Der ABox-Vervollständigungsalgorithmus für ALC

Christoph Benzmüller

Freie Universität Berlin

Freie Universität Berlin

Handouts und weiteres Begleitmaterial

http://christoph-benzmueller.de/varia/ALC_Handout.pdf
http://christoph-benzmueller.de/varia/ALC_in_Isabelle.thy
http://christoph-benzmueller.de/varia/ALC_in_Isabelle.mov

Literatur (siehe auch Referenzen darin)

- The Description Logic Handbook (F. Baader, D. Calvanese, D. L. McGuiness, D. Nardi and P. F. Patel-Schneider, eds.), Cambridge University Press, 2nd edition, May 2010.
  Siehe Kapitel Basic Description Logics von F. Baader und W. Nutt.
- F. Baader and C. Lutz. Description Logic. The Handbook of Modal Logic (P. Blackburn, J. van Benthem, and F. Wolter, eds.), Elsevier, 2006.
- Matthias Knorr and Pascal Hitzler, Description Logics. Handbook of the History of Logic—Volume 9. Computational Logic (Dov M. Gabbay, Jörg H. Siekmann and John Woods, eds.), Elsevier, November 2014.
- ...

Syntax	Semantik	Beschreibung	Beispiele
A	$A^I \subseteq \Delta^I$	atomares Konzept	$Human, Female, \ldots$
r	$r^I \subseteq \Delta^I \times \Delta^I$	binäre Relation	$married, \ldots$
$\bot$	$\emptyset$	leeres Konzept	
$\top$	Δ^I	universelles Konzept	
$\sim A$	$\Delta^I \setminus A^I$	Komplement	$\sim Female$
$A \sqcup B$	$A^I \cup B^I$	Disjunktion	$Female \sqcup Male$
$A \sqcap B$	$A^I \cap B^I$	Konjunktion	$Female \sqcap Human$
$\exists r\ C$	$\{x \mid \exists y. r^I(x,y) \wedge C^I(y)\}$	Existentielle Restriktion	$\exists married\ Female$
$\forall r\ C$	$\{x \mid \forall y. r^I(x,y) \rightarrow C^I(y)\}$	Universelle Restriktion	$\forall married\ Female$

Einfache Beweisaufgaben (nützliche Lemmata)

$$\top = A \ \sqcup \sim A \qquad (L1)$$

$$\bot = \sim \top \qquad (L2)$$

$$A \sqcap B = \sim(\sim A \ \sqcup \sim B) \qquad (L3)$$

$$\forall r\ C = \sim(\exists r \sim C) \qquad (L4)$$

Syntax	Semantik	Beschreibung	Beispiele
$A \sqsubseteq B$	$A^I \subseteq B^I$	B subsumiert A	$Doctor \sqsubseteq Human$
$A \doteq B$	$A^I \sqsubseteq B^I$ *und* $B^I \sqsubseteq A^I$	A definiert durch B	$Parent \doteq Human \sqcap \exists hasChild\ Human$

Einfache Beweisaufgaben:

$$A \sqsubseteq B \quad gdw. \quad \exists C.A \doteq C \sqcap B \qquad (L5)$$

$$A \sqsubseteq B \quad gdw. \quad (A \sqcap \sim B) \sqsubseteq \bot \qquad (L6)$$

$$gdw. \quad \exists x.(A \sqcap \sim B)(x) \text{ unerfüllbar}$$

TBox (Terminologisches Wissen, Taxonomie)

Beispiel:

$$\begin{aligned} HappyMan \doteq{} & Human \sqcap {\sim}Female \sqcap \\ & (\exists married\ Doctor) \sqcap (\forall hasChild(Doctor \sqcup Professor)) \\ Doctor \sqsubseteq{} & Human \end{aligned}$$

ABox (Assertionelles Wissen, Annahmen über konkrete Individuen)

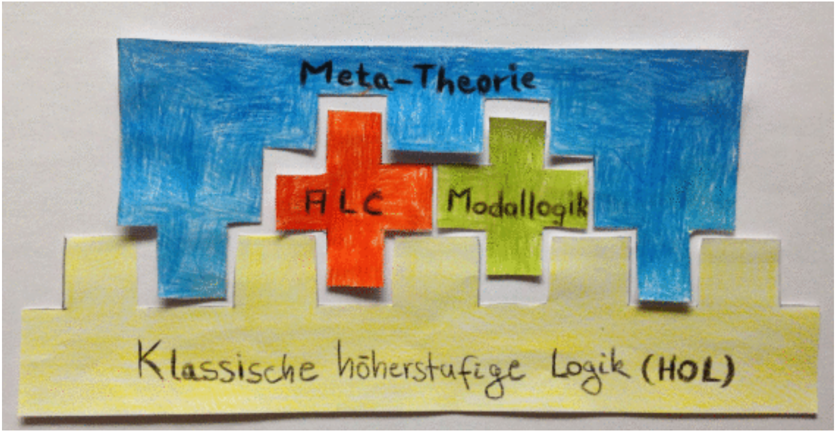
Beispiel:

$$HappyMan(BOB), \quad hasChild(BOB, MARY), \quad {\sim}Doctor(MARY)$$

alternative Schreibweise

$$BOB : HappyMan, \quad (BOB, MARY) : hasChild, \quad MARY : {\sim}Doctor$$

EINSCHUB: Wissensmodellierung & Inferenz über ALC in HOL



Animation: Max Benzmüller (5 Jahre)

EINSCHUB: Meta-Theorie von ALC in Isabelle/HOL

Isabelle

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

TECHNISCHE UNIVERSITÄT MÜNCHEN

Home
Overview
Installation
Documentation
Community

Site Mirrors:
Cambridge (.uk)
Munich (.de)
Sydney (.au)

What is Isabelle?

Isabelle is a generic proof assistant. It allows mathematical formulas to be expressed in a formal language and provides tools for proving those formulas in a logical calculus. Isabelle is developed at University of Cambridge (Larry Paulson), Technische Universität München (Tobias Nipkow) and Université Paris-Sud (Makarius Wenzel). See the Isabelle overview for a brief introduction.

Now available: Isabelle2014

Download for Mac OS X

Download for Linux - Download for Windows

Some highlights:

- Improved Isabelle/jEdit Prover IDE: navigation, completion, spell-checking, Query panel, Simplifier Trace panel.
- Support for auxiliary files within the Prover IDE, notably Isabelle/ML.
- Support for official Standard ML within the Prover IDE, independently of Isabelle theory and proof development.
- HOL: BNF datatypes and codatatypes within theory Main, with numerous add-on tools.
- HOL tool enhancements: Nitpick, Sledgehammer.
- HOL: internal SAT solver "cdclite" with models and proof traces.
- HOL: updated SMT module, with support for SMT-LIB 2 and recent versions of Z3, as well as CVC3, CVC4.
- HOL: numerous library enhancements: main HOL, HOL-Word, HOL-Multivariate_Analysis, HOL-Probability.
- System integration: improved support of LaTeX on Windows platform.
- Updated and extended manuals: codegen, datatypes, implementation, isar-ref, jedit, system.

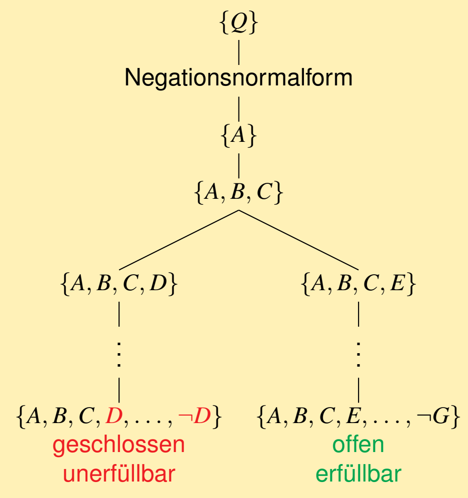
See also the cumulative NEWS.

Inferenzprobleme für $\mathcal{ALC}$

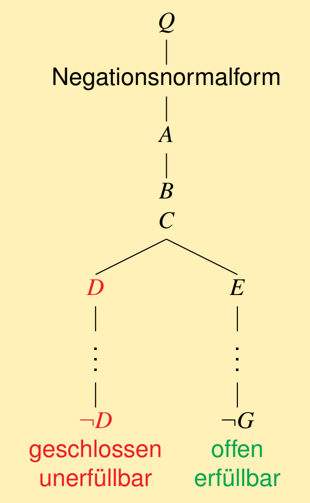
1. Ist eine ABox $\{x_1 : C_1, \ldots, x_n : C_n\}$ erfüllbar?
   - ABox-Vervollständigungsalgorithmus für ALC
   - Tableau-basiertes Entscheidungsverfahren (PSPACE)
   - ... kommt jetzt ...
2. $(TBox, ABox) \models A \sqsubseteq B$
   - Subsumiert Konzept B das Konzept A bzgl. gegebener $(TBox, ABox)$?
   - Reduzierbar auf (1.)
   - ... falls noch Zeit bleibt ...
3. $(TBox, ABox) \not\models False$
   - Erfüllbarkeitstest für gegebene T- und $ABox$
   - Reduzierbar auf (1.)
4. $(TBox, ABox) \models A \not\sqsubseteq \bot$
   - Ist Konzept A erfüllbar bzgl. gegebener T- und $ABox$?
   - Reduzierbar auf (1.)
5. $(TBox, ABox) \models a : A$
   - Ist a eine Instanz von A bzgl. gegebener T- und $ABox$?
   - Reduzierbar auf (1.)

Auffrischung: Tableau-Verfahren

Analyse der Unerfüllbarkeit bzw. Erfüllbarkeit (Modell) einer Formel Q



Manipulation von (konj.) Formelmengen



Alternative Darstellung

Beispiel — Erfüllbarkeitstest für $Q := \neg(\neg a \vee (p \vee \neg p) \wedge \neg(s \wedge \neg a))$

$$\neg(\neg a \vee (p \vee \neg p) \wedge \neg(s \wedge \neg a))$$

Normaliserung:

$$\begin{array}{rcl} \neg(A \wedge B) & \longrightarrow & \neg A \vee \neg B \\ \neg(A \vee B) & \longrightarrow & \neg A \wedge \neg B \\ \neg\neg A & \longrightarrow & A \end{array}$$

Beispiel — Erfüllbarkeitstest für $Q := \neg(\neg a \vee (p \vee \neg p) \wedge \neg(s \wedge \neg a))$

$$\neg(\neg a \vee (p \vee \neg p) \wedge \neg(s \wedge \neg a))$$

|

Negationsnormalform

|

$$a \wedge ((\neg p \wedge p) \vee (s \wedge \neg a))$$

$$\frac{A \wedge B}{\begin{array}{c}A\\B\end{array}} \; \wedge Regel$$

Beispiel — Erfüllbarkeitstest für $Q := \neg(\neg a \lor (p \lor \neg p) \land \neg(s \land \neg a))$

$$\neg(\neg a \lor (p \lor \neg p) \land \neg(s \land \neg a))$$

|

Negationsnormalform

|

$$a \land ((\neg p \land p) \lor (s \land \neg a))$$

|

$$a$$

$$(\neg p \land p) \lor (s \land \neg a)$$

$$\frac{A \lor B}{A \qquad B} \lor Regel$$

Beispiel — Erfüllbarkeitstest für $Q := \neg(\neg a \vee (p \vee \neg p) \wedge \neg(s \wedge \neg a))$

Freie Universität Berlin

$$\neg(\neg a \vee (p \vee \neg p) \wedge \neg(s \wedge \neg a))$$

Negationsnormalform

$$a \wedge ((\neg p \wedge p) \vee (s \wedge \neg a))$$

$$a$$

$$(\neg p \wedge p) \vee (s \wedge \neg a)$$

$$\neg p \wedge p \qquad\qquad s \wedge \neg a$$

$$\neg p$$

$$p$$

geschlossen

Beispiel — Erfüllbarkeitstest für $Q := \neg(\neg a \vee (p \vee \neg p) \wedge \neg(s \wedge \neg a))$

$$\neg(\neg a \vee (p \vee \neg p) \wedge \neg(s \wedge \neg a))$$

Negationsnormalform

$$a \wedge ((\neg p \wedge p) \vee (s \wedge \neg a))$$

$$a$$
$$(\neg p \wedge p) \vee (s \wedge \neg a)$$

$$\neg p \wedge p \qquad\qquad s \wedge \neg a$$

$$\neg p$$
$$p$$

geschlossen

$$\frac{A \wedge B}{\begin{array}{c} A \\ B \end{array}} \ \wedge Regel$$

Beispiel — Erfüllbarkeitstest für $Q := \neg(\neg a \vee (p \vee \neg p) \wedge \neg(s \wedge \neg a))$ —unerfüllbar—

$$\neg(\neg a \vee (p \vee \neg p) \wedge \neg(s \wedge \neg a))$$

Negationsnormalform

$$a \wedge ((\neg p \wedge p) \vee (s \wedge \neg a))$$

$$a$$
$$(\neg p \wedge p) \vee (s \wedge \neg a)$$

$\neg p \wedge p$	$s \wedge \neg a$
$\neg p$	s
p	$\neg a$
geschlossen	geschlossen

—unerfüllbar—

Beispiel — Erfüllbarkeitstest für $Q := \neg(\neg b \vee (p \vee \neg p) \wedge \neg(s \wedge \neg a))$ **—erfüllbar—**

$$\neg(\neg b \vee (p \vee \neg p) \wedge \neg(s \wedge \neg a))$$

Negationsnormalform

$$a \wedge ((\neg p \wedge p) \vee (s \wedge \neg a))$$

$$b$$

$$(\neg p \wedge p) \vee (s \wedge \neg a)$$

$\neg p \wedge p$	$s \wedge \neg a$
$\neg p$	s
p	$\neg a$
geschlossen	offen

—erfüllbar—

ABox-Vervollständigungsalgorithmus für $\mathcal{ALC}$

1. Ist eine ABox $\{x_1 : C_1, \ldots, x_n : C_n\}$ erfüllbar?
   - ABox-Vervollständigungsalgorithmus für ALC
   - Tableau-basiertes Entscheidungsverfahren (ExpTime/PSPACE)
   - . . . kommt jetzt . . .

ABox-Beispiel:

$$\{x : Human \sqcap \sim Female \sqcap (\exists married\ Doctor) \sqcap (\forall hasChild(Doctor \sqcup Professor))\}$$

Frage: Ist diese ABox erfüllbar?

Idee und Vorgehen:

1. Initialisiere Tableau mit ABox $\{x : Human \sqcap \ldots Professor))\}$
2. Wende $\mathcal{ALC}$-Tableauregeln (Vervollständigungsregeln) erschöpfend an
3. Tableau ist offen $\longrightarrow$ *ABox* erfüllbar
   Tableau ist geschlossen $\longrightarrow$ *ABox* unerfüllbar

$\mathcal{ALC}$-Normalisierungsregeln:

$$\begin{array}{rcl}
\sim(A \sqcap B) & \longrightarrow & \sim A \sqcup \sim B \\
\sim(A \sqcup B) & \longrightarrow & \sim A \sqcap \sim B \\
\sim\sim A & \longrightarrow & A \\
\sim(\forall r\ C) & \longrightarrow & \exists r \sim C \\
\sim(\exists r\ C) & \longrightarrow & \forall r \sim C
\end{array}$$

Beispiel (sehr leichte Übung):

$$x : \sim(\exists hasChild(\sim Doctor \sqcap \sim Professor))$$

$$|$$

$$Negationsnormalform$$

$$|$$

$$x : (\forall hasChild(Doctor \sqcup Professor))$$

$\mathcal{ALC}$-Tableauregeln (Vervollständigungsregeln)

$$\frac{x : C \sqcap D}{\begin{array}{c} x : C \\ x : D \end{array}} \ \sqcap Regel$$

$S \xrightarrow{\sqcap\textbf{Regel}} S \cup \{x : C, x : D\}$
falls
(1) $x : C \sqcap D$ ist in S
(2) $x : C$ und $x : D$ sind noch nicht in S

$$\frac{x : C \sqcup D}{x : C \quad x : D} \ \sqcup Regel$$

$S \xrightarrow{\sqcup\textbf{Regel}} S \cup \{x : C\} \quad | \quad S \cup \{x : D\}$
falls
(1) $x : C \sqcup D$ ist in S
(2) weder $x : C$ noch $x : D$ sind in S

$$\frac{\begin{array}{c} x : \forall r\, C \\ (x, y) : r \end{array}}{y : C} \ \forall Regel$$

$S \xrightarrow{\forall\textbf{Regel}} S \cup \{y : C\}$
falls
(1) $x : \forall r\, C$ ist in S (2) $(x, y) : r$ ist in S
(3) $y : C$ ist nicht in S

$$\frac{x : \exists r\, C}{\begin{array}{c} (x, y) : r \\ y : C \end{array}} \ \exists Regel$$

$S \xrightarrow{\exists\textbf{Regel}} S \cup \{(x, y) : r, y : C\}$
falls
(1) $x : \exists r\, C$ ist in S (2) y ist neues Variablensymbol
(3) es kein gibt z, so dass $(x, z) : r$ und $z : C$ sind in S

$$\frac{\begin{array}{c} x : C \\ x : \sim C \end{array}}{geschl.} \ Clash \ \frac{x : \perp}{geschl.}$$

$S \xrightarrow{\textbf{Clash}} S \cup \{geschlossen\}$
falls
(1) $x : C$ und $x : \sim C$ sind in S oder (2) $x : \perp$ in S

ABox-Vervollständigung am Beispiel

$\{x : Human \sqcap \sim Female \sqcap (\exists married\ Doctor) \sqcap (\forall hasChild(Doctor \sqcup Professor))\}$

$$x : Human \sqcap \sim Female \sqcap (\exists married\ Doctor) \sqcap (\forall hasChild(Doctor \sqcup Professor))$$

$\mathcal{ALC}$-Normalisierungsregeln:

$$\begin{aligned}
\sim(A \sqcap B) &\longrightarrow \sim A \sqcup \sim B \\
\sim(A \sqcup B) &\longrightarrow \sim A \sqcap \sim B \\
\sim\sim A &\longrightarrow A \\
\sim(\forall r\ C) &\longrightarrow \exists r \sim C \\
\sim(\exists r\ C) &\longrightarrow \forall r \sim C
\end{aligned}$$

ABox-Vervollständigung am Beispiel

$\{x : Human \sqcap \sim Female \sqcap (\exists married\ Doctor) \sqcap (\forall hasChild(Doctor \sqcup Professor))\}$

$$x : Human \sqcap \sim Female \sqcap (\exists married\ Doctor) \sqcap (\forall hasChild(Doctor \sqcup Professor))$$

$$|$$

$$Negationsnormalform$$

$$|$$

$$x : Human \sqcap \sim Female \sqcap (\exists married\ Doctor) \sqcap (\forall hasChild(Doctor \sqcup Professor))$$

$$\frac{x : C \sqcap D}{\begin{array}{c} x : C \\ x : D \end{array}} \ \sqcap Regel$$

ABox-Vervollständigung am Beispiel

$\{x : Human \ \sqcap \sim Female \sqcap (\exists married\ Doctor) \sqcap (\forall hasChild(Doctor \sqcup Professor))\}$

$$x : Human \sqcap \sim Female \sqcap (\exists married\ Doctor) \sqcap (\forall hasChild(Doctor \sqcup Professor))$$

$$|$$

$$Negationsnormalform$$

$$|$$

$$x : Human \sqcap \sim Female \sqcap (\exists married\ Doctor) \sqcap (\forall hasChild(Doctor \sqcup Professor))$$

$$|$$

$$x : Human$$

$$x : \sim Female \sqcap (\exists married\ Doctor) \sqcap (\forall hasChild(Doctor \sqcup Professor))$$

$$\frac{x : C \sqcap D}{\begin{array}{c} x : C \\ x : D \end{array}} \ \sqcap Regel$$

ABox-Vervollständigung am Beispiel

$\{x : Human \sqcap \sim Female \sqcap (\exists married\ Doctor) \sqcap (\forall hasChild (Doctor \sqcup Professor))\}$

$$x : Human \sqcap \sim Female \sqcap (\exists married\ Doctor) \sqcap (\forall hasChild (Doctor \sqcup Professor))$$

$$|$$

$$Negationsnormalform$$

$$|$$

$$x : Human \sqcap \sim Female \sqcap (\exists married\ Doctor) \sqcap (\forall hasChild (Doctor \sqcup Professor))$$

$$|$$

$$x : Human$$

$$x : \sim Female \sqcap (\exists married\ Doctor) \sqcap (\forall hasChild (Doctor \sqcup Professor))$$

$$|$$

$$x : \sim Female$$

$$x : (\exists married\ Doctor) \sqcap (\forall hasChild (Doctor \sqcup Professor))$$

$$\frac{x : C \sqcap D}{\begin{array}{c} x : C \\ x : D \end{array}} \ \sqcap Regel$$

ABox-Vervollständigung am Beispiel

$\{x : Human \sqcap \sim Female \sqcap (\exists married\ Doctor) \sqcap (\forall hasChild(Doctor \sqcup Professor))\}$

$$x : Human \sqcap \sim Female \sqcap (\exists married\ Doctor) \sqcap (\forall hasChild(Doctor \sqcup Professor))]$$

$$|$$

$$Negationsnormalform$$

$$|$$

$$x : Human \sqcap \sim Female \sqcap (\exists married\ Doctor) \sqcap (\forall hasChild(Doctor \sqcup Professor))$$

$$|$$

$$x : Human$$
$$x : \sim Female \sqcap (\exists married\ Doctor) \sqcap (\forall hasChild(Doctor \sqcup Professor))$$

$$|$$

$$x : \sim Female$$
$$x : \sqcap(\exists married\ Doctor) \sqcap (\forall hasChild(Doctor \sqcup Professor))$$

$$|$$

$$x : \exists married\ Doctor$$
$$x : (\forall hasChild(Doctor \sqcup Professor)$$

$$\frac{x : \exists r\ C}{\begin{array}{c}(x, y) : r \\ y : C\end{array}}\ \exists Regel$$

ABox-Vervollständigung am Beispiel

$\{x : Human \sqcap \sim Female \sqcap (\exists married\ Doctor) \sqcap (\forall hasChild(Doctor \sqcup Professor))\}$

$$x : Human \sqcap \sim Female \sqcap (\exists married\ Doctor) \sqcap (\forall hasChild(Doctor \sqcup Professor))]$$

$$|$$

$$Negationsnormalform$$

$$|$$

$$x : Human \sqcap \sim Female \sqcap (\exists married\ Doctor) \sqcap (\forall hasChild(Doctor \sqcup Professor))$$

$$|$$

$$x : Human$$
$$x : \sim Female \sqcap (\exists married\ Doctor) \sqcap (\forall hasChild(Doctor \sqcup Professor))$$

$$|$$

$$x : \sim Female$$
$$x : \sqcap(\exists married\ Doctor) \sqcap (\forall hasChild(Doctor \sqcup Professor))$$

$$|$$

$$x : \exists married\ Doctor$$
$$x : (\forall hasChild(Doctor \sqcup Professor)$$

$$|$$

$$(x, y) : married$$
$$y : Doctor$$

offen

—erfüllbar—

Subsumtionstest durch Reduktion auf Erfüllbarkeitstest

2. $(TBox, ABox) \models A \sqsubseteq B$

- Subsumiert Konzept B das Konzept A bzgl. gegebener $(TBox, ABox)$?
- Reduzierbar auf ABox-Vervollständigkeitsalgorithmus
- … falls noch Zeit bleibt …

Beispiel:

TBox:

$HappyMan \doteq$

$Human \sqcap \sim Female \sqcap (\exists married\ Doctor) \sqcap (\forall hasChild(Doctor \sqcup Professor))$

$Doctor \sqsubseteq Human$

ABox:

$\emptyset$

Anfrage:

$HappyMan \sqsubseteq \exists married\ Human$

Subsumtionstest durch Reduktion auf Erfüllbarkeitstest

2. $(TBox, ABox) \models A \sqsubseteq B$

- Subsumiert Konzept B das Konzept A bzgl. gegebener $(TBox, ABox)$?
- Reduzierbar auf ABox-Vervollständigkeitsalgorithmus
- . . . falls noch Zeit bleibt . . .

Beispiel:

TBox:

$HM \doteq H \sqcap \sim F \sqcap (\exists ma\ D) \sqcap (\forall hc(D \sqcup P))$

$D \sqsubseteq H$

ABox:

$\emptyset$

Anfrage:

$HM \sqsubseteq \exists ma\ H$

Idee und Vorgehen

1. Modifiziere TBox so, dass sie nur Definitionen ($\ldots \doteq \ldots$) enthält.
2. Ersetze alle definierten Konzepte in Anfrage durch ihr Definiendum.
3. Wende Lemma L5 auf Anfrage an.

(Lemma L5: $A \sqsubseteq B$ gdw. $\exists x.(A \sqcap \sim B)(x)$ unerfüllbar)

Subsumtionstest durch Reduktion auf Erfüllbarkeitstest

2. $(TBox, ABox) \models A \sqsubseteq B$

- Subsumiert Konzept B das Konzept A bzgl. gegebener $(TBox, ABox)$?
- Reduzierbar auf ABox-Vervollständigkeitsalgorithmus
- ... falls noch Zeit bleibt ...

Beispiel:

TBox:

$HM \doteq H \sqcap \sim F \sqcap (\exists ma\ D) \sqcap (\forall hc(D \sqcup P))$

$D \sqsubseteq H$ (Lemma L5: gilt $D \sqsubseteq H$ gdw. $\exists D^*.D \doteq D^* \sqcap H$)

ABox:

$\emptyset$

Anfrage:

$HM \sqsubseteq \exists ma\ H$

Idee und Vorgehen

1. Modifiziere TBox so, dass sie nur Definitionen $(\ldots \doteq \ldots)$ enthält.
2. Ersetze alle definierten Konzepte in Anfrage durch ihr Definiendum.
3. Wende Lemma L6 auf Anfrage an.

(Lemma L6: $A \sqsubseteq B$ gdw. $\exists x.(A \sqcap \sim B)(x)$ unerfüllbar)

Subsumtionstest durch Reduktion auf Erfüllbarkeitstest

2. $(TBox, ABox) \models A \sqsubseteq B$

- Subsumiert Konzept B das Konzept A bzgl. gegebener $(TBox, ABox)$?
- Reduzierbar auf ABox-Vervollständigkeitsalgorithmus
- ... falls noch Zeit bleibt ...

Beispiel:

TBox:

$HM \doteq H \sqcap \sim F \sqcap (\exists ma\ D) \sqcap (\forall hc(D \sqcup P))$

$D \doteq D^* \sqcap H$ (Lemma L5: gilt $D \sqsubseteq H$ gdw. $\exists D^*.D \doteq D^* \sqcap H$)

ABox:

$\emptyset$

Anfrage:

$HM \sqsubseteq \exists ma\ H$

Idee und Vorgehen

1. Modifiziere TBox so, dass sie nur Definitionen $(\ldots \doteq \ldots)$ enthält.
2. Ersetze alle definierten Konzepte in Anfrage durch ihr Definiendum.
3. Wende Lemma L6 auf Anfrage an.

(Lemma L6: $A \sqsubseteq B$ gdw. $\exists x.(A \sqcap \sim B)(x)$ unerfüllbar)

Subsumtionstest durch Reduktion auf Erfüllbarkeitstest

2. $(TBox, ABox) \models A \sqsubseteq B$

- Subsumiert Konzept B das Konzept A bzgl. gegebener $(TBox, ABox)$?
- Reduzierbar auf ABox-Vervollständigkeitsalgorithmus
- . . . falls noch Zeit bleibt . . .

Beispiel:

TBox:

$HM \doteq H \sqcap \sim F \sqcap (\exists ma\ D) \sqcap (\forall hc(D \sqcup P))$

$D \doteq D^* \sqcap H$

ABox:

$\emptyset$

Anfrage:

$HM \sqsubseteq \exists ma\ H$

Idee und Vorgehen

1. Modifiziere TBox so, dass sie nur Definitionen $(\ldots \doteq \ldots)$ enthält.
2. Ersetze alle definierten Konzepte in Anfrage durch ihr Definiendum.
3. Wende Lemma L6 auf Anfrage an.

(Lemma L6: $A \sqsubseteq B$ gdw. $\exists x.(A \sqcap \sim B)(x)$ unerfüllbar)

Subsumtionstest durch Reduktion auf Erfüllbarkeitstest

2. $(TBox, ABox) \models A \sqsubseteq B$

- Subsumiert Konzept B das Konzept A bzgl. gegebener $(TBox, ABox)$?
- Reduzierbar auf ABox-Vervollständigkeitsalgorithmus
- ... falls noch Zeit bleibt ...

Beispiel:

TBox:

$HM \doteq H \sqcap \sim F \sqcap (\exists ma\ D) \sqcap (\forall hc(D \sqcup P))$

$D \doteq D^* \sqcap H$

ABox:

$\emptyset$

Anfrage:

$(H \sqcap \sim F \sqcap (\exists ma(D^* \sqcap H)) \sqcap (\forall hc((D^* \sqcap H) \sqcup P))) \sqsubseteq \exists ma\ H$

Idee und Vorgehen

1. Modifiziere TBox so, dass sie nur Definitionen $(\ldots \doteq \ldots)$ enthält.
2. Ersetze alle definierten Konzepte in Anfrage durch ihr Definiendum.
3. Wende Lemma L6 auf Anfrage an.

(Lemma L6: $A \sqsubseteq B$ gdw. $\exists x.(A \sqcap \sim B)(x)$ unerfüllbar)

Subsumtionstest durch Reduktion auf Erfüllbarkeitstest

2. $(TBox, ABox) \models A \sqsubseteq B$

- Subsumiert Konzept B das Konzept A bzgl. gegebener $(TBox, ABox)$?
- Reduzierbar auf ABox-Vervollständigkeitsalgorithmus
- ... falls noch Zeit bleibt ...

Beispiel:

TBox:

$$HM \doteq H \sqcap \sim F \sqcap (\exists ma\ D) \sqcap (\forall hc(D \sqcup P))$$

$$D \doteq D^* \sqcap H$$

ABox:

$$\emptyset$$

Anfrage:

$$(H \sqcap \sim F \sqcap (\exists ma(D^* \sqcap H)) \sqcap (\forall hc((D^* \sqcap H) \sqcup P))) \sqsubseteq \exists ma\ H$$

Idee und Vorgehen

1. Modifiziere TBox so, dass sie nur Definitionen $(\ldots \doteq \ldots)$ enthält.
2. Ersetze alle definierten Konzepte in Anfrage durch ihr Definiendum.
3. Wende Lemma L6 auf Anfrage an.

(Lemma L6: $A \sqsubseteq B$ gdw. $\exists x.(A \sqcap \sim B)(x)$ unerfüllbar)

Subsumtionstest durch Reduktion auf Erfüllbarkeitstest

2. $(TBox, ABox) \models A \sqsubseteq B$
   - Subsumiert Konzept B das Konzept A bzgl. gegebener $(TBox, ABox)$?
   - Reduzierbar auf ABox-Vervollständigkeitsalgorithmus
   - ... falls noch Zeit bleibt ...

Beispiel:

TBox:

$HM \doteq H \sqcap \sim F \sqcap (\exists ma\ D) \sqcap (\forall hc (D \sqcup P))$

$D \doteq D^* \sqcap H$

ABox:

$\emptyset$

Anfrage:

$\exists x.((H \sqcap \sim F \sqcap (\exists ma (D^* \sqcap H)) \sqcap (\forall hc ((D^* \sqcap H) \sqcup P))) \sqcap \sim \exists ma\ H)(x)$

unerfüllbar

Idee und Vorgehen

1. Modifiziere TBox so, dass sie nur Definitionen $(\ldots \doteq \ldots)$ enthält.
2. Ersetze alle definierten Konzepte in Anfrage durch ihr Definiendum.
3. Wende Lemma L6 auf Anfrage an.

(Lemma L6: $A \sqsubseteq B$ gdw. $\exists x.(A \sqcap \sim B)(x)$ unerfüllbar)

Freie Universität Berlin

2. $(TBox, ABox) \models A \sqsubseteq B$
   - Subsumiert Konzept B das Konzept A bzgl. gegebener $(TBox, ABox)$?
   - Reduzierbar auf ABox-Vervollständigkeitsalgorithmus
   - ... falls noch Zeit bleibt ...

Beispiel:

TBox:

$HM \doteq H \sqcap \sim F \sqcap (\exists ma\ D) \sqcap (\forall hc(D \sqcup P))$

$D \doteq D^* \sqcap H$

ABox:

$\emptyset$

Anfrage:

$x : ((H \sqcap \sim F \sqcap (\exists ma(D^* \sqcap H)) \sqcap (\forall hc((D^* \sqcap H) \sqcup P))) \sqcap \sim\exists ma\ H)$

führt zu geschlossenem Tableau

Idee und Vorgehen

1. Modifiziere TBox so, dass sie nur Definitionen $(\ldots \doteq \ldots)$ enthält.
2. Ersetze alle definierten Konzepte in Anfrage durch ihr Definiendum.
3. Wende Lemma L6 auf Anfrage an.

(Lemma L6: $A \sqsubseteq B$ gdw. $\exists x.(A \sqcap \sim B)(x)$ unerfüllbar)

$$x : H \sqcap \sim F \sqcap (\exists ma(D^* \sqcap H)) \sqcap (\forall hc((D^* \sqcap H) \sqcup P)) \sqcap \forall ma{\sim}H$$

$|$

$$x : H$$

$$x : \sim F \sqcap (\exists ma(D^* \sqcap H)) \sqcap (\forall hc((D^* \sqcap H) \sqcup P)) \sqcap \forall ma{\sim}H$$

$|$

$$x : \sim F$$

$$x : (\exists ma(D^* \sqcap H)) \sqcap (\forall hc((D^* \sqcap H) \sqcup P)) \sqcap \forall ma{\sim}H$$

$|$

$$x : (\exists ma(D^* \sqcap H))$$

$$x : (\forall hc((D^* \sqcap H) \sqcup P)) \sqcap \forall ma{\sim}H$$

$|$

$$x : (\forall hc((D^* \sqcap H) \sqcup P))$$

$$x : \forall ma{\sim}H$$

$|$

$$(x, y) : ma$$

$$y : D^* \sqcap H$$

$|$

$$y : D^*$$

$$y : H$$

$|$

$$y : \sim H$$

geschlossen

(d.h. die ursprüngliche Subsumtionsanfrage gilt)

Weitere Themen

- Reduktion von weiteren Inferenzproblemen auf Erfüllbarkeitstest
- Korrektheit
- Terminierung
- Vollständigkeit
- Komplexität
- Spracherweiterungen/-restriktionen und Komplexität
- Beispiele und Anwendungen
- Systeme: RACER, PELLET, FACT++, HERMIT

. . . in den kommenden Vorlesungen . . .