Logical Aspects of Artificial Intelligence
Tableaux for DLs

Stéphane Demri demri@lsv.fr

December 2nd, 2020
Plan of the lecture

- Tableaux calculus for checking $\mathcal{ALC}$ concept satisfiability.
- Tableaux calculus for checking $\mathcal{ALC}$ knowledge base consistency.
- Exercises session.
Recapitulation of the previous lecture(s)
\(\mathcal{ALC}\) in a nutshell

\[C ::= \top \mid \bot \mid A \mid \neg C \mid C \cap C \mid C \cup C \mid \exists r.C \mid \forall r.C\]

- Interpretation \(\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})\).

- TBox \(\mathcal{T} = \{C \sqsubseteq D, \ldots\}\).

- ABox \(\mathcal{A} = \{a : C, (b, b') : r, \ldots\}\).

- Knowledge base \(\mathcal{K} = (\mathcal{T}, \mathcal{A})\). (a.k.a. ontology)

- Decision problems include concept satisfiability, knowledge base consistency, and other problems for classification.
\( \top^\mathcal{I} \overset{\text{def}}{=} \Delta^\mathcal{I} \)

\( \bot^\mathcal{I} \overset{\text{def}}{=} \emptyset \)

\( (\neg C)^\mathcal{I} \overset{\text{def}}{=} \Delta^\mathcal{I} \setminus C^\mathcal{I} \)

\( (C_1 \sqcup C_2)^\mathcal{I} \overset{\text{def}}{=} C_1^\mathcal{I} \cup C_2^\mathcal{I} \)

\( (C_1 \sqcap C_2)^\mathcal{I} \overset{\text{def}}{=} C_1^\mathcal{I} \cap C_2^\mathcal{I} \)

\( (\exists r.C)^\mathcal{I} \overset{\text{def}}{=} \{ a \in \Delta^\mathcal{I} \mid r^\mathcal{I}(a) \cap C^\mathcal{I} \neq \emptyset \} \)

\( (\forall r.C)^\mathcal{I} \overset{\text{def}}{=} \{ a \in \Delta^\mathcal{I} \mid r^\mathcal{I}(a) \subseteq C^\mathcal{I} \} \)
A few properties about \textit{ALC}

- Concept satisfiability problem is PSPACE-complete.

- Knowledge base consistency problem is EXPTime-complete.

- \textit{ALC} has many well-known fragments and extensions, some of them to deal with
  - inverse roles,
  - number restrictions,
  - properties on the role interpretations,
  - inclusions between the composition of roles,
  - etc..

- Reduction of decision problems for DLs to first-order logic and modal logics.

- Filtration construction leading to an NEXPTime upper bound for the \textit{ALC} knowledge base consistency problem.
Tableaux Proof System for $\mathcal{ALC}$
Automated reasoning for non-classical logics

▶ Direct methods:
  ▶ Analytical calculi: tableaux, sequents ($\Gamma \vdash \Delta$), etc.
  ▶ Resolution. ($\frac{L \lor C_1 \quad -L' \lor C_2}{C_1 \sigma \lor C_2 \sigma}$ with $L\sigma = L'\sigma$)
  ▶ Automata-based decision procedures. ($\varphi$ sat. iff $L(A\varphi) \neq \emptyset$)

▶ Translation into
  ▶ other modal logics ($ML^+, PDL$, modal $\mu$-calculus, . . .)
  ▶ decidable fragments of first-order logic ($FO2$, $GF$, $CGF$, . . .)
  ▶ second-order monadic logics ($S2S$, $S\omega S$, . . .)
Tableaux-based proof systems

- Analytical method: the rules of the calculi perform a syntactic decomposition of the concepts (formulae, etc.).

- Method developed initially by R. Smullyan for first-order logic (circa 1968).


- Close relationships with sequent-style proof systems.

- Modular approach as new conditions or new ingredients may correspond to the addition of new rules.

- Labels are sometimes used in such proof systems.
  - Labels are interpreted as entities of the domain under construction.
  - Expressions external to the original logical language.
  - Labels can be also used as control data structures.
Methodology

- Satisfiability of $C$ by checking consistency of $\mathcal{A} = \{ a : C \}$.

- First part dedicated to a tableaux calculus for checking the consistency status of ABoxes $\mathcal{A}$.

- Second part presents an extension for checking the consistency status of a knowledge base $(\mathcal{T}, \mathcal{A})$.

- This leads to a decision procedure (terminating) for knowledge base consistency in which a few optimisations leads to EXPTIME (not presented herein).

- The proof system works by rewriting ABoxes. ($\mathcal{A} \rightarrow^* \mathcal{A}'$)

- Our presentation follows the usual way to present tableaux-style proof systems for description logics.
Negation normal form

- C is in negation normal form (NNF) \[ \iff \] the concept negation \( \neg \) occurs only in front of concept names.

- Every concept has an equivalent concept in NNF:

\[
\neg (C \sqcup D) \equiv \neg C \cap \neg D \quad \neg (C \cap D) \equiv \neg C \sqcup \neg D
\]

\[
\neg \exists r. C \equiv \forall r. \neg C \quad \neg \forall r. C \equiv \exists r. \neg C \quad \neg \neg C \equiv C
\]

- Transforming a concept into an equivalent concept in NNF takes polynomial time (only).

\( \neg \top \equiv \bot \quad \neg \bot \equiv \top \)

- NNFs are not a must but this simplifies forthcoming developments.

(Each connective has its dual)
Principles of the tableaux-style proof systems

- The calculus is made of rewriting rules that transform an ABox $\mathcal{A}$ into another ABox $\mathcal{A}'$ nondeterministically. (ex: $\sqcap$-rule)

- ABoxes $\mathcal{A}$ are understood as partial description of interpretations. (individual names play the role of labels)

- To guarantee termination, provisos are added to the application of the rules. (see the blocking technique)

- ABoxes with no contradiction and for which no rule application adds value correspond to interpretations.
Example: the \( \cap \)-rule

\( \cap \)-rule: If \( a : C \cap D \in \mathcal{A} \) and \( \{a : C, a : D\} \not\subseteq \mathcal{A} \) then

\[
\mathcal{A} \rightarrow \mathcal{A} \cup \{a : C, a : D\}
\]

- Applying the rule can be viewed as repairing a defect.
- Condition \( \{a : C, a : D\} \not\subseteq \mathcal{A} \) rules out void rule applications.
- The other rules are designed on the same pattern.

(guess the \( \cup \)-rule)
Expansion rules for \( \mathcal{ALC} \) ABox consistency

\( \sqcap \)-rule: If \( a : C \ sqcap D \in A \) and \( \{ a : C, a : D \} \not\subseteq A \) then

\[
A \rightarrow A \cup \{ a : C, a : D \}
\]

\( \sqcup \)-rule: If \( a : C \ sqcup D \in A \) and \( \{ a : C, a : D \} \cap A = \emptyset \) then

\[
A \rightarrow A \cup \{ a : E \} \quad \text{for some } E \in \{ C, D \}
\]

\( \exists \)-rule: If \( a : \exists r.C \in A \) and there is no \( b \) such that
\[
\{(a, b) : r, b : C\} \subseteq A
\]
then

\[
A \rightarrow A \cup \{(a, c) : r, c : C\} \quad \text{where } c \text{ is fresh}
\]

\( \forall \)-rule: If \( \{(a, b) : r, a : \forall r.C\} \subseteq A \) and \( b : C \not\in A \), then

\[
A \rightarrow A \cup \{ b : C \}
\]
Complete and clash-free ABox

- An ABox $\mathcal{A}$ contains a **clash** if $\{ a : A, a : \neg A \} \subseteq \mathcal{A}$ or $a : \bot \in \mathcal{A}$.

⚠️ Notion of clash to be extended if the concepts are not in NNF.

- An ABox $\mathcal{A}$ is **clash-free** if it contains no clash.

- An ABox $\mathcal{A}$ is **complete** if it contains a clash or if no rule is applicable (saturation).

**Objective:** to show that $\mathcal{A}$ is consistent iff $\mathcal{A} \overset{*}{\rightarrow} \mathcal{A}'$ for some complete and clash-free ABox $\mathcal{A}'$.

- The only nondeterministic rule is the $\sqcup$-rule. (is it true?)
Example

\[ A = \{(a, b) : s, (a, c) : r\} \cup \{a : A_1 \cap \exists s.A_5, a : \forall s. (\neg A_5 \sqcup \neg A_2), b : A_2, c : A_3 \cap \exists s.A_4\} \]

\[ A \xrightarrow{\ast} A \cup \{a : A_1, a : \exists s.A_5, a_{\text{new}} : A_5, (a, a_{\text{new}}) : s, b : \neg A_5 \sqcup \neg A_2, a_{\text{new}} : (\neg A_5 \sqcup \neg A_2), b : \neg A_5, a_{\text{new}} : \neg A_2, c : A_3, c : \exists s.A_4, c_{\text{new}} : A_4, (c, c_{\text{new}}) : s\} \]

(is it complete?)
Why “Tableaux”?

\[(a, b) : s\]
\[(a, c) : r\]
- \[a : \exists s \cdot A_s A_5\]
- \[c : A_3 \cup \exists s A_4\]
- \[a : \exists s A_5\]
- \[a : \exists s A_5\]
- \[a_{new} : A_5\]
- \[(a, a_{new}) : s\]
- \[b : \exists s \cup \exists A_2\]
- \[a_{new} : \exists A_5 \cup \exists A_2\]
- \[b : \exists A_2\]
- \[a_{new} : \exists A_5\]
- \[c : A_3\]
- \[c : \exists s A_4\]
- \[c_{new} : A_4\]
- \[(c, c_{new}) : s\]
Termination

- **The $\exists$-weight** of $C$ is the number of its subconcepts of the form $\exists r.D$.

$$w_\exists(C) \overset{\text{def}}{=} \text{card}\{\exists r.D \mid \exists r.D \in \text{sub}(C)\}$$

⚠️ The definition assumes that $C$ is in NNF.

- $w_\exists(\mathcal{A}) \overset{\text{def}}{=} \sum_{a:C \in \mathcal{A}} w_\exists(C)$.

- The $\forall \exists$-depth of $C$, written $d_{\forall \exists}(C)$, is the maximal number of imbrications of $\exists r.$ and $\forall s.$ in $C$.

- $d_{\forall \exists}(\exists r.r \top \sqcup \forall r.\exists s.\mathcal{A}) = 2$

- $d_{\forall \exists}(\mathcal{A}) = \max\{d_{\forall \exists}(C) \mid a : C \in \mathcal{A}\}$. 
Decorating individual names

Let \( \mathcal{A} \) be an ABox with \( W = w_\exists(\mathcal{A}) \), \( D = d_\forall(\mathcal{A}) \) and \( N \) is the number of distinct individual names in \( \mathcal{A} \).

Let \( \mathcal{A}^0 \) be the variant of \( \mathcal{A} \) where \( a : C \) is replaced by \( a^0 : C \).

\( \cap \)-rule: If \( a^i : C \cap D \in \mathcal{A} \) and \( \{ a^i : C, a^i : D \} \not\subseteq \mathcal{A} \) then
\[
\mathcal{A} \rightarrow \mathcal{A} \cup \{ a^i : C, a^i : D \}
\]

\( \sqcup \)-rule: If \( a^i : C \sqcup D \in \mathcal{A} \) and \( \{ a^i : C, a^i : D \} \cap \mathcal{A} = \emptyset \) then
\[
\mathcal{A} \rightarrow \mathcal{A} \cup \{ a^i : E \} \quad \text{for some } E \in \{ C, D \}
\]

\( \exists \)-rule: If \( a^i : \exists r . C \in \mathcal{A} \) and there is no \( b^i \) such that \( \{(a^i, b^i) : r, b^i : C\} \subseteq \mathcal{A} \) then
\[
\mathcal{A} \rightarrow \mathcal{A} \cup \{(a^i, c^{i+1}) : r, c^{i+1} : C\} \quad \text{where } c \text{ is fresh}
\]

\( \forall \)-rule: If \( \{(a^i, b^i) : r, a^i : \forall r . C\} \subseteq \mathcal{A} \) and \( b^i : C \notin \mathcal{A} \), then
\[
\mathcal{A} \rightarrow \mathcal{A} \cup \{ b^i : C \}
Quantities about $A^0 \rightarrow A'$

- If $a^i : C \in A'$, then $i + d_{\exists}(C) \leq D$.
  Trees from individual names labelled by zero have depth at most $D$.

- $a^i : C \in A'$ implies
  \[ \text{card}(\{(a^i, b^j) \mid (a^i, b^j) : r \in A'\}) \leq N + W \]
  (necessarily $j = i + 1$)

  The maximum branching degree of nodes in the trees is at most $N + W$.

- $a^i : C \in A'$ implies $C \in \text{sub}(A)$.

- The length of the derivation $A^0 \rightarrow A'$ is at most
  \[ N \times (N + W)^{D+1} \times \text{card}({\text{sub}(A)}) \]
  (why?)
The auxiliary function \( \text{exp} \)

- **Expansion function** \( \text{exp}(\mathcal{A}, R, X) \) taking as arguments
  - an ABox \( \mathcal{A} \),
  - an expansion rule \( R \),
  - a subset \( X \) of \( \mathcal{A} \) (with one or two elements) allowing the application of \( R \)

- ... and returning the set of ABoxes obtained from \( \mathcal{A} \) by applying the rule \( R \) with main assertions in \( X \).

- \( \text{exp}(\{a : E, a : C \sqcup D\}, \sqcup\text{-rule}, a : C \sqcup D) \) is equal to

\[
\{\{a : E, a : C \sqcup D, a : C\}, \{a : E, a : C \sqcup D, a : D\}\}
\]
Main algorithm

We shall show that \( \mathcal{A} \) is consistent iff \( \mathcal{A} \xrightarrow{\ast} \mathcal{A}' \) for some complete and clash-free ABox \( \mathcal{A}' \).

Existence of \( \mathcal{A}' \) amounts to explore a finite tree of bounded depth and bounded degree.

1: procedure EXPAND(\( \mathcal{A} \))
2: if \( \mathcal{A} \) has a clash then return \( \emptyset \)
3: end if
4: if \( \mathcal{A} \) is clash-free and complete then return \( \mathcal{A} \)
5: end if
6: for applicable \( R, X \) on \( \mathcal{A} \) and \( \mathcal{A}' \in \exp(\mathcal{A}, R, X) \) do
7: if EXPAND(\( \mathcal{A}' \)) \( \neq \emptyset \) then return EXPAND(\( \mathcal{A}' \))
8: end if
9: end for
10: return \( \emptyset \)
11: end procedure
procedure EXPAND($\mathcal{A}$)

2: if $\mathcal{A}$ has a clash then return $\emptyset$

3: end if

4: if $\mathcal{A}$ is clash-free and complete then return $\mathcal{A}$

5: end if

6: for applicable $R, X$ on $\mathcal{A}$ and $\mathcal{A}' \in \exp(\mathcal{A}, R, X)$ do

7: if $\text{EXPAND}(\mathcal{A}') \neq \emptyset$ then return $\text{EXPAND}(\mathcal{A}')$

8: end if

9: end for

10: return $\emptyset$

11: end procedure
Root individuals and tree individuals

- **Tree individuals** are generated by application of the ∃-rule.
- If \((a, b) : r\) is added by application of the ∃-rule, \(b\) is an *r-successor* of \(a\).
- Root individuals have no predecessors or ancestors.
Soundness

Let $\mathcal{A}$ be a finite ABox with at least one concept assertion, complete, clash-free and all the concepts in NNF. Then, $\mathcal{A}$ is consistent.

For each individual name $a$ occurring in $\mathcal{A}$, we write $\text{con}_\mathcal{A}(a)$ to denote the set $\{C \mid a : C \in \mathcal{A}\}$.

Let us define $\mathcal{I} \overset{\text{def}}{=} (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ as follows.

- $\Delta^\mathcal{I} \overset{\text{def}}{=} \{a \mid a : C \in \mathcal{A}\}$.
- $a^\mathcal{I} \overset{\text{def}}{=} a$ for all individual names $a$ in $\mathcal{A}$.
- $A^\mathcal{I} \overset{\text{def}}{=} \{a \mid A \in \text{con}_\mathcal{A}(a)\}$ for all concept names $A \in \text{sub}(\mathcal{A})$.
- $r^\mathcal{I} \overset{\text{def}}{=} \{(a, b) \mid (a, b) : r \in \mathcal{A}\}$.

Let us show that for all $a : C \in \mathcal{A}$, we have $a^\mathcal{I} \in C^\mathcal{I}$. 
Proof by structural induction

- The base case with concept assertions $a : A$ is immediate by definition of $A^\mathcal{I}$.

- The base case with concept assertions $a : \neg A$ is immediate by definition of $A^\mathcal{I}$ as $A$ is clash-free.

- Case $a : C \sqcup D$ in the induction step.
  - As $A$ is complete, $a : C \in A$ or $a : D \in A$.
  - W.l.o.g., suppose $a : C \in A$. By (IH), $a^\mathcal{I} \in C^\mathcal{I}$.
  - By definition of $\cdot^\mathcal{I}$, we conclude $a^\mathcal{I} \in (C \sqcup D)^\mathcal{I}$.

- Case $a : \exists r.C$ in the induction step.
  - As $A$ is complete, $\{(a, b) : r, b : C\} \subseteq A$ for some $b$.
  - By definition of $r^\mathcal{I}$, $(a, b) \in r^\mathcal{I}$.
  - By (IH), $b^\mathcal{I} \in C^\mathcal{I}$.
  - By definition of $\cdot^\mathcal{I}$, we conclude $a^\mathcal{I} \in (\exists r.C)^\mathcal{I}$.
Concluding the soundness

- The cases in the induction step for $\sqcap$-concept assertions and $\forall$-concept assertions are similar.

- If $\text{expand}(\mathcal{A}) \neq \emptyset$, then $\mathcal{A}$ is consistent.

- Indeed, $\text{expand}(\mathcal{A}) \neq \emptyset$ if there is some $\mathcal{A}'$ with $\mathcal{A} \subseteq \mathcal{A}'$ such that $\mathcal{A}'$ is complete and clash-free.

- Consistency of $\mathcal{A}'$ leads to the consistency of $\mathcal{A}$. 
Moving towards completeness

- If $A$ is consistent, then $A \rightarrow^* A'$ for some complete and clash-free ABox $A'$.

- Let $\mathcal{I} \overset{\text{def}}{=} (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ be such that $\mathcal{I} \models A$.

- If $A$ is complete, we are done. Otherwise, at least one rule is applicable to $A$ preserving consistency.

- Otherwise, if $A$ is not complete, we show that there is $A'$ such that $A \rightarrow A'$ and $A'$ is consistent.

- As the length of a derivation from $A$ is bounded by an exponential in the size of $A$, there is $A'$ such that $A \rightarrow^* A'$ and $A'$ is complete, clash-free (and consistent).

- It remains to prove that non-completeness implies the existence of one expansion preserving consistency.
Single steps in the completeness proof

- If the $\sqcup$-rule is applicable on $a : C \sqcup D$, then there is $E \in \{C, D\}$ such that $I \models A \cup \{a : E\}$.

- $A \rightarrow A \cup \{a : E\}$ and $I \models A \cup \{a : E\}$.

- If the $\exists$-rule is applicable on $a : \exists r.C$, then we use the fact that $a^I \in (\exists r.C)^I$.

- There is $a \in \Delta^I$ such that $a \in C^I$ and $(a^I, a) \in r^I$.

- Let $I'$ be equal to $I$ except that $I'(c) = a$ for some fresh $c$.

- Then, $A \rightarrow A \cup \{c : C, (a, c) : r\}$ and $I' \models A \cup \{c : C, (a, c) : r\}$.
Decision procedure of ABox consistency

- $\mathcal{A}$ is consistent iff $\mathcal{A} \overset{*}{\rightarrow} \mathcal{A}'$ for some complete and clash-free ABox $\mathcal{A}'$.

- Derivations $\mathcal{A} \overset{*}{\rightarrow} \mathcal{A}'$ have length bounded by an exponential in size($\mathcal{A}$).

- Existence of $\mathcal{A}'$ amounts to explore a tree of bounded depth and bounded degree.
Adding a TBox – First properties

- $\mathcal{I} \models C \subseteq D$ iff $\mathcal{I} \models T \subseteq \neg C \cup D$.

- $\mathcal{I} \models C \equiv D$ iff $\mathcal{I} \models T \subseteq (\neg C \cup D) \cap (\neg D \cup C)$.

- In the sequel, GCIs are of the form $T \subseteq E$ with $E$ in NNF.

$\sqsubseteq$-rule: If $a : C \in \mathcal{A}$, $T \subseteq D \in \mathcal{T}$ and $a : D \notin \mathcal{A}$, then

$$\mathcal{A} \rightarrow \mathcal{A} \cup \{a : D\}$$

- The termination argument for ABox consistency does not work anymore. (Why?)
Termination with the blocking technique

Given $\mathcal{A} \rightarrow^* \mathcal{A}'$, $a$ is an **ancestor** of $b$ in $\mathcal{A}'$ iff

$$\{(a_1, a_2) : r_1, \ldots, (a_k, a_{k+1}) : r_k\} \subseteq \mathcal{A}'$$

with $a_1 = a$, $a_{k+1} = b$ and $b$ is a tree individual.

The notion of ancestor assumes that one can distinguish the **root individuals** (individual names from $\mathcal{A}$) from the **tree individuals** (those introduced by applying the $\exists$-rule).

Termination can be regained thanks to the blocking technique.

An individual name $b$ in $\mathcal{A}'$ is **blocked by** $a$ if

- $a$ is an ancestor of $b$,
- $\text{con}_{\mathcal{A}'}(b) \subseteq \text{con}_{\mathcal{A}'}(a)$. 
$b$ blocked by $a$

$A_0 \rightarrow A_2 \rightarrow \ldots \rightarrow A_k = \{ \}

\{D_1, \ldots, D_m\} \leq \{c_1, \ldots, c_m\}$
Expansion rules with blocking

\(\sqcap\)-rule: If \(a : C \cap D \in A\), \(a\) is not blocked and \(\{a : C, a : D\} \not\subseteq A\) then \(A \rightarrow A \cup \{a : C, a : D\}\).

\(\sqcup\)-rule: If \(a : C \cup D \in A\), \(a\) is not blocked and \(\{a : C, a : D\} \cap A = \emptyset\) then \(A \rightarrow A \cup \{a : E\}\) for some \(E \in \{C, D\}\).

\(\exists\)-rule: If \(a : \exists r.C \in A\), \(a\) is not blocked and there is no \(b\) such that \(\{(a, b) : r, b : C\} \subseteq A\) then

\[A \rightarrow A \cup \{(a, c) : r, c : C\}\] where \(c\) is fresh

\(\forall\)-rule: If \(\{(a, b) : r, a : \forall r.C\} \subseteq A\), \(a\) is not blocked and \(b : C \not\in A\), then \(A \rightarrow A \cup \{b : C\}\).

\(\sqsubseteq\)-rule: If \(a : C \in A\), \(\top \sqsubseteq D \in T\), \(a\) is not blocked and \(a : D \not\in A\), then \(A \rightarrow A \cup \{a : D\}\).
Termination

- $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ with concepts in NNF, and GCIs of the form $\top \sqsubseteq D$.

- $N$: number of root individuals in $\mathcal{A}$, $M = \text{card}(\text{sub}(\mathcal{K}))$, $W = w_\exists(\mathcal{K})$.

- $\mathcal{A} \rightarrow^* \mathcal{A}'$ and $a : C \in \mathcal{A}'$ imply $\text{card}(\{(a, b) \mid (a, b) : r \in \mathcal{A}'\}) \leq N + W$.

- $\mathcal{A} \rightarrow^* \mathcal{A}'$ and $a : C \in \mathcal{A}'$ imply $C \in \text{sub}(\mathcal{K})$.

- $\{(a_1, a_2) : r_1, \ldots, (a_k, a_{k+1}) : r_k\} \subseteq \mathcal{A}'$ and $a_2$ is a tree individual imply $k \leq 2^M$.

- The length of the derivation $\mathcal{A} \rightarrow^* \mathcal{A}'$ is at most

$$N \times (N + W)^{(2^M + 1)} \times M$$
Soundness

- $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ with concepts in NNF, and GCI s of the form $T \sqsubseteq D$.

- $\mathcal{A} \xrightarrow{*} \mathcal{A}'$ with $\mathcal{A}'$ complete and clash-free.

- We construct $\mathcal{A}''$ as the ABox made of the following assertions

$$\{a : C \mid a : C \in \mathcal{A}', \ a \text{ is not blocked}\} \cup$$

$$\{(a, b) : r \mid (a, b) : r \in \mathcal{A}', \ b \text{ is not blocked}\} \cup$$

$$\{(a, b') : r \mid (a, b) : r \in \mathcal{A}', \ a \text{ is not blocked and } b \text{ is blocked by } b'\}$$
Construction of $A''$
Properties of $\mathcal{A}''$

- $\mathcal{A} \subseteq \mathcal{A}''$ as root individual cannot be blocked and $\mathcal{A} \subseteq \mathcal{A}'$.

- None of the individual names occurring in $\mathcal{A}''$ is blocked.

- For all $a$ in $\mathcal{A}''$, we have $\text{con}_{\mathcal{A}''}(a) = \text{con}_{\mathcal{A}'}(a)$. 

  (left as an exercise)

- $\mathcal{A}''$ is complete and clash-free.
Proof: $\mathcal{A}''$ is complete

$(\star)$ $\text{con}_{\mathcal{A}''}(a) = \text{con}_{\mathcal{A}'}(a)$ for all $a \in A''$

- Suppose $a : C \cap D \in \mathcal{A}''$.
  By $(\star)$, $a : C \cap D \in \mathcal{A}'$.
  As $\mathcal{A}'$ is complete, $\{a : C, a : D\} \subseteq \mathcal{A}'$.
  By $(\star)$, $\{a : C, a : D\} \subseteq \mathcal{A}''$.

- Suppose that $a : C \in \mathcal{A}''$ and $\top \subseteq D \in \mathcal{T}$.
  By $(\star)$, $a : C \in \mathcal{A}'$.
  As $\mathcal{A}'$ is complete, $a : D \in \mathcal{A}'$.
  By $(\star)$, $a : D \in \mathcal{A}''$. 
Case with the $\exists$-rule

- Suppose that $a : \exists r. C \in A''$.
  By $(\star)$, $a : \exists r. C \in A'$ and $a$ not blocked.
  By completeness of $A'$, there is $b$ such that
  $\{(a, b) : r, b : C\} \subseteq A'$.

- If $b$ is not blocked, then $\{(a, b) : r, b : C\} \subseteq A''$.

- As $a$ is not blocked, if $b$ is blocked, then $b$ is blocked by $b'$ in $A'$ and $b'$ is not blocked.

- By definition of $A''$, $(a, b') : r \in A''$.

- $\text{con}_{A'}(b) \subseteq \text{con}_{A'}(b')$ (blocking). By $(\star)$,
  $$C \in \text{con}_{A'}(b) \subseteq \text{con}_{A'}(b') = \text{con}_{A''}(b')$$
  So, $b' : C \in A''$.

- Case with the $\forall$-rule left as an exercise.
More about the soundness proof

- $\mathcal{A} \rightarrow^* \mathcal{A}'$ with $\mathcal{A}'$ complete and clash-free and $\mathcal{A}''$ computed as above.

- Let us define $\mathcal{I} \overset{\text{def}}{=} (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ from $\mathcal{A}''$ as follows.

  - $\Delta^\mathcal{I} \overset{\text{def}}{=} \{ a \mid a : C \in \mathcal{A}'' \}$.
  
  - $a^\mathcal{I} \overset{\text{def}}{=} a$ for all individual names $a$ in $\mathcal{A}''$.
  
  - $A^\mathcal{I} \overset{\text{def}}{=} \{ a \mid A \in \text{con}_A''(a) \}$ for all concept names $A \in \text{sub}(\mathcal{A}'')$.
  
  - $r^\mathcal{I} \overset{\text{def}}{=} \{ (a, b) \mid (a, b) : r \in \mathcal{A}'' \}$.

- One can show that for all $a : C \in \mathcal{A}''$, we have $a^\mathcal{I} \in C^\mathcal{I}$.

  (left as an exercise.)
The final step about soundness

- It remains to check that $\mathcal{I} \models (\mathcal{T}, \mathcal{A})$.

- One can show that for all $a : C \in \mathcal{A}''$, we have $a^\mathcal{I} \in C^\mathcal{I}$.

- Consequently, $\mathcal{I} \models \mathcal{A}$ as $\mathcal{A} \subseteq \mathcal{A}''$.

- Moreover, $\mathcal{I} \models \top \subseteq C$ for all $\top \subseteq C \in \mathcal{T}$.

- $a \in \Delta^\mathcal{I}$
  $\rightarrow a : C \in \mathcal{A}''$ (\mathcal{A}'' is complete)
  $\rightarrow a \in C^\mathcal{I}$ (see above)
Completeness (bis)

- If $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is consistent, then $\mathcal{A} \overset{*}{\rightarrow} \mathcal{A}'$ for some complete and clash-free ABox $\mathcal{A}'$.

- Let $\mathcal{I} \overset{\text{def}}{=} (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be such that $\mathcal{I} \models (\mathcal{T}, \mathcal{A})$.

- If $\mathcal{A}$ is complete, we are done. Otherwise, at least one rule is application to $\mathcal{A}$ preserving consistency.

- Otherwise ($\mathcal{A}$ is not complete), we show there is $\mathcal{A}'$ such that $\mathcal{A} \rightarrow \mathcal{A}'$ and $\mathcal{A}'$ is consistent.

- As the length of a derivation from $\mathcal{A}$ is bounded by a double-exponential in the size of $\mathcal{A}$, there is $\mathcal{A}'$ such that $\mathcal{A} \overset{*}{\rightarrow} \mathcal{A}'$ and $\mathcal{A}'$ is complete, clash-free (and consistent).

- One can prove that non-completeness implies the existence of one expansion preserving consistency.
Complexity issues

- *ALC* concept satisfiability in PSPACE, knowledge base consistency in EXPTIME.

- The algorithm for ABox consistency runs in exponential space:
  - Because of the nondeterministic $\sqcap$-rule, exponentially many ABoxes may be generated.
  - Complete ABoxes may be exponentially large.

- PSPACE bound for ABox consistency can be regained by exploring the tree-like interpretations in a depth-first manner having only one path at a time.
Tableaux for $\mathcal{ALCI}$ ($\mathcal{ALC} + \text{inverse}$)

\[(a, b) : r, b : \forall r^{-}.D, a : \forall r.C \models_{\mathcal{ALCI}} \{b : C, a : D\}\]

- $b$ is an $r$-neighbour of $a$ if $(a, b) : r$ or $(b, a) : r^{-}$.
- $b$ is an $r^{-}$-neighbour of $a$ if $(a, b) : r^{-}$ or $(b, a) : r$.
- Below, $R$ is either some $r$ or some $r^{-}$.

$\exists$-rule: If $a : \exists R.C \in \mathcal{A}$ and there is no $b$ such that $b : C \in \mathcal{A}$ and $b$ is an $R$-neighbour of $a$ then

\[
\mathcal{A} \rightarrow \mathcal{A} \cup \{(a, c) : R, c : C\} \quad \text{where } c \text{ is fresh}
\]

$\forall$-rule: If $a : \forall R.C \in \mathcal{A}$ and $b$ is an $R$-neighbour of $a$, then

\[
\mathcal{A} \rightarrow \mathcal{A} \cup \{b : C\}
\]
Equality blocking

- We need to strengthen the blocking (inclusion of sets of concepts is not anymore sufficient).

- An individual name $b$ in $\mathcal{A}'$ is **blocked by** $a$ if
  - $a$ is an ancestor of $b$,
  - $\text{con}_{\mathcal{A}'}(b) = \text{con}_{\mathcal{A}'}(a)$ (equality blocking).

- Proofs for $\mathcal{ALC}$ can be adapted to $\mathcal{ALCI}$ to show
  - **Termination**: number of steps for $\mathcal{A} \rightarrow^* \mathcal{A}'$ is bounded.
  
  - **Soundness**: $\mathcal{A} \rightarrow^* \mathcal{A}'$ with complete and clash-free $\mathcal{A}'$ implies $(\mathcal{T}, \mathcal{A})$ is consistent.

  - **Completeness**: $(\mathcal{T}, \mathcal{A})$ is consistent implies $\mathcal{A} \rightarrow^* \mathcal{A}'$ for some complete and clash-free $\mathcal{A}'$. 
Recapitulation:
Tableaux for $\mathcal{ALC}$ knowledge base consistency

- Tableaux-based algorithm to decide $\mathcal{ALC}$ knowledge base consistency.

- All other standard decision problems can be handled too.

- Termination is guaranteed thanks to the blocking technique.

- In the worst-case, exponential space is used but optimisations exist to meet the optimal upper bound $\text{EXP TIME}$. 
Today lecture: tableaux for DLs.

- Rules for checking concept satisfiability.
- Rules for checking knowledge base consistency.
- Termination, soundness, completeness.

Next week lecture: Decidability/complexity status for DLs.