Semantic Web Technologies II

SS 2008

28.05.2008

Reasoning

Dr. Peter Haase
PD Dr. Pascal Hitzler
Dr. Steffen Lamparter
Denny Vrandecic

Was ist eigentlich ein Beweis?

- Wann ist ein Beweis "richtig"?
- Wer entscheidet das?
- Was für eine Rolle spielt das "Überzeugen"?

- Was sind mathematische Objekte? "Existieren" sie?
- Konkret: Was ist eine Menge?

- Warum können wir die Welt (oder Teile davon) mit Mathematik modellieren?

Übersicht

- **Grundproblem automatischen Beweisens**
- Wichtige Inferenzprobleme
- Tableauverfahren für ALC

Grundproblem automatischen Beweisens

A ist eine logische Konsequenz von K

geschrieben $K \models A$

genau dann, wenn

jedes Modell von K auch ein Modell für A ist.

- Wenn man von dieser Definition ausgeht, muss man **alle** Modelle betrachten.
- Es gibt aber **unendlich viele** davon!

Grundproblem automatischen Beweisens

Beweis*verfahren* werden benötigt, die nicht vom allgemeinen Modellbegriff ausgehen bzw. von diesem abstrahieren:

$\rightarrow$ Beweistheoretische Semantik (durch Algorithmen)

Für solche Algorithmen muss formal bewiesen werden, dass sie korrekt und vollständig bzgl. der modelltheoretischen Semantik sind!

Naive Idee

- Regeln angeben, wie man aus vorhandenem Wissen neues Wissen ableitet.
  Das macht man dann solange, bis die gesuchte Aussage abgeleitet ist.

Probleme:

- Grundsätzlich geht das schon, ist aber viel zu aufwändig.
- Wann wissen wir, ob eine Aussage *nicht* ableitbar ist?

Für manche (einfache) Logiken geht das. Nicht für ALC.

Übersicht

- Grundproblem automatischen Beweisens
- **Wichtige Inferenzprobleme**
- Tableauverfahren für ALC

Wichtige Inferenzprobleme

- Globale Konsistenz der Wissensbasis — KB $\models$ **false**?
  - Ist Wissensbasis sinnvoll?
- Klassenkonsistenz — $C \equiv \bot$?
  - Muss Klasse C leer sein?
- Klasseninklusion (Subsumption) — $C \sqsubseteq D$?
  - Strukturierung der Wissensbasis
- Klassenäquivalenz — $C \equiv D$?
  - Sind zwei Klassen eigentlich dieselbe?
- Klassendisjunktheit — $C \sqcap D = \bot$?
  - Sind zwei Klassen disjunkt?
- Klassenzugehörigkeit — $C(a)$?
  - Ist Individuum a in der Klasse C?
- Instanzgenerierung (Retrieval) „alle X mit C(X) finden“
  - Finde alle (bekannten!) Individuen zur Klasse C.

Rückführung auf Unerfüllbarkeit

- Klassenkonsistenz — $C \equiv \bot$ gdw
  - $KB \cup \{C(a)\}$ unerfüllbar (a neu)
- Klasseninklusion (Subsumption) — $C \sqsubseteq D$ gdw
  - $KB \cup \{C \sqcap \neg D(a)\}$ unerfüllbar (a neu)
- Klassenäquivalenz — $C \equiv D$ gdw
  - $C \sqsubseteq D$ und $D \sqsubseteq C$
- Klassendisjunktheit — $C \sqcap D = \bot$ gdw
  - $KB \cup \{(C \sqcap D)(a)\}$ unerfüllbar (a neu)
- Klassenzugehörigkeit — $C(a)$ gdw
  - $KB \cup \{\neg C(a)\}$ unerfüllbar (a neu)
- Instanzgenerierung (Retrieval) — alle C(X) finden
  - Prüfe Klassenzugehörigkeit für alle Individuen

Entscheidbarkeit von OWL DL

- Entscheidbarkeit: zu jedem Inferenzproblem gibt es einen immer terminierenden Algorithmus.
- OWL DL ist Fragment von FOL, also könnten (im Prinzip) FOL-Inferenzalgorithmen (Resolution, Tableaux) verwendet werden.
- Diese terminieren aber nicht immer!

- Problem: Finde immer terminierende Algorithmen! Keine „naiven“ Lösungen in Sicht!

Rückführung auf Unerfüllbarkeit

- Wir werden das Tableauverfahren behandeln

- Das Tableauverfahren versucht, auf "abstrakte" Weise ein Modell zu konstruieren.
  - misslingt das, dann ist die Wissensbasis unerfüllbar
  - gelingt es, ist sie erfüllbar

$\rightarrow$ Rückführung der Inferenzprobleme auf das Zeigen der Unerfüllbarkeit der Wissensbasis!

Übersicht

- Grundproblem automatischen Beweisens
- Wichtige Inferenzprobleme
- **Tableauverfahren für ALC**

ALC Tableauverfahren: Inhalt

- **Transformation in Negationsnormalform**
- Naives Tableauverfahren
- Tableauverfahren mit Blocking

Transformation in Negationsnormalform

Gegeben eine Wissensbasis W.

- Ersetze $C \equiv D$ durch $C \sqsubseteq D$ und $D \sqsubseteq C$.
- Ersetze $C \sqsubseteq D$ durch $\neg C \sqcup D$.
- Wende die Regeln auf der folgenden Folie an, bis es nicht mehr geht.

Resultierende Wissensbasis: NNF(W)

Negationsnormalform von W.

Negation steht nur noch direkt vor atomaren Klassen.

$\mathrm{NNF}(C) = C$, falls C atomar ist

$\mathrm{NNF}(\neg C) = \neg C$, falls C atomar ist

$\mathrm{NNF}(\neg\neg C) = \mathrm{NNF}(C)$

$\mathrm{NNF}(C \sqcup D) = \mathrm{NNF}(C) \sqcup \mathrm{NNF}(D)$

$\mathrm{NNF}(C \sqcap D) = \mathrm{NNF}(C) \sqcap \mathrm{NNF}(D)$

$\mathrm{NNF}(\neg(C \sqcup D)) = \mathrm{NNF}(\neg C) \sqcap \mathrm{NNF}(\neg D)$

$\mathrm{NNF}(\neg(C \sqcap D)) = \mathrm{NNF}(\neg C) \sqcup \mathrm{NNF}(\neg D)$

$\mathrm{NNF}(\forall R.C) = \forall R.\mathrm{NNF}(C)$

$\mathrm{NNF}(\exists R.C) = \exists R.\mathrm{NNF}(C)$

$\mathrm{NNF}(\neg\forall R.C) = \exists R.\mathrm{NNF}(\neg C)$

$\mathrm{NNF}(\neg\exists R.C) = \forall R.\mathrm{NNF}(\neg C)$

W und NNF(W) sind logisch äquivalent.

Negationsnormalform: Beispiel

$P \sqsubseteq (E \sqcap U) \sqcup \neg(\neg E \sqcup D).$

In Negationsnormalform:

$\neg P \sqcup (E \sqcap U) \sqcup (E \sqcap \neg D).$

ALC Tableauverfahren: Inhalt

- Transformation in Negationsnormalform
- **Naives Tableauverfahren**
- Tableauverfahren mit Blocking

Naives Tableauverfahren

Rückführung auf Unerfüllbarkeit

Idee:

- Gegeben Wissensbasis W.
- Versuch der Erzeugung eines Baumes (genannt *Tableau*), der ein Modell von W repräsentiert. (Eigentlich ein *Wald*.)
- Schlägt der Versuch fehl, ist W unerfüllbar.

Das Tableau

- Knoten repräsentieren Elemente des Definitionsbereichs des Modells
  $\rightarrow$ Jeder Knoten x ist mit einer Menge L(x) von Klassenausdrücken beschriftet.
  C $\in$ L(x) steht für: "x ist in der Extension von C"
- Kanten stehen für Rollenverbindungen
  $\rightarrow$ Jede Kante <x,y> is mit einer Menge von Rollennamen L(<x,y>) beschriftet.
  R $\in$ L(<x,y>) steht für: "(x,y) ist in der Extension von R"

Einfaches Beispiel

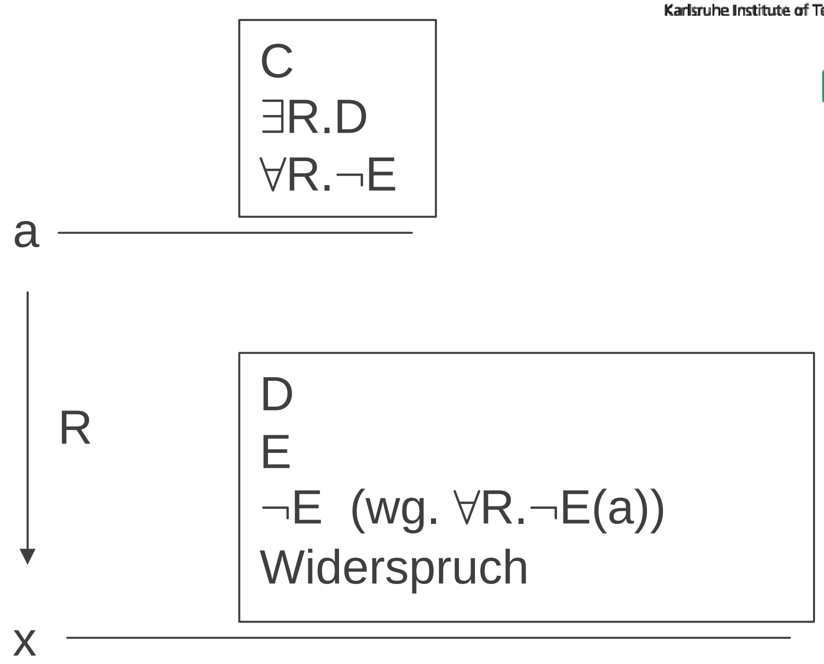
$C(a)$

$C \sqsubseteq \exists R.D$

$D \sqsubseteq E$

Folgt $(\exists R.E)(a)$?

(füge $\forall R.\neg E(a)$
hinzu und zeige
Unerfüllbarkeit)



Weiteres Beispiel

C(a)

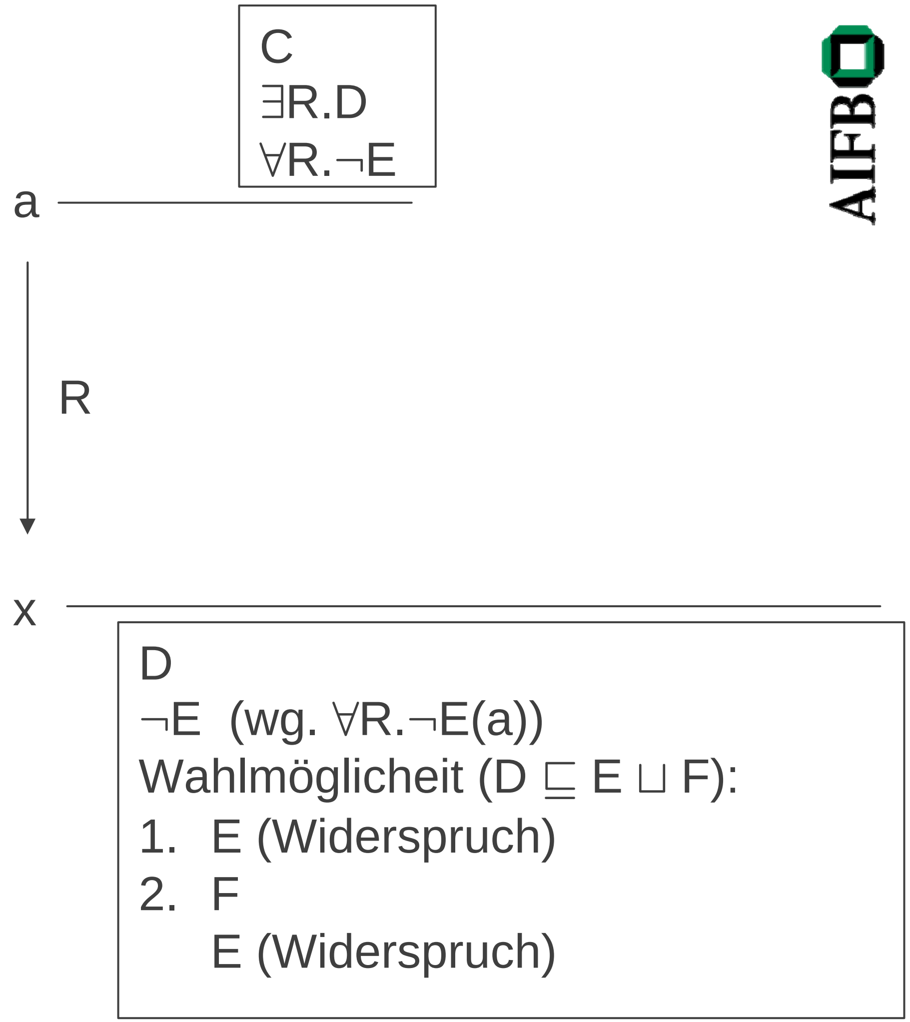
C ⊑ ∃R.D

D ⊑ E ⊔ F

F ⊑ E

Folgt (∃R.E)(a)?

(füge ∀R.¬E(a)
hinzu und zeige
Unerfüllbarkeit)



Formale Definition

- Input: K=TBox + ABox (in NNF)
- Ausgabe: Angabe, ob K erfüllbar ist

- Ein Tableau ist ein gerichteter beschrifteter Graph.
  - Knoten sind Individuen oder (neue) Variablennamen
  - Knoten x sind beschriftet mit Mengen L(x) von Klassen
  - Kanten <x,y> sind beschriftet mit Mengen L(<x,y>) von Rollennamen

Initialisierung

- Ein Knoten für jedes Individuum in der ABox.
- Jeder Knoten beschriftet mit den entsprechenden Klassennamen aus der ABox.
- Eine mit R beschriftete Kante zwischen a und b falls R(a,b) in der ABox steht.

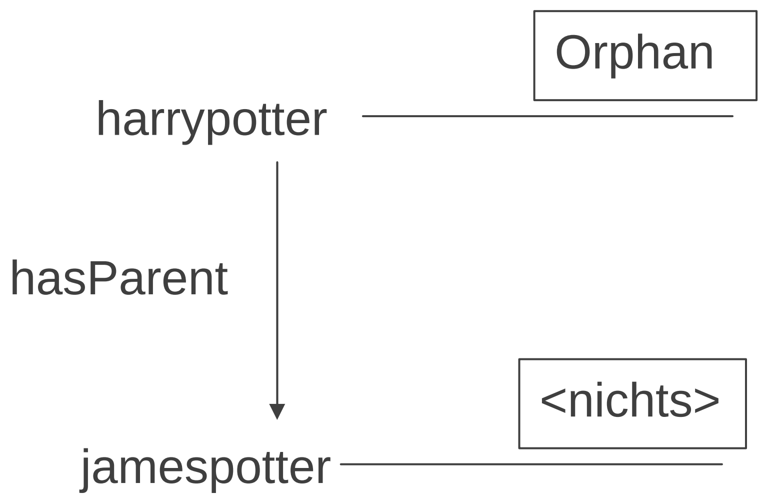
Beispiel: Initialisierung

$\text{Human} \sqsubseteq \exists \text{hasParent.Human}$

$\text{Orphan} \sqsubseteq \text{Human} \sqcap \neg \exists \text{hasParent.Alive}$

Orphan(harrypotter)

hasParent(harrypotter,jamespotter)



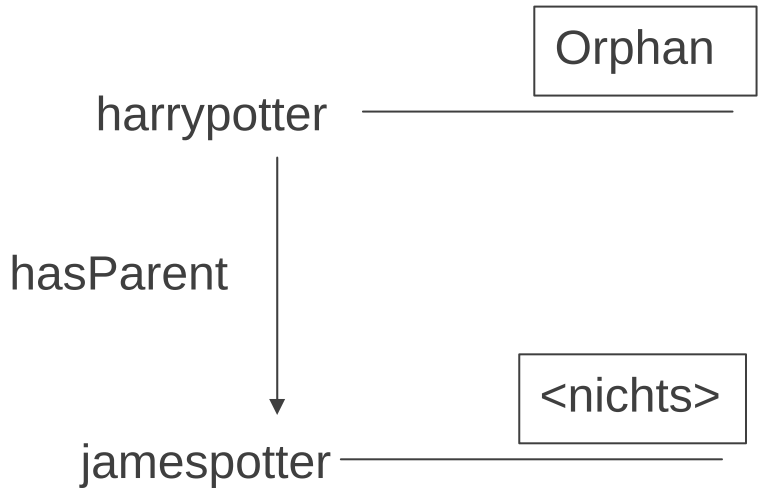
Nicht vergessen: Negationsnormalform!

$\neg\text{Human} \sqcup \exists\text{hasParent.Human}$

$\neg\text{Orphan} \sqcup (\text{Human} \sqcap \forall\text{hasParent.}\neg\text{Alive})$

Orphan(harrypotter)

hasParent(harrypotter,jamespotter)



Konstruktion des Tableaus

- Erweitere das Tableau nichtdeterministisch mit den Regeln auf der folgenden Seite.
- Terminiere, wenn
  - die Beschriftung eines Knotens einen Widerspruch enthält (d.h. Klassen C und $\neg$ C enthält) oder
  - keine der Regeln mehr anwendbar ist
- Falls das Tableau keinen Widerspruch enthält, ist die Wissensbasis erfüllbar.
  (genauer: sind die Regeln durch geeignete Auswahl so anwendbar, dass kein Widerspruch entsteht und keine Regel mehr anwendbar ist, so ist die Wissensbasis erfüllbar)

$\sqcap$: Wenn $C \sqcap D \in L(x)$ und $\{C,D\} \not\subseteq L(x)$, dann setze

$L(x) = L(x) \cup \{C,D\}$

$\sqcup$: Wenn $C \sqcup D \in L(x)$ und $\{C,D\} \cap L(x)=\emptyset$, dann setze

$L(x) = L(x) \cup \{C\}$ oder $L(x) = L(x) \cup \{D\}$

$\exists$: Wenn $\exists R.C \in L(x)$ und x keinen R-Nachfolger y mit $C \in L(y)$ hat, dann

füge einen neuen Knoten y hinzu mit

$L(<x,y>) = \{R\}$ und $L(y) = C$

$\forall$: Wenn $\forall R.C \in L(x)$ und es gibt einen R-Nachfolger y von x mit $C \notin L(y)$, dann setze

$L(y) = L(y) \cup \{C\}$

TBox: Ist C in der TBox und $C \notin L(x)$, dann setze

$L(x) = L(x) \cup \{C\}$

Beispiel

¬Alive(jamespotter)
d.h. füge hinzu: Alive(jamespotter)
und suche Widerspruch

$\neg$Human $\sqcup$ $\exists$hasParent.Human

$\neg$Orphan $\sqcup$ (Human $\sqcap$ $\forall$hasParent.$\neg$Alive)

Orphan(harrypotter)

hasParent(harrypotter,jamespotter)

harrypotter

Orphan
¬Orphan ⊔ (Human ⊓ ∀hasParent.¬Alive)
1. ¬ Orphan (Widerspruch)
2. Human ⊓ ∀hasParent.¬Alive
   Human
   ∀hasParent.¬Alive

hasParent

Alive
2. ¬Alive (Widerspruch)

jamespotter

ALC Tableauverfahren: Inhalt

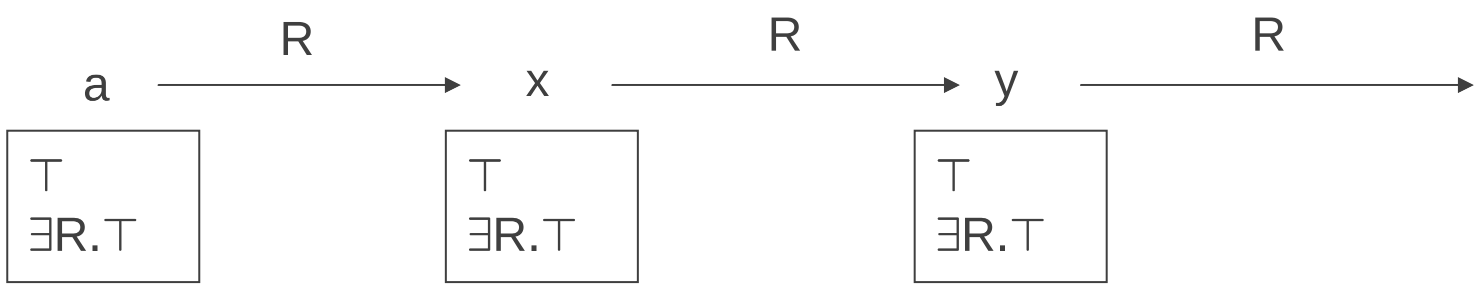
- Transformation in Negationsnormalform
- Naives Tableauverfahren
- **Tableauverfahren mit Blocking**

Es gibt ein Problem mit der Terminierung:

TBox: $\exists R.\top$

ABox: $\top(a_1)$

- Ist offensichtlich erfüllbar:
  Modell M enthält Individuen $a_1^M, a_2^M, \ldots$
  und $R^M(a_i^M, a_{i+1}^M)$ für alle $i \geq 1$
- aber Tableauverfahren terminiert nicht!

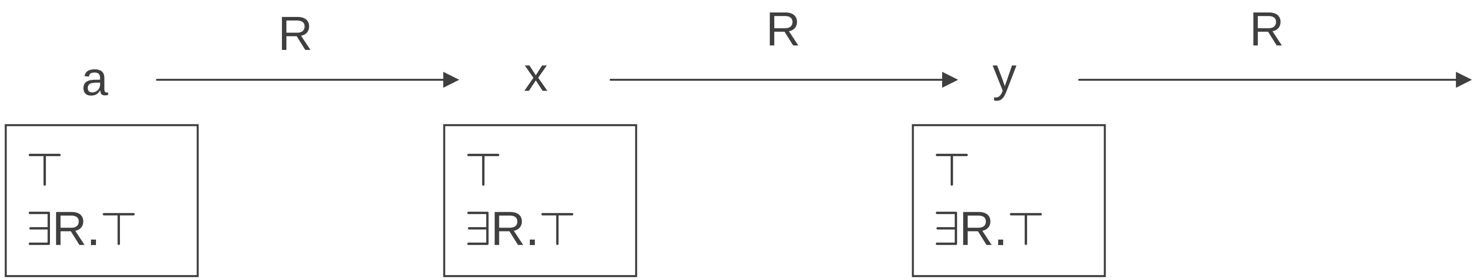


Lösung?

Eigentlich passiert ja nichts neues.

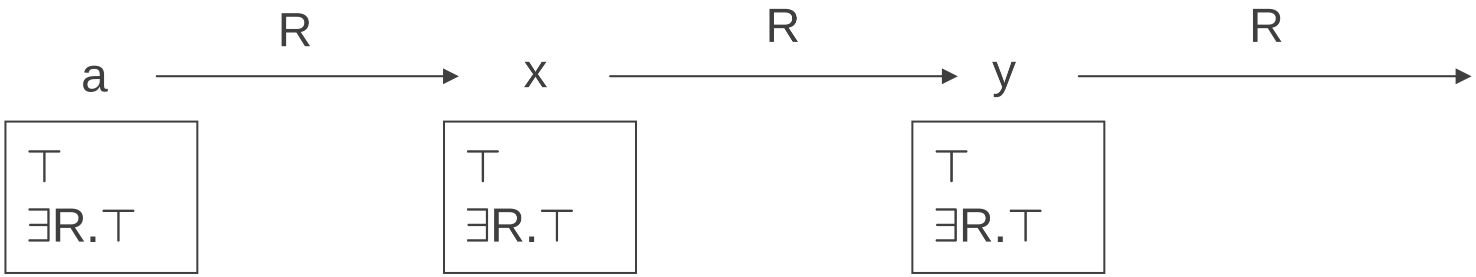
Idee: x muss nicht expandiert werden, da es eigentlich nur eine Kopie von a ist.

$\Rightarrow$ Blocking



Blocking

- x ist *geblockt* (duch y) falls
  - y ein Vorgänger von x ist und L(x) ⊆ L(y) ist
  - oder ein Vorgänger von x geblockt ist.



x ist in diesem Beispiel durch a geblockt!

Konstruktion des Tableaus mit Blocking

- Erweitere das Tableau nichtdeterministisch mit den Regeln auf der folgenden Seite, aber wende eine Regel nur an, falls x nicht geblockt ist!
- Terminiere, wenn
  - die Beschriftung eines Knotens einen Widerspruch enthält (d.h. Klassen C und $\neg$ C enthält) oder
  - keine der Regeln mehr anwendbar ist
- Falls das Tableau keinen Widerspruch enthält, ist die Wissensbasis erfüllbar.
  (genauer: sind die Regeln durch geeignete Auswahl so anwendbar, dass kein Widerspruch entsteht und keine Regel mehr anwendbar ist, so ist die Wissensbasis erfüllbar)

$\sqcap$: Wenn $C \sqcap D \in L(x)$ und $\{C,D\} \not\subseteq L(x)$, dann setze

$L(x) = L(x) \cup \{C,D\}$

$\sqcup$: Wenn $C \sqcup D \in L(x)$ und $\{C,D\} \cap L(x)=\emptyset$, dann setze

$L(x) = L(x) \cup \{C\}$ oder $L(x) = L(x) \cup \{D\}$

$\exists$: Wenn $\exists R.C \in L(x)$ und x keinen R-Nachfolger y mit $C \in L(y)$ hat, dann

füge einen neuen Knoten y hinzu mit

$L(\langle x,y\rangle) = \{R\}$ und $L(y) = C$

$\forall$: Wenn $\forall R.C \in L(x)$ und es gibt einen R-Nachfolger y von x mit $C \notin L(y)$, dann setze

$L(y) = L(y) \cup \{C\}$

TBox: Ist C in der TBox und $C \notin L(x)$, dann setze

$L(x) = L(x) \cup \{C\}$

Regeln nur anwenden, falls x nicht geblockt ist!

Beispiel

- F … Frau
  h … hatMutter
  V … Vogel
  t … Tweety
- Wissensbasis $\{F \sqsubseteq \exists h.F, V(t)\}$
- Wir wollen zeigen, dass Tweety *keine* Frau ist, d.h. dass $\neg F(t)$ logische Konsequenz ist.
- Dies wird uns nicht gelingen.
  D.h. Tweety *kann* eine Frau sein.

Beispiel

Wissensbasis $\{\neg F \sqcup \exists h.F, V(t), F(t)\}$

$$t \xrightarrow{h} x$$

Knoten t:

F
V
$\neg F \sqcup \exists h.F$
1. $\neg F$ (Widerspruch)
2. $\exists h.F$

Knoten x:

2.:
F
geblockt durch t!

keine weitere Expansion möglich. Es kann also kein Widerspruch nachgewiesen werden!

Beispiel nochmal – anders

Wissensbasis $\{\neg F \sqcup \exists h.F, V(t), \neg F(t)\}$

$$t \xrightarrow{h} x \xrightarrow{h} y$$

$\neg F$
V
$\neg F \sqcup \exists h.F$

1. $\neg F$ kann nicht hinzugefügt werden. In diesen Zweig keine Expansion möglich
2. $\exists h.F$

2.:
F
$\neg F \sqcup \exists h.F$
2.1: $\neg F$ (Widerspruch)
2.2: $\exists h.F$

2.2:
F
geblockt durch x

keine weitere Expansion möglich. Wissensbasis ist erfüllbar!

Tableauverfahren für OWL DL

- Die Grundidee ist dieselbe!
- Kompliziertere Blockingregeln müssen verwendet werden.
- Schlechte Unterstützung von Instanzgenerierung.
- Algorithmen skalieren *sehr* schlecht ohne weitere aufwändige Optimierungen.

Tableaux-Beweiser

- Fact
  - http://www.cs.man.ac.uk/~horrocks/FaCT/
  - SHIQ
- Fact++
  - http://owl.man.ac.uk/factplusplus/
  - SHOIQ(D)
- Pellet
  - http://www.mindswap.org/2003/pellet/index.shtml
  - SHOIN(D)
- RacerPro
  - http://www.sts.tu-harburg.de/~r.f.moeller/racer/
  - SHIQ(D)

Literatur

- Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph, York Sure. **Semantic Web – Grundlagen.** Springer 2008. (ISBN 9783540339939)
- Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, Peter Patel-Schneider (eds.), **The Description Logic Handbook.** Cambridge University Press, 2007. (ISBN 9780521781763)
- Ian Horrocks, Ulrike Sattler, Stephan Tobies: Reasoning with Individuals for the Description Logic SHIQ. In: David A. McAllester (ed.), Automated Deduction – CADE-17, 17th International Conference on Automated Deduction, Pittsburgh, PA, USA, June 17-20, 2000, Proceedings. Lecture Notes in Computer Science 1831 Springer 2000, pp. 482-496