Tableaualgorithmen für Beschreibungslogiken

24.06.2008

Pascal Hitzler

Oberseminar Wissensmanagement, AIFB, Universität Karlsruhe (TH)

Übersicht

- **ALC, OWL etc.**
- Grundproblem automatischen Beweisens
- Wichtige Inferenzprobleme
- Tableauverfahren für ALC
- Tableauverfahren für SHIQ

ALC: Syntax

- Folgende Syntaxregeln erzeugen Klassen in ALC. Dabei ist A eine atomare Klasse und R eine Rolle.

  $C,D \to A \mid \top \mid \bot \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \forall R.C \mid \exists R.C$
- Eine *ALC TBox* besteht aus Aussagen der Form $C \sqsubseteq D$ und $C \equiv D$, wobei C, D Klassen sind.
- Eine *ALC ABox* besteht aus Aussagen der Form C(a) und R(a,b), wobei C eine komplexe Klasse, R eine Rolle und a,b Individuen sind.
- Eine *ALC-Wissensbasis* besteht aus einer ABox und einer TBox.

Bezeichner für Beschreibungslogiken

- ALC: Attribute Language with Complement
- S: ALC + Rollentransitivität
- H: Subrollenbeziehung R $\sqsubseteq$ S
- O: abgeschlossene Klassen {o}
- I: inverse Rollen R^-
- N: Zahlenrestriktionen $\leq$n R etc.
  - Q: Qualifizierende Zahlenrestriktionen $\leq$n R.C etc.
- (D): Datentypen
- F: Funktionale Rollen
- R: Rollenkonkatenation hasParent$\circ$hasBrother $\sqsubseteq$ hasUncle

- OWL DL ist SHOIN(D)
- OWL 2 ist SROIQ(D)
- OWL Lite ist SHIF(D)

Übersicht

- ALC, OWL etc.
- **Grundproblem automatischen Beweisens**
- Wichtige Inferenzprobleme
- Tableauverfahren für ALC
- Tableauverfahren für SHIQ

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!

Übersicht

- ALC, OWL etc.
- Grundproblem automatischen Beweisens
- **Wichtige Inferenzprobleme**
- Tableauverfahren für ALC
- Tableauverfahren für SHIQ

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

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

- ALC, OWL etc.
- Grundproblem automatischen Beweisens
- Wichtige Inferenzprobleme
- **Tableauverfahren für ALC**
- Tableauverfahren für SHIQ

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.

$$
\begin{aligned}
NNF(C) &= C, \text{ falls } C \text{ atomar ist}\\
NNF(\neg C) &= \neg C, \text{ falls } C \text{ atomar ist}\\
NNF(\neg\neg C) &= NNF(C)\\
NNF(C \sqcup D) &= NNF(C) \sqcup NNF(D)\\
NNF(C \sqcap D) &= NNF(C) \sqcap NNF(D)\\
NNF(\neg(C \sqcup D)) &= NNF(\neg C) \sqcap NNF(\neg D)\\
NNF(\neg(C \sqcap D)) &= NNF(\neg C) \sqcup NNF(\neg D)\\
NNF(\forall R.C) &= \forall R.NNF(C)\\
NNF(\exists R.C) &= \exists R.NNF(C)\\
NNF(\neg\forall R.C) &= \exists R.NNF(\neg C)\\
NNF(\neg\exists R.C) &= \forall R.NNF(\neg C)
\end{aligned}
$$

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

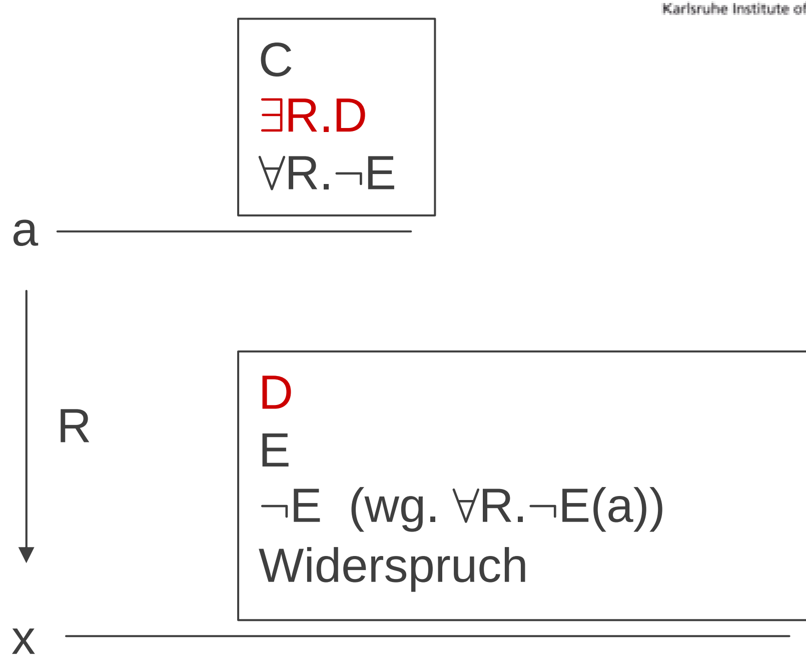
$C(a)$

$C \sqsubseteq \exists R.D$

$D \sqsubseteq E$

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

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



Weiteres Beispiel

$C(a)$

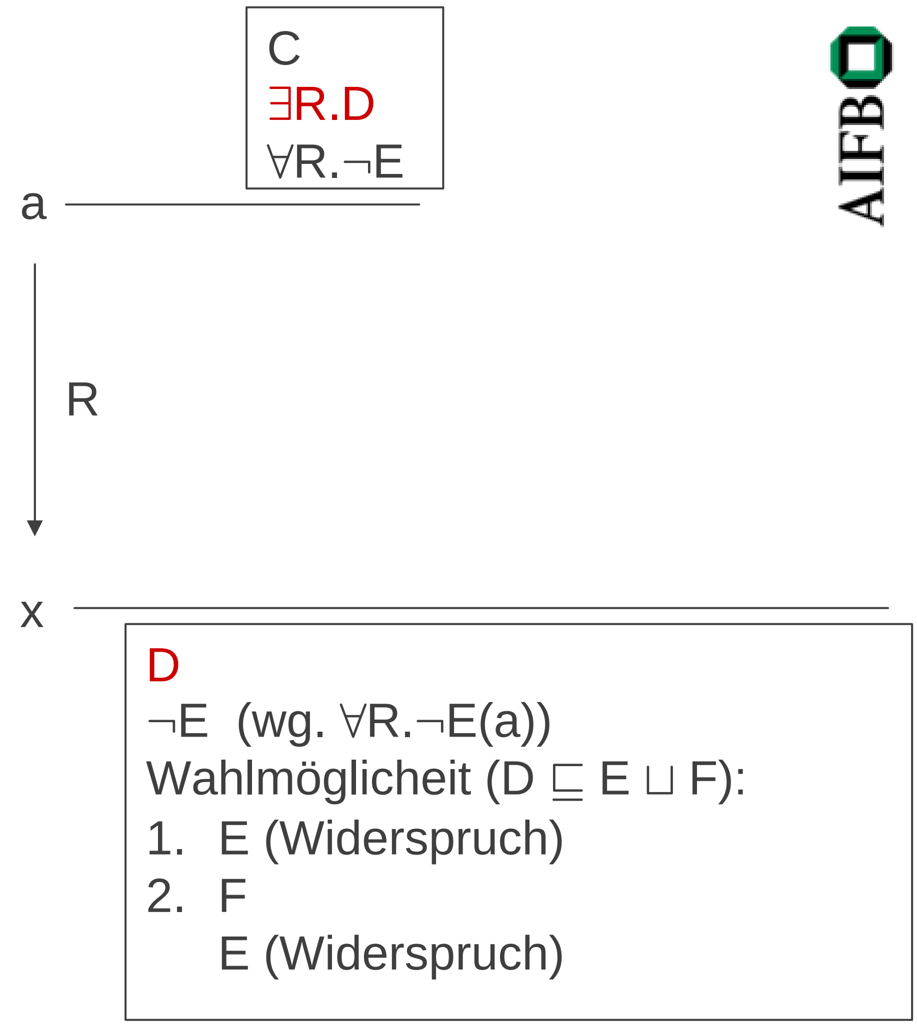
$C \sqsubseteq \exists R.D$

$D \sqsubseteq E \sqcup F$

$F \sqsubseteq E$

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

(füge $\forall R.\neg E(a)$ 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.

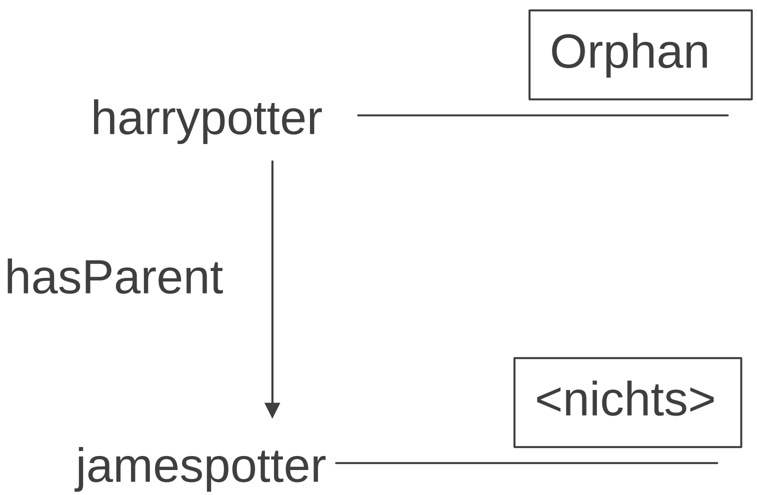
Beispiel: Initialisierung

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

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

Orphan(harrypotter)

hasParent(harrypotter,jamespotter)



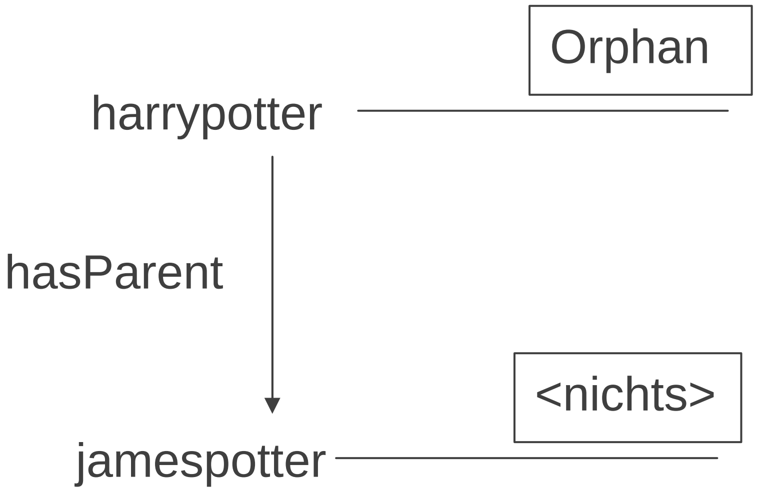
Nicht vergessen: Negationsnormalform!

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

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

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 $\{C,D\} \nsubseteq L(x)$, 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

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

Beispiel

$\neg$Alive(jamespotter)
d.h. füge hinzu: Alive(jamespotter)
und suche Widerspruch

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

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

Orphan(harrypotter)

hasParent(harrypotter,jamespotter)

harrypotter

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

hasParent

Alive
2. $\neg$Alive (Widerspruch)

jamespotter

ALC Tableauverfahren: Inhalt

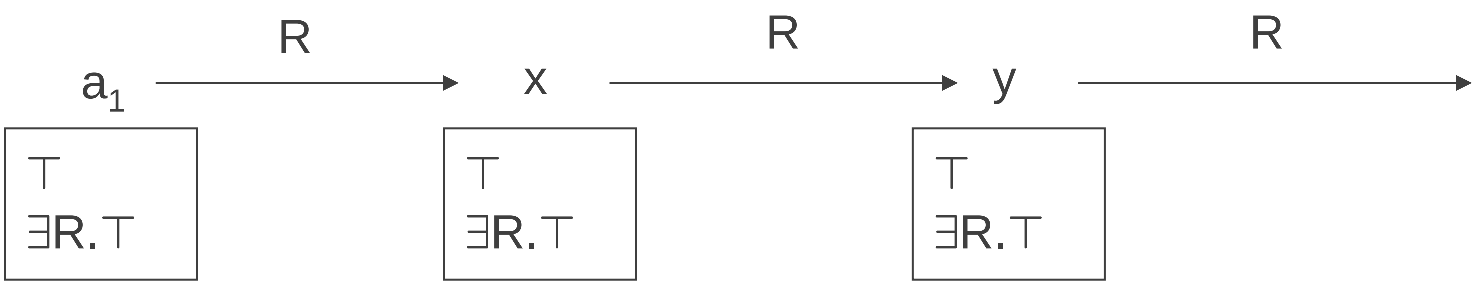
- Transformation in Negationsnormalform
- Naives Tableauverfahren
- **Tableauverfahren mit Blocking**

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!

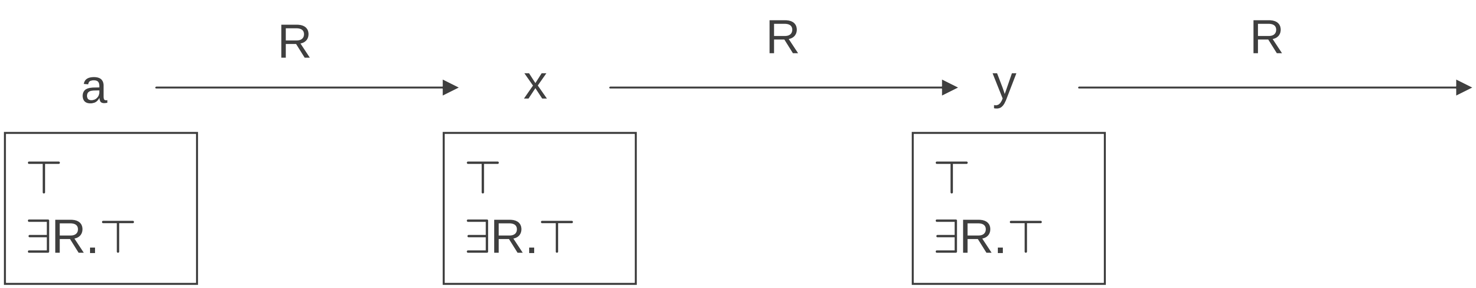


Lösung?

Eigentlich passiert ja nichts neues.

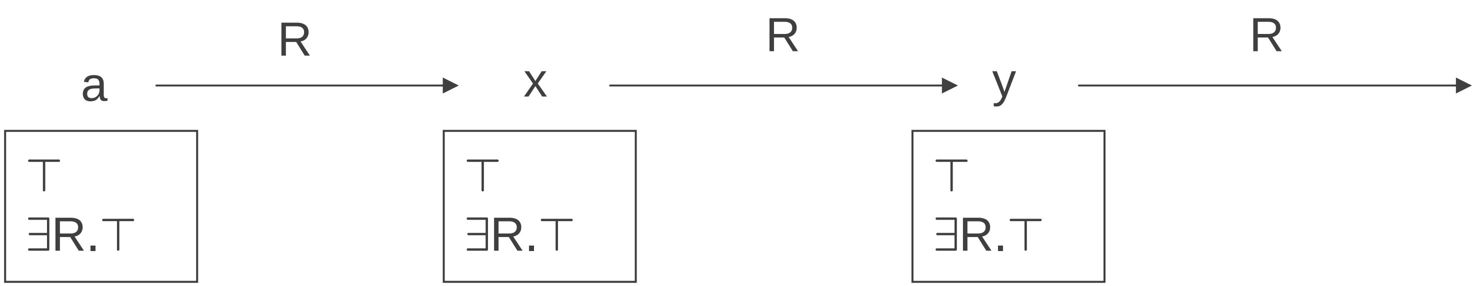
Idee: x muss nicht expandiert werden, da es eigentlich nur eine Kopie von a ist.

$\Rightarrow$ Blocking



Blocking

- x ist *geblockt* (duch y) falls
  - x kein Individuum ist (sondern eine Variable)
  - y ein Vorgänger von x ist und $L(x) \subseteq L(y)$ ist
  - oder ein Vorfahr 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 (0)

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

Beispiel (0)

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

$$t \xrightarrow{h} x$$

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

2.:
M
geblockt durch t!

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

Beispiel (0) nochmal – anders

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

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

t:

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

x:

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

y:

2.2:
M
geblockt durch x

keine weitere Expansion möglich. Wissensbasis ist erfüllbar!

Beispiel (1)

Beweisen Sie, dass aus

$$\text{Professor} \sqsubseteq (\text{Person} \sqcap \text{Unversitaetsangehoeriger}) \sqcup (\text{Person} \sqcap \neg\text{Doktorand})$$

folgt, dass jeder Professor eine Person ist.

Widerspruch zu finden in:
$\neg P \sqcup (E \sqcap U) \sqcup (E \sqcap \neg U)$
$P \sqcap \neg E(x)$

x

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

1. $\neg P$ (Widerspruch)
2. $(E \sqcap U) \sqcup (E \sqcap \neg U)$
   1. $E \sqcap U$
      E (Widerspruch)
   2. $(E \sqcup \neg U)$
      E (Widerspruch)

Beispiel (2)

Beweisen Sie, dass aus

hasChild(john, peter)

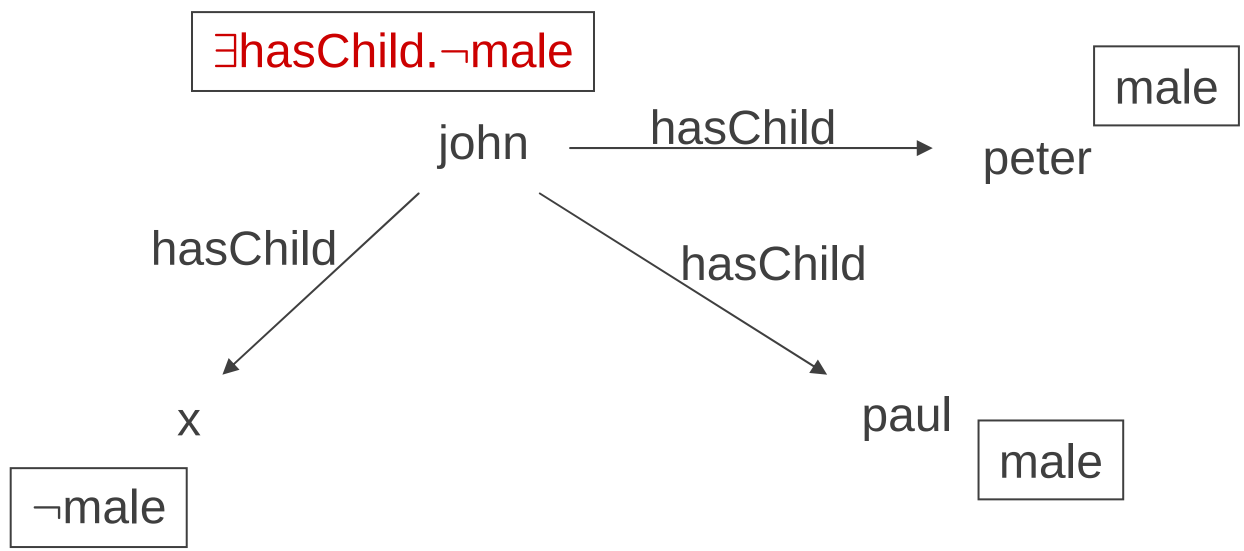
hasChild(john, paul)

male(peter)

male(paul)

die Aussage ∀hasChild.male(john) *nicht* folgt.

$\neg\forall$hasChild.male $\equiv$ $\exists$hasChild.$\neg$male



Beispiel (3)

TBox:
$\neg v \sqcup f$
$\neg p \sqcup v$
$\neg p \sqcup \neg f \sqcup \bot$

Zeigen Sie, dass die Wissensbasis

vogel $\sqsubseteq$ fliegt
pinguin $\sqsubseteq$ vogel
pinguin $\sqcap$ fliegt $\sqsubseteq \bot$
pinguin(tweety)

unerfüllbar ist.

tweety

p
$\neg p \sqcup v$
$\neg v \sqcup f$
$\neg p \sqcup \neg f$

1. $\neg p$ (Widerspruch)
2. v
    1. $\neg v$ (Widerspruch)
    2. f
        1. $\neg p$ (Widerspruch)
        2. $\neg f$ (Widerspruch)

Beispiel (4)

Zeigen Sie, dass aus der Wissensbasis

$C(a)$

$R(a,b)$

$S(a,a)$

$C(c)$

$R(a,c)$

$S(c,b)$

$C \sqsubseteq \forall S.A$

$A \sqsubseteq \exists R.\exists S.A$

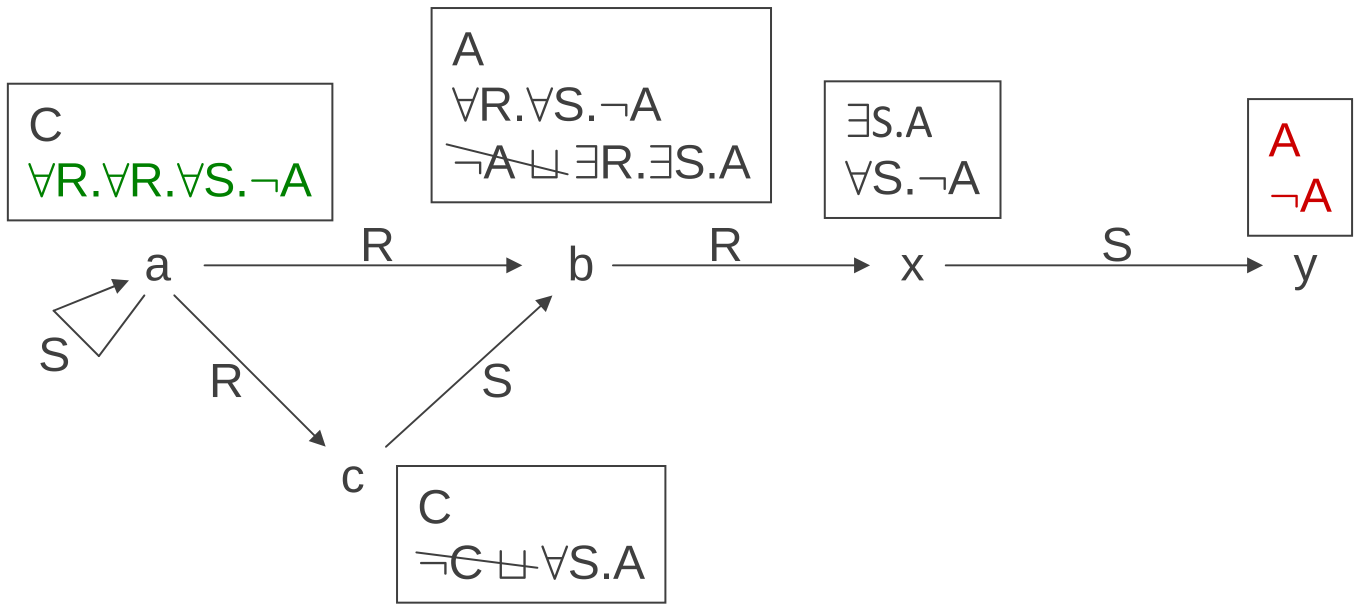
$A \sqsubseteq \exists R.C$

die Aussage $\exists R.\exists R.\exists S.A(a)$ folgt.

(4)

$\neg\exists R.\exists R.\exists S.A \equiv \forall R.\forall R.\forall S.\neg A$

TBox:
$\neg C \sqcup \forall S.A$
$\neg A \sqcup \exists R.\exists S.A$
$\neg A \sqcup \exists R.C$



Übersicht

- ALC, OWL etc.
- Grundproblem automatischen Beweisens
- Wichtige Inferenzprobleme
- Tableauverfahren für ALC
- **Tableauverfahren für SHIQ**

Tableauverfahren für OWL DL

- Die Grundidee ist dieselbe!
- Kompliziertere Blockingregeln müssen verwendet werden.
- Weitere Modifikationen benötigt.

Bezeichner für Beschreibungslogiken

- ALC: Attribute Language with Complement
- S: ALC + Rollentransitivität
- H: Subrollenbeziehung $R \sqsubseteq S$
- O: abgeschlossene Klassen {o}
- I: inverse Rollen R^{-}
- N: Zahlenrestriktionen $\leq n$ R etc.
  - Q: Qualifizierende Zahlenrestriktionen $\leq n$ R.C etc.
- (D): Datentypen
- F: Funktionale Rollen
- R: Rollenkonkatenation hasParent$\circ$hasBrother $\sqsubseteq$ hasUncle

- OWL DL ist SHOIN(D)
- OWL 2 ist SROIQ(D)
- OWL Lite ist SHIF(D)

Negationsnormalform für SHIQ

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.

$NNF(C) = C$, falls C atomar ist

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

$$
\begin{aligned}
NNF(\neg\neg C) &= NNF(C) \\
NNF(C \sqcup D) &= NNF(C) \sqcup NNF(D) \\
NNF(C \sqcap D) &= NNF(C) \sqcap NNF(D) \\
NNF(\neg(C \sqcup D)) &= NNF(\neg C) \sqcap NNF(\neg D) \\
NNF(\neg(C \sqcap D)) &= NNF(\neg C) \sqcup NNF(\neg D) \\
NNF(\forall R.C) &= \forall R.NNF(C) \\
NNF(\exists R.C) &= \exists R.NNF(C) \\
NNF(\neg\forall R.C) &= \exists R.NNF(\neg C) \\
NNF(\neg\exists R.C) &= \forall R.NNF(\neg C) \\
NNF(\leq n\ R.C) &= \leq n\ R.NNF(C) \\
NNF(\geq n\ R.C) &= \geq n\ R.NNF(C) \\
NNF(\neg \leq n\ R.C) &= \geq (n+1) R.NNF(C)
\end{aligned}
$$

$NNF(\neg \geq n\ R.C) = \leq (n-1) R.NNF(C)$, where $\leq(-1)R.C = \perp$

W und NNF(W) sind logisch äquivalent.

SHIQ Tableaus

- Ein Tableau ist ein gerichteter beschrifteter Graph.
  - Knoten sind Individuen oder (neue) Variablennamen
  - Knoten x sind beschriftet mit Mengen L(x) von Klassen
  - jede Kante <x,y> ist beschriftet mit
    - einer Menge L(<x,y>) von Rollennamen R oder inversen Rollennamen R^{-}
  - es gibt zusätzlich zwei Relationen zwischen Knoten, nämlich
    = (für Gleichheit) und
    $\neq$ (für Ungleichheit)

Initialisierung

- Ein Knoten für jedes Individuum in der ABox. Diese Knoten heißen *Wurzelknoten*.
- 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.
- $a \neq b$ für jede entsprechende Aussage in der ABox.
- Die Gleichheitsrelation = ist leer.

Begriffe

- Wir schreiben S^{--} als S.
- Ist $R \in L(<x,y>)$ und $R \sqsubseteq S$ (wobei R,S inverse Rollen sein können), dann ist
  - y ein S-Nachfolger x und
  - x ein S-Vorgänger von y.
- Ist y ein S-Nachfolger oder ein S^{-}-Vorgänger von x, dann ist y ein S-Nachbar von x.
- *Vorfahr* ist der transitive Abschluss von *Vorgänger*

Blocking für SHIQ

- x ist *geblockt* (duch y) falls
  - x,y keine Wurzelknoten sind
  - folgendes gilt: ["x ist direkt geblockt"]
    - – kein Vorfahr von x ist geblockt
    - – es gibt Vorfahren y', x' von x
    - – y ein Nachfolger von y' und x ist ein Nachfolger von x'
    - – L(x) = L(y) und L(x') = L(y')
    - – L(<x',x>) = L(<y',y>)
  - oder folgendes gilt: ["x ist indirekt geblockt"]
    - – ein Vorfahr von x ist geblockt oder
    - – x ist Nachfolger eines y mit L(<y,x>) = ∅

Konstruktion des Tableaus mit Blocking

- Erweitere das Tableau nichtdeterministisch mit den Regeln auf den folgenden Seiten.
- Terminiere, wenn
  - die Beschriftung eines Knotens x einen Widerspruch enthält, d.h.
    - Klassen C und $\neg$C enthält oder
    - eine Klasse $\leq$nR.C enthält und x gleichzeitig (n+1) R-Nachbarn y_i enthält, für die paarweise $y_i \neq y_j$ gilt.
  - 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 x nicht geblockt ist, $\exists R.C \in L(x)$ 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

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

Regeln nur anwenden, falls x nicht indirekt geblockt ist!

$\forall_+$: Wenn $\forall S.C \in L(x)$ und es ein $R \sqsubseteq S$ mit Trans(R) und einen R-Nachbarn y von x mit $\forall R.C \notin L(y)$ gibt, dann setze
$L(y) = L(y) \cup \{\forall R.C\}$.

choose: Wenn $\leq nS.C \in L(x)$ oder $\geq nS.C \in L(x)$ und es einen S-Nachbarn y von x mit $\{C,\neg C\} \sqcap L(y) = \emptyset$ gibt, dann setze
$L(y) = L(y) \sqcup \{C\}$ oder $L(y) = L(y) \sqcup \{\neg C\}$.

$\geq$: Wenn x nicht geblockt ist, $\geq nS.C \in L(x)$ und es keine n S-Nachbarn $y_1,\ldots,y_n$ von x gibt mit $C \in L(y_i)$ und $y_i \neq y_j$ für alle i,j,
dann erzeuge n neue Knoten $y_1,\ldots,y_n$ mit $L(<x,y_i>) = \{S\}$, $L(y_i) = \{C\}$ und $y_i \neq y_j$ für alle i,j.

Regeln nur anwenden, falls x nicht indirekt geblockt ist!

$\leq$: Wenn x nicht indirekt geblockt ist, $\leq$nS.C$\in$ L(x) und es mehr als n S-Nachbarn y_i von x mit C$\in$ L(y_i) gibt, und x zwei S-Nachbarn y,z hat für die nicht y $\neq$ z gilt, so dass y weder ein Wurzelknoten noch ein Vorfahr von z sowie C$\in$ L(y) $\cap$ L(z) ist, dann:

1. setze L(z) = L(z) $\cup$ L(y),
2. falls z ein Vorfahr von x ist, dann setze L(<z,x>) = L(<z,x>) $\cup$ {R^- | R$\in$ L(x,y>)}, sonst setze L(<x,z>) = L(<x,z>) $\cup$ L(<x,y>),
3. setze L(<x,y>) = $\emptyset$ und
4. setze u $\neq$ z für alle u mit u $\neq$ y.

$\leq_r$: Wenn $\leq nS.C \in L(x)$ und es mehr als n S-Nachbarn y_i von x mit $C \in L(y_i)$ gibt, und x zwei S-Nachbarn y,z hat für die nicht $y \neq z$ gilt, so dass y und z Wurzelknoten sind und $C \in L(y) \cap L(z)$ ist, dann:

1. setze $L(z) = L(z) \cup L(y)$,
2. für alle Kanten <y,w>, setze $L(\langle z,w \rangle) = L(\langle z,w \rangle) \cup L(\langle y,w \rangle)$ (die Kante wird erzeugt, falls sie noch nicht existiert),
3. für alle Kanten <w,y>, setze $L(\langle w,z \rangle) = L(\langle w,z \rangle) \cup L(\langle w,y \rangle)$ (die Kante wird erzeugt, falls sie noch nicht existiert),
4. setze $L(y) = \emptyset$ und entferne alle Kanten von und nach y,
5. setze $u \neq z$ für alle u mit $u \neq y$,
6. setze $y = z$.

Beispiel (1): cardinalities

Beweisen Sie, dass aus

$\text{hasChild}(\text{john}, \text{peter})$

$\text{hasChild}(\text{john}, \text{paul})$

$\text{male}(\text{peter})$

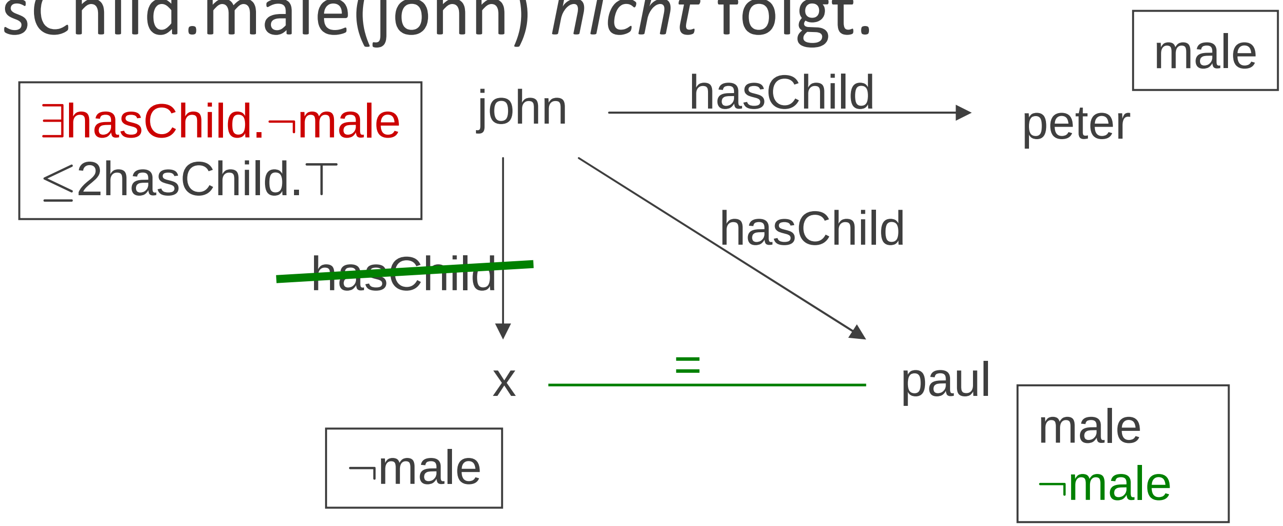
$\text{male}(\text{paul})$

$\leq 2\text{hasChild}.\top(\text{john})$

die Aussage $\forall\text{hasChild}.\text{male}(\text{john})$ *nicht* folgt.

$\boxed{\neg\forall\text{hasChild}.\text{male} \equiv \exists\text{hasChild}.\neg\text{male}}$

now apply $\leq$



Beispiel (1): cardinalities

Beweisen Sie, dass aus

$hasChild(john, peter)$

$hasChild(john, paul)$

$male(peter)$

$male(paul)$

$\leq 2 hasChild.\top(john)$

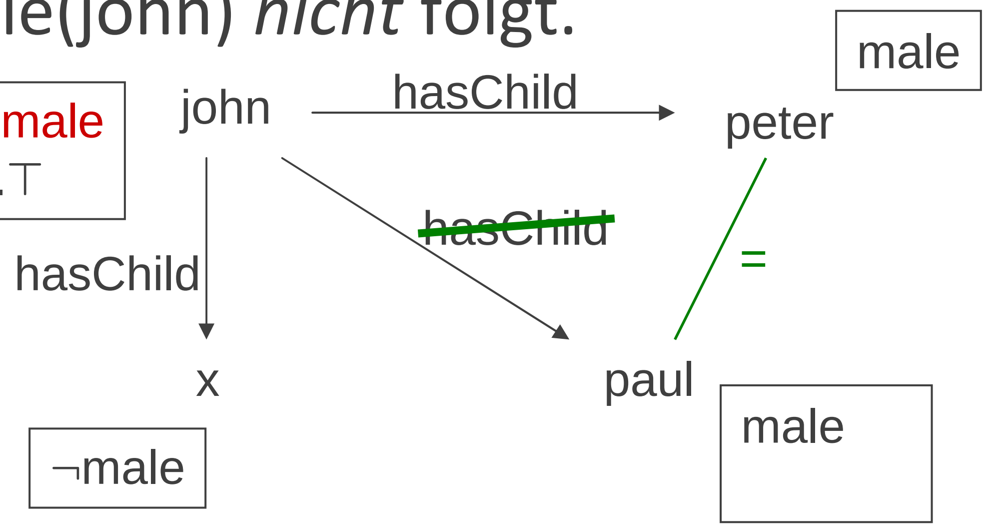
die Aussage $\forall hasChild.male(john)$ *nicht* folgt.

$\neg\forall hasChild.male \equiv \exists hasChild.\neg male$

backtracking!

now apply $\leq$

$\exists hasChild.\neg male$
$\leq 2 hasChild.\top$



Beispiel (1): cardinalities – nochmal

Beweisen Sie, dass aus

$\quad$ hasChild(john, peter)

$\quad$ hasChild(john, paul)

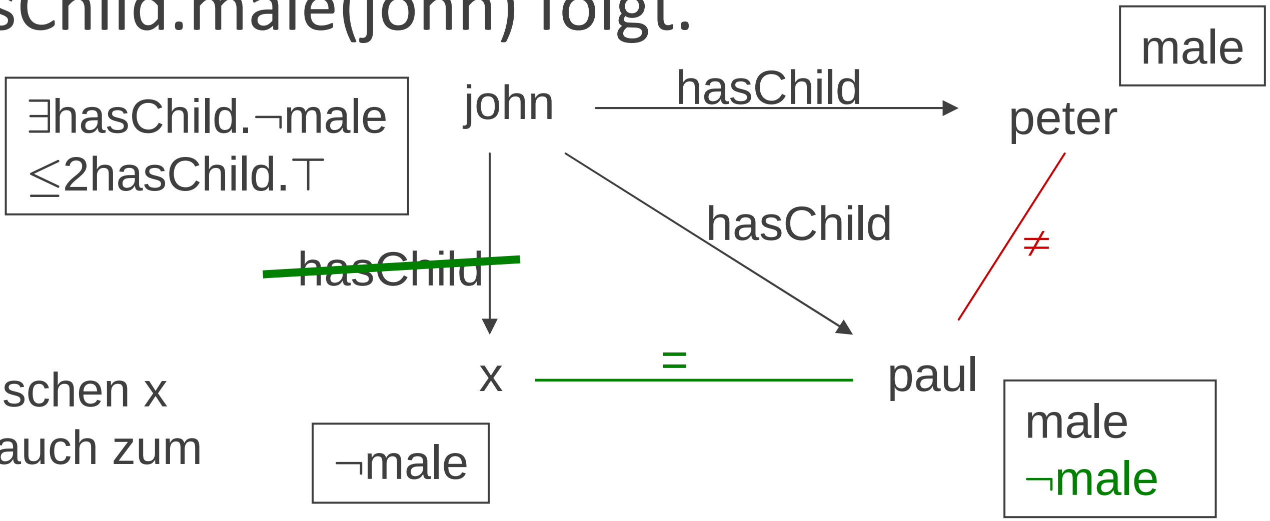
$\quad$ male(peter)

$\quad$ male(paul)

$\quad$ $\leq 2\text{hasChild}.\top(\text{john})$ und peter $\neq$ paul

die Aussage $\forall\text{hasChild.male(john)}$ folgt.

$\boxed{\neg\forall\text{hasChild.male} \equiv \exists\text{hasChild.}\neg\text{male}}$



now apply $\leq$

backtracking nur noch zwischen x und peter möglich – führt auch zum Widerspruch

Beispiel (2): cardinalities

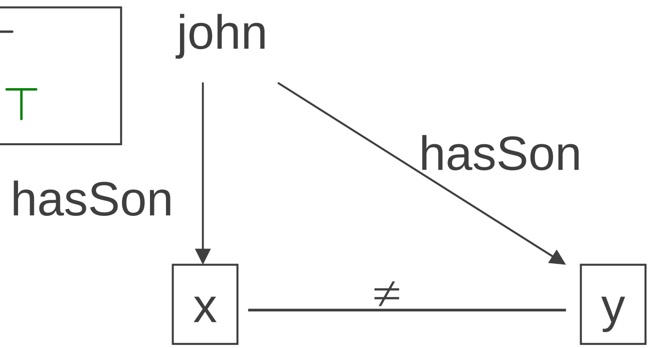
Beweisen Sie, dass aus

$\neg\geq 2\text{hasSon}.\top \equiv \leq 1\text{hasChild}.\top$

$\geq 2\text{hasSon}.\top(\text{john})$ $\qquad$ $\text{hasSon} \sqsubseteq \text{hasChild}$

die Aussage $\geq 2\text{hasChild}.\top(\text{john})$ folgt.

$\geq 2\text{hasSon}.\top$
$\leq 1\text{hasChild}.\top$



hasSon-Nachbarn sind auch hasChild-Nachbarn,
Tableau terminiert mit Widerspruch

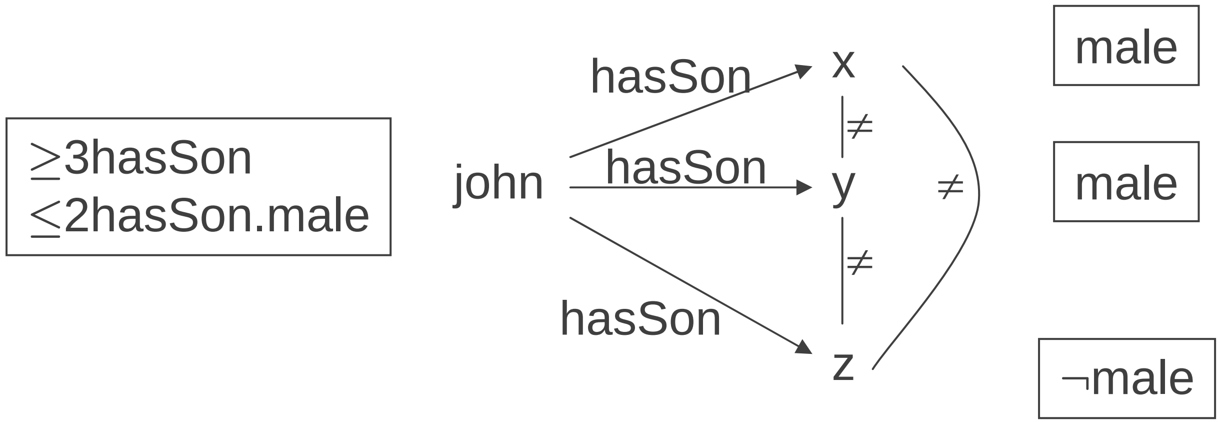
Beispiel (3): choose

$\geq$3hasSon(john)

$\leq$2hasSon.male(john)

Ist dies widersprüchlich?

nein, da folgendes Tableau abgeschlossen ist



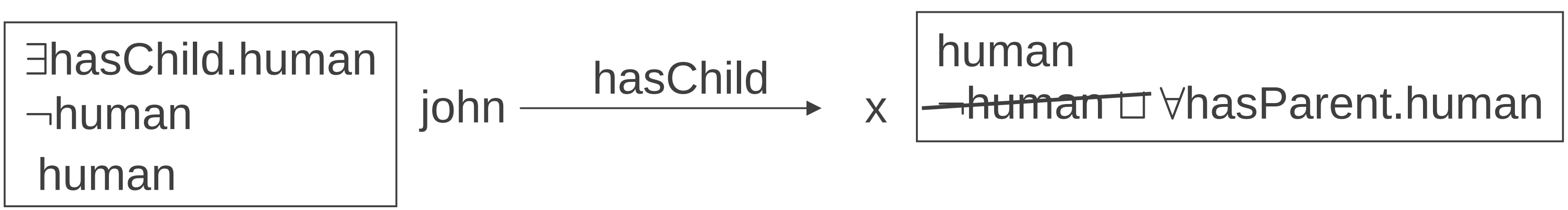
Beispiel (4): inverse roles

$\exists$hasChild.human(john)

human $\sqsubseteq$ $\forall$hasParent.human

hasChild $\sqsubseteq$ hasParent$^-$

zu zeigen: human(john)



john ist hP$^-$-Vorgänger von x, also hP-Nachbar von x

Beispiel (5): Transitivität und Blocking

$\text{human} \sqsubseteq \exists \text{hasFather}.\top$

$\text{human} \sqsubseteq \forall \text{hasAncestor}.\text{human}$

$\text{hasFather} \sqsubseteq \text{hasAncestor}$ $\quad$ $\text{Trans}(\text{hasAncestor})$

$\text{human}(\text{john})$

Frage:

gilt $\leq 1 \text{hasFather}.\top(\text{john})$?

Negation davon: $\geq 2 \text{hasFather}.\top(\text{john})$

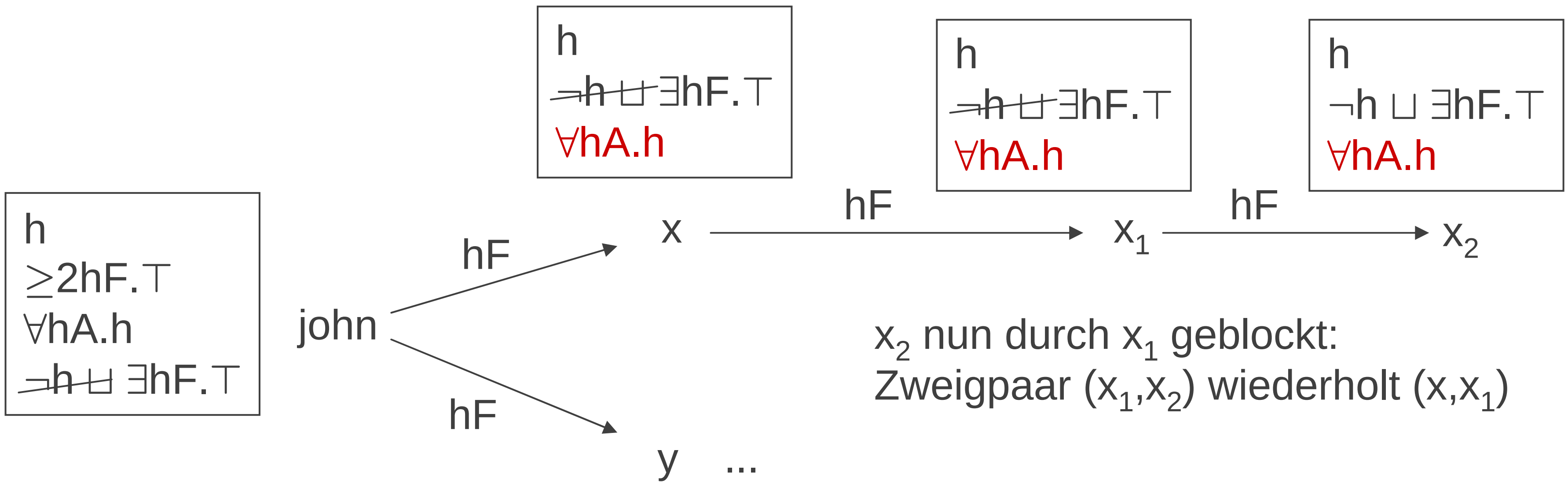
Beispiel (5): Transitivität und Blocking

$\text{human} \sqsubseteq \exists \text{hasFather}.\top$

$\text{hasFather} \sqsubseteq \text{hasAncestor}$ $\qquad$ $\text{Trans(hasAncestor)}$

$\forall \text{hasAncestor.human(john)}$

human(john) $\qquad$ $\geq 2\text{hasFather}.\top\text{(john)}$



x_2 nun durch x_1 geblockt:
Zweigpaar (x_1,x_2) wiederholt (x,x_1)

analog zum obigen Zweig

Beispiel (6): Blocking durch Paare

$\neg C \sqcap (\leq 1F) \sqcap \exists F^{-}.D \sqcap \forall R^{-}.(\exists F^{-}.D)$, where
$D = C \sqcap (\leq 1F) \sqcap \exists F.\neg C$, Trans(R), and $F \sqsubseteq R$,
is not satisfiable.

$$x \xrightarrow{F^{-}} y \xrightarrow{F^{-}} z$$

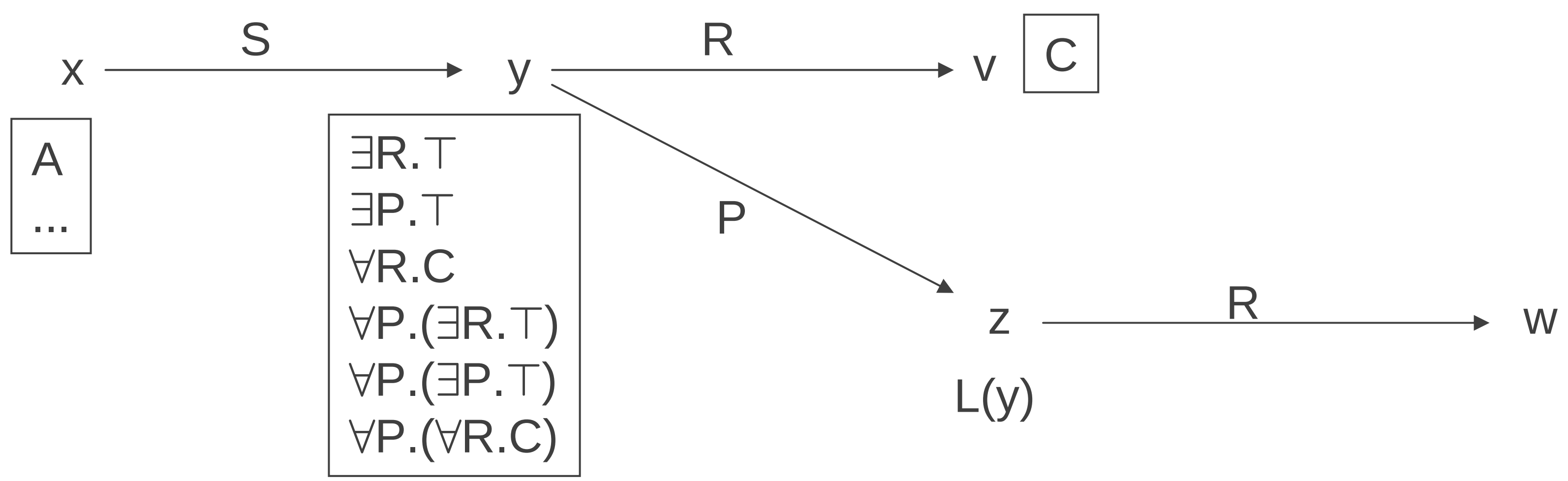
x	y	z
$\neg C$	D	D
$\leq 1F$	$\exists F^{-}.D$	$\exists F^{-}.D$
$\exists F^{-}.D$	$\forall R^{-}.(\exists F^{-}.D)$	$\forall R^{-}.(\exists F^{-}.D)$
$\forall R^{-}.(\exists F^{-}.D)$	C	C
	$\leq 1F$	$\leq 1F$
	$\exists F.\neg C$	$\exists F.\neg C$

Ohne Blocking durch Paare wäre z geblockt, sollte es aber nicht sein: Expansion von $\exists F.\neg C$ gibt y das label $\neg C$ wie benötigt.

Beispiel (7): Dynamisches Blocking

$A \sqcap \exists S.(\exists R.\top \sqcap \exists P.\top \sqcap \forall R.C \sqcap \forall P.(\exists R.\top) \sqcap \forall P.(\forall R.C) \sqcap \forall P.(\exists P.\top))$
with $C = \forall R^{-}.(\forall P^{-}.(\forall S^{-}.\neg A))$ and Trans(P), is not satisfiable.

Teil des Tableaus:



At this stage, z would be blocked by y (assuming the presence of another pair). However, when C from v is expanded, z becomes unblocked, which is necessary in order to label w with C which in turn labels x with $\neg A$, yielding the required contradiction.

Tableaux-Beweiser

- Fact++
  - http://owl.man.ac.uk/factplusplus/

- Pellet
  - http://www.mindswap.org/2003/pellet/index.shtml

- RacerPro
  - http://www.sts.tu-harburg.de/~r.f.moeller/racer/

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.
- Ian Horrocks and Ulrike Sattler. A Description Logic with Transitive and Inverse Roles and Role Hierarchies. *J. of Logic and Computation*, 9(3):385-410, 1999.
- Some slides by Ulrike Sattler, and some talking with her ...
- Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph. **Foundations of Semantic Web Technologies.** CRC Press, 2009(?), to appear.