f^- \\
v_2 \\
\downarrow f \\
v_3
\end{array}
\]

$$L(v_0) = \{ \neg C, \exists f^- . D, C_T, \ldots, \neg D, \leq 1 f \}$$

$$L(v_1) = \{ D, C_T, \ldots, C, \exists f . (\neg C), \exists f^- . D, \leq 1 f \}$$

$$L(v_2) = \{ D, C_T, \ldots, C, \exists f . (\neg C), \exists f^- . D, \leq 1 f \}$$

$v_1$ cannot block $v_2$ pairwise

$$L(v_3) = \{ \neg C \}$$
Example: Is $\neg C \sqcap \exists f^-.D$ satisfiable w.r.t. $T$?

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

$$C_T = (\neg D \sqcup (C \sqcap \exists f.(-C) \sqcap \exists f^- .D)) \sqsubseteq 1f$$

$v_0$

$f^-$

$v_1$

$v_1$ cannot block $v_2$ pairwise

$L(v_0) = \{-C, \exists f^- .D, C_T, \ldots, \neg D, \leq 1f\}$

$L(v_1) = \{D, C_T, \ldots, C, \exists f.(-C), \exists f^- .D, \leq 1f\}$

$L(v_2) = \{D, C_T, \ldots, C, \exists f.(-C), \exists f^- .D, \leq 1f\}$

$v_3$ is merged into $v_1$

$L(v_3) = \{-C\}$

$L(v_1) = L(v_1) \cup L(v_3) \supseteq \{-C, C\}$

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 $\mathcal{ALCI}F$ 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