Knowledge Representation and Reasoning

Exercise Sheet 7

Albert-Ludwigs-Universität Freiburg

Bernhard Nebel, Gregor Behnke, Thorsten Engesser, Rolf-David Bergdoll, Leonardo Mieschendahl, Johannes Herrmann

10th December 2021

Exercise 7.1 – Structural Subsumption Algorithm

Exercise 7.1 (Structural Subsumption Algorithm, 4)

Given the $\mathcal{FL}^-$ concepts C' and D' and terminology $\mathcal{T}$ you are asked to use the structural subsumption algorithm from the lecture to prove or disprove $\mathcal{T} \models C' \sqsubseteq D'$. You will need to apply normalization and unfolding as preprocessing steps.

- $\mathcal{T} = \{A \sqsubseteq \forall r_1.(\exists r_2 \sqcap C), B \sqsubseteq \forall r_1.(\exists r_3 \sqcap \forall r_2.D), D \doteq \exists r_2 \sqcap \forall r_2.C\}$
- $C' \doteq A \sqcap B$
- $D' \doteq \forall r_1.(\forall r_2.\exists r_2) \sqcap A$

Exercise 7.1

Preprocessing: Normalize, unfold and reorder

$$\mathcal{T} = \{A \sqsubseteq \forall r_1.(\exists r_2 \sqcap C), B \sqsubseteq \forall r_1.(\exists r_3 \sqcap \forall r_2.D), D \doteq \exists r_2 \sqcap \forall r_2.C\}$$

$$\widetilde{\mathcal{T}} = \{A \doteq A^* \sqcap \forall r_1.(\exists r_2 \sqcap C), B \doteq B^* \sqcap \forall r_1.(\exists r_3 \sqcap \forall r_2.D), D \doteq \exists r_2 \sqcap \forall r_2.C\}$$

$$\widehat{\mathcal{T}} = \{A \doteq A^* \sqcap \forall r_1.(\exists r_2 \sqcap C), B \doteq B^* \sqcap \forall r_1.(\exists r_3 \sqcap \forall r_2.(\exists r_2 \sqcap \forall r_2.C)),$$
$$D \doteq \exists r_2 \sqcap \forall r_2.C\}$$

$$\begin{aligned} C' &\doteq A \sqcap B \\ &\equiv A^* \sqcap \forall r_1.(\exists r_2 \sqcap C) \sqcap B^* \sqcap \forall r_1.(\exists r_3 \sqcap \forall r_2.(\exists r_2 \sqcap \forall r_2.C)) \\ &\equiv A^* \sqcap B^* \sqcap \forall r_1.(C \sqcap \exists r_2 \sqcap \exists r_3 \sqcap \forall r_2.(\exists r_2 \sqcap \forall r_2.C)) \end{aligned}$$

$$\begin{aligned} D' &\doteq \forall r_1.(\forall r_2.\exists r_2) \sqcap A \\ &\equiv \forall r_1.(\forall r_2.\exists r_2) \sqcap A^* \sqcap \forall r_1.(\exists r_2 \sqcap C) \\ &\equiv A^* \sqcap \forall r_1.(C \sqcap \exists r_2 \sqcap \forall r_2.\exists r_2) \end{aligned}$$

Exercise 7.1

Apply the subsumption algorithm

Go through D' and check whether every conjunct also occurs in C' (apply recursively within quantified terms).

$$C' \equiv A^* \sqcap B^* \sqcap \forall r_1.(C \sqcap \exists r_2 \sqcap \exists r_3 \sqcap \forall r_2.(\exists r_2 \sqcap \forall r_2.C))$$

$$D' \equiv A^* \sqcap \forall r_1.(C \sqcap \exists r_2 \sqcap \forall r_2.\exists r_2)$$

All terms in D' are accounted for.
$\rightsquigarrow C' \sqsubseteq D'$

Exercise 7.2(a) – Tableau Algorithm

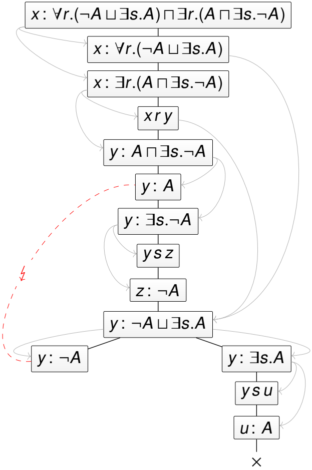
Exercise 7.2 (Tableau Algorithm, 2 + 3 + 3)

In this exercise you are asked to apply the tableau algorithm for description logics and to use it to answer questions about TBoxes and ABoxes.[1]

(a) Use the tableau algorithm to show that the $\mathcal{ALC}$ concept $C \doteq \forall r.(\neg A \sqcup \exists s.A) \sqcap \exists r.(A \sqcap \exists s.\neg A)$ is satisfiable. Extract a model of C from your tableau.

Exercise 7.2(a)

Constraint tableau for $C \doteq \forall r.(\neg A \sqcup \exists s.A) \sqcap \exists r.(A \sqcap \exists s.\neg A)$



Model $\mathcal{I}$ with $\mathcal{D} = \{x, y, z, u\}$:

$C^{\mathcal{I}} = \{x\}$, $A^{\mathcal{I}} = \{y, u\}$, $r^{\mathcal{I}} = \{(x, y)\}$, $s^{\mathcal{I}} = \{(y, z), (y, u)\}$

Exercise 7.2(b) – Tableau Algorithm

(b) The logic $\mathcal{ALCQ}$ extends $\mathcal{ALC}$ by cardinality restrictions, i.e. $(\leq nr.C)$ and $(\geq nr.C)$. Given two $\mathcal{ALCQ}$ concepts A and B, use tableau to prove that $A \sqsubseteq B$.

- $A \doteq \exists r.(\leq 2r.C) \sqcap \forall r.C$
- $B \doteq \forall r.(C \sqcup D) \sqcap \exists r.(\leq 3r.C)$

Hint: To transform a concept description that contains (qualified) number restrictions to negation normal form, you may need to make use of the following equivalences: $\neg(\geq n+1r.C) \equiv (\leq nr.C), \neg(\geq 0r.C) \equiv \bot, \neg(\leq nr.C) \equiv (\geq n+1r.C)$. The rules for expanding (qualified) number restrictions are:

- $S \rightarrow_{\geq} \{x\, r\, y_i \mid 1 \leq i \leq n\} \cup \{y_i \colon C \mid 1 \leq i \leq n\} \cup \{y_i \neq y_j \mid 1 \leq i < j \leq n\} \cup S$
  if $(x\colon \geq nr.C) \in S$, $y_i, \ldots, y_n$ are fresh variables, and there are no $z_1, \ldots, z_n$ s.t. $(x\, r\, z_i) \in S$, $(z_i \colon C) \in S$ and $(z_i \neq z_j) \in S$ for all z_i, z_j with $1 \leq i < j \leq n$.
- $S \rightarrow_{\leq} [y_i/y_j]S$ (replace all occurrences of y_i by y_j)
  if $(x\colon \leq nr.C) \in S$, $y_i \neq y_j \notin S$ and there are $y_1, \ldots, y_{n+1}$ s.t $(x\, r\, y_k) \in S$, $(y_k \colon C) \in S$ for all $1 \leq k \leq n+1$.

Moreover, one needs to extend the clash detection, viz., there is a clash in a constraint system S if $\{x\colon \leq nr.C\} \cup \{x r y_i \mid 1 \leq i \leq n+1\} \cup \{y_i : C \mid 1 \leq i \leq n+1\} \cup \{y_i \neq y_j \mid 1 \leq i < j \leq n+1\} \subseteq S$ for individual names $x, y_1, \ldots, y_{n+1}$, a nonnegative integer n, a role name r and a concept description C.

Exercise 7.2(b)

Transform the subsumption $A \sqsubseteq B$ to a NNF concept description

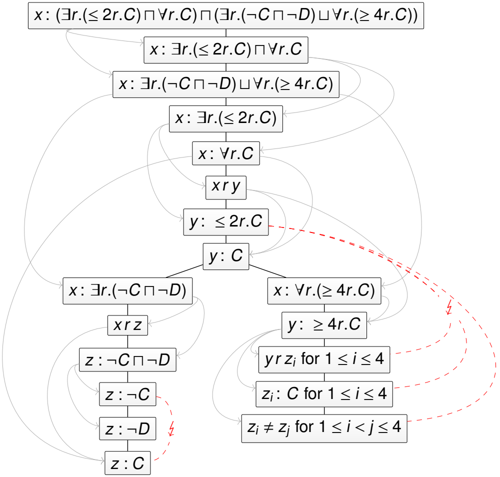
$A \sqsubseteq B$ iff $A \sqcap \neg B$ is unsatisfiable.

$$\begin{aligned}
A \sqcap \neg B &\equiv (\exists r.(\leq 2r.C) \sqcap \forall r.C) \sqcap \neg(\forall r.(C \sqcup D) \sqcap \exists r.(\leq 3r.C)) \\
&\equiv (\exists r.(\leq 2r.C) \sqcap \forall r.C) \sqcap (\neg\forall r.(C \sqcup D) \sqcup \neg\exists r.(\leq 3r.C)) \\
&\equiv (\exists r.(\leq 2r.C) \sqcap \forall r.C) \sqcap (\exists r.\neg(C \sqcup D) \sqcup \forall r.\neg(\leq 3r.C)) \\
&\equiv (\exists r.(\leq 2r.C) \sqcap \forall r.C) \sqcap (\exists r.(\neg C \sqcap \neg D) \sqcup \forall r.(\geq 4r.C))
\end{aligned}$$

Now we construct a constraint set for this description using the tableau algorithm, starting from $\{x : A \sqcap \neg B\}$.

Exercise 7.2(b)

Constraint tableau for $(\exists r.(\leq 2 r.C) \sqcap \forall r.C) \sqcap (\exists r.(\neg C \sqcap \neg D) \sqcup \forall r.(\geq 4 r.C))$



Both constraint systems contain a clash, thus $A \sqcap \neg B$ is not satisfiable $\rightsquigarrow A \sqsubseteq B$

Exercise 7.2(c) – Tableau Algorithm

(c) Explain how one can you use the tableau procedure to retrieve all instances of the concept C' given the ABox $\mathcal{A}$. Exemplify your approach by choosing any of the individuals and prove or disprove that it is an instance of C'.

- $\mathcal{A} = \{A(a), A(c), B(b), \neg C(d), r(b, c), r(c, d), s(a, b), s(a, c), s(c, c)\}$
- $C' \doteq \exists s.B \sqcup \exists s.\exists r.\neg C$

Exercise 7.2(c)

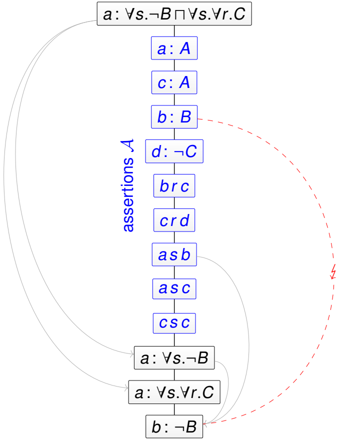
Determine the instances of C' using the tableau algorithm

- An individual a' is an instance of concept C', if for all satisfying interpretations $\mathcal{I}$ it holds that $a'^{\mathcal{I}} \in C'^{\mathcal{I}}$.
- Thus, we need to show that there is no model with $a'^{\mathcal{I}} \notin C'^{\mathcal{I}}$.
- Idea: View the individuals as variables and try to construct a tableau for $a' : \neg C'$, using the given assertions $\mathcal{A}$ as initial constraints.
- To do so, we first transform $\neg C'$ to NNF:

$$\neg C' \equiv \neg(\exists s.B \sqcup \exists s.\exists r.\neg C) \equiv \forall s.\neg B \sqcap \forall s.\forall r.C$$

Exercise 7.2(c)

Is a an instance of C'?

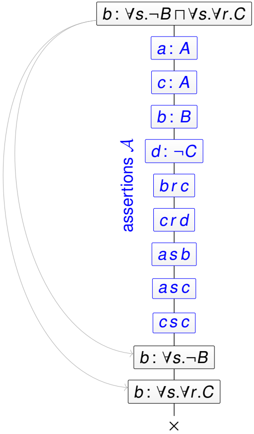


We cannot construct a model with $a^{\mathcal{I}} \notin C'^{\mathcal{I}}$

$\rightsquigarrow$ a is an instance of C'

Exercise 7.2(c)

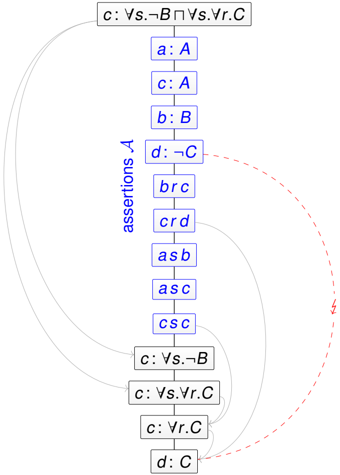
Is b an instance of C'?



For the model $\mathcal{I}$ with $A^{\mathcal{I}} = \{a, c\}$, $B^{\mathcal{I}} = \{b\}$, $C^{\mathcal{I}} = \emptyset$, $r^{\mathcal{I}} = \{(b, c), (c, d)\}$ and $s^{\mathcal{I}} = \{(a, b), (a, c), (c, d)\}$ we get $b^{\mathcal{I}} \notin C'^{\mathcal{I}}$. Thus b is not an instance of C'.

Exercise 7.2(c)

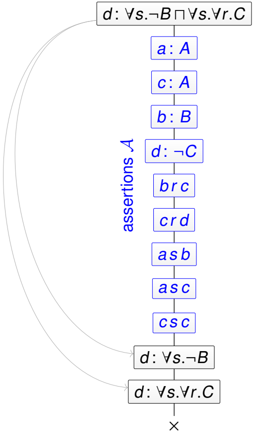
Is c an instance of C'?



$\rightsquigarrow$ c is an instance of C'

Exercise 7.2(c)

Is d an instance of C'?



With the same interpretation $\mathcal{I}$ we used for b, we also get $d^{\mathcal{I}} \notin C'^{\mathcal{I}}$. Thus d is not an instance of C'.