OWL - Semantik und Reasoning

Foliensatz: Markus Krötzsch, Sebastian Rudolph

Webbasierte Informationssysteme
15. November 2010

Vorlesungsübersicht

Outline

Beschreibungslogiken

- engl.: description logics (DLs)
- Fragmente von FOL
- meist entscheidbar
- vergleichsweise ausdrucksstark
- entwickelt aus semantischen Netzwerken
- enge Verwandtschaft mit Modallogiken
- W3C Standard OWL DL basiert auf der Beschreibungslogik SHOIN(D)
- Wir besprechen zunächst ALC (Basis für komplexere DLs)

DLs - Aufbau

- DLs sind eine Familie logikbasierter Formalismen zur Wissensrepräsentation
- Spezielle Sprachen v.a. charakterisiert durch:
  1. Konstruktoren für komplexe Klassen und Rollen aus einfacheren.
  2. Menge von Axiomen um Fakten über Klassen, Rollen und Individuen auszudrücken.
- ALC ist die kleinste DL, die aussagenlogisch abgeschlossen ist
  1. Konjunktion, Disjunktion, Negation sind Konstruktoren, geschrieben $\sqcap, \sqcup, \neg$.
  2. Quantoren schränken Rollenbereiche ein, z.B.:
- $\text{Man} \sqcap \exists \text{hasChild.Female} \sqcap \exists \text{hasChild.Male} \sqcap \forall \text{hasChild.(Rich} \sqcup \text{Happy)}$

DLs - weitere Sprachmittel

Andere Konstruktoren sind z.B.

- number restrictions (Kardinalitätseinschränkungen) für Rollen: ≥ 3 hasChild, ≤ 1 hasMother
- qualified number restrictions (klassenspezifische Kardinalitätseinschränkungen) für Rollen: ≥ 2 hasChild.Female, ≤ 1 hasParent.Male
- abgeschlossene Klassen (Definition durch Aufzählung): {Italy, France, Spain}
- konkrete Bereiche (Datentypen): hasAge.(≥ 21)
- inverse Rollen: hasChild$^-$
- transitive Rollen: Trans(hasAncestor)
- Rollenverknüpfung: Sibling $\circ$ Parent

Outline

ALC - Grundbausteine

Grundbausteine:

- Klassennamen (auch als Konzepte bezeichnet)
- Rollennamen
- Individuennamen

Angabe von Klassen- und Rolleninstanzen:

Professor(RudiStuder)

- Individuum RudiStuder ist in Klasse Professor

Zugehoerigkeit(RudiStuder,AIFB)

- RudiStuder ist dem AIFB zugehörig

ALC - Subklassenbeziehungen

Professor $\sqsubseteq$ Fakultätsmitglied

- Jeder Professor ist ein Fakultätsmitglied.
- entspricht $\forall$x(Professor(x) $\rightarrow$ Fakultaetsmitglied(x))
- entspricht owl:subClassOf

Professor $\equiv$ Fakultaetsmitglied

- Die Fakultätsmitglieder sind genau die Professoren.
- entspricht $\forall$x(Professor(x) $\leftrightarrow$ Fakultaetsmitglied(x))
- entspricht owl:equivalentClass

ALC - komplexe Klassen

Konjunktion $\sqcap$ entspricht owl:intersectionOf Disjunktion $\sqcup$ entspricht owl:unionOf Negation $\neg$ entspricht owl:complementOf

Beispiel:

- Professor $\sqsubseteq$ (Person $\sqcap$ Unversitaetsangehoeriger) $\sqcup$ (Person $\sqcap$ $\neg$Doktorand)
- $\forall$x(Professor(x) $\rightarrow$ [(Person(x) $\wedge$ Universitaetsangehoeriger(x)) $\vee$ (Person(x) $\wedge$ $\neg$Doktorand(x))]

ALC - Quantoren auf Rollen

Pruefung $\sqsubseteq \forall$hatPruefer.Professor

- Jede Prüfung hat nur Professoren als Prüfer.
- $\forall x(\text{Pruefung}(x) \rightarrow \forall y(\text{hatPruefer}(x,y) \rightarrow \text{Professor}(y)))$
- entspricht owl:allValuesFrom

Pruefung $\sqsubseteq \exists$hatPruefer.Person

- Jede Prüfung hat mindestens einen Prüfer.
- $\forall x(\text{Pruefung}(x) \rightarrow \exists y(\text{hatPruefer}(x,y) \wedge \text{Person}(y)))$
- entspricht owl:someValuesFrom

Weitere OWL-Konstrukte in ALC

- owl:nothing: $\bot \sqsubseteq C \sqcap \neg C$
- owl:thing: $\top \sqsubseteq C \sqcup \neg C$
- owl:disjointWith: $C \sqcap D \sqsubseteq \bot$
  (gleichbedeutend:) $C \sqsubseteq \neg D$
- rdfs:range: $\top \sqsubseteq \forall R.C$
- rdfs:domain: $\exists R.\top \sqsubseteq C$

ALC - Formale Syntax

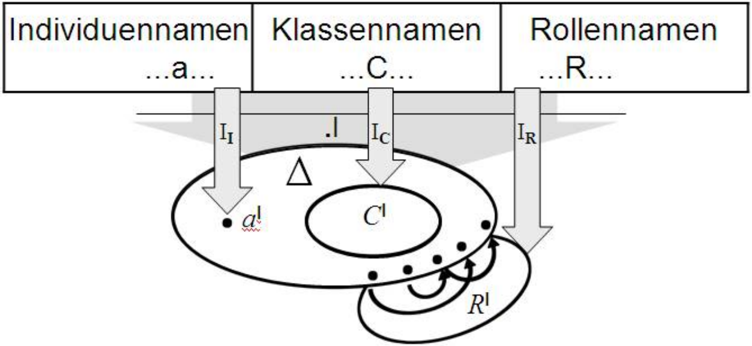
- Folgende Syntaxregeln erzeugen Klassen in ALC. Dabei ist A eine atomare Klasse und R eine Rolle.
  $C, D \rightarrow A \mid \top \mid \bot \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \forall R.C \mid \exists R.C$
- Eine ALC-TBox besteht aus Aussagen der Form $C \sqsubseteq D$ und $C \equiv D$, 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.

ALC - Semantik (Interpretationen)

- Wir definieren modelltheoretische Semantik fur ALC (d.h. Folgerung wird uber Interpretationen definiert)
- Eine Interpretation $I = (\Delta^I, .^I)$ besteht aus
  - einer Menge Δ^I, genannt Domäne und
  - einer Funktion $.^I$, die abbildet von
  - Individuennamen a auf Domänenelemente $a^I \in \Delta^I$
  - Klassennamen C auf Mengen von Domänenelementen $C^I \subseteq \Delta^I$
  - Rollennamen R auf Mengen von Paaren von Domänenelementen $R^I \subseteq \Delta^I \times \Delta^I$

ALC - Semantik (Interpretationen)

schematisch:



- wird auf komplexe Klassen erweitert:
  - $\top^I = \Delta^I$ und $\bot^I = \emptyset$
  - $(C \sqcap D)^I = C^I \cap D^I$ und $(C \sqcup D)^I = C^I \cup D^I$
  - $(\neg C)^I = \Delta^I\ C^I$
  - $(\forall R.C)^I = \{x \in \Delta^I | \forall y \in \Delta^I : (x, y) \in R^I \rightarrow y \in C^I\}$
  - $(\exists R.C)^I = \{x \in \Delta^I | \exists y \in \Delta^I : (x, y) \in R^I$ mit $y \in C^I\}$

- ...und schliesslich auf Axiome:
  - $I \models C(a)$ gilt, wenn $a^I \in C^I$
  - $I \models R(a, b)$ gilt, wenn $(a^I, b^I) \in R^I$
  - $I \models C \sqsubseteq D$ gilt, wenn $C^I \subseteq D^I$
  - $I \models C \equiv D$ gilt, wenn $C^I = D^I$

ALC - alternative Semantik

- Übersetzung von TBox-Aussagen in die Pradikatenlogik mittels der Abbildung π.
- Dabei sind C,D komplexe Klassen, R eine Rolle und A eine atomare Klasse.

$$
\begin{aligned}
\pi(C \sqsubseteq D) &= (\forall x)(\pi_x(C) \rightarrow \pi_x(D)) \\
\pi(C \equiv D) &= (\forall x)(\pi_x(C) \leftrightarrow \pi_x(D)) \\
\pi_x(A) &= A(x) \\
\pi_x(\neg C) &= \neg \pi_x(C) \\
\pi_x(C \sqcap D) &= \pi_x(C) \wedge \pi_x(D) \\
\pi_x(C \sqcup D) &= \pi_x(C) \vee \pi_x(D) \\
\pi_x(\forall R.C) &= (\forall y)(R(x,y) \rightarrow \pi_y(C)) \\
\pi_x(\exists R.C) &= (\exists y)(R(x,y) \wedge \pi_y(C)) \\
\pi_y(A) &= A(x) \\
\pi_y(\neg C) &= \neg \pi_y(C) \\
\pi_y(C \sqcap D) &= \pi_y(C) \wedge \pi_y(D) \\
\pi_y(C \sqcup D) &= \pi_y(C) \vee \pi_y(D) \\
\pi_y(\forall R.C) &= (\forall x)(R(y,x) \rightarrow \pi_x(C)) \\
\pi_y(\exists R.C) &= (\exists x)(R(y,x) \wedge \pi_x(C))
\end{aligned}
$$

DL-Wissensbasen

DL Wissensbasen bestehen aus 2 Teilen:

- TBox: Axiome, die die Struktur der zu modellierenden Domäne beschreiben (konzeptionelles Schema):
  - $\text{HappyFather} \equiv \text{Man} \sqcap \exists \text{hasChild.Female} \sqcap \ldots$
  - $\text{Elephant} \sqsubseteq \text{Animal} \sqcap \text{Large} \sqcap \text{Grey}$
  - transitive(hasAncestor)
- Abox: Axiome, die konkrete Situationen (Daten) beschreiben:
  - HappyFather(John)
  - hasChild(John, Mary)

Einfaches Beispiel

- **Terminologisches Wissen (TBox):**
  $\text{Human} \sqsubseteq \exists \text{hasParent}.\text{Human}$
  $\text{Orphan} \equiv \text{Human} \sqcap \neg\exists \text{hasParent}.\text{Alive}$

- **Wissen um Individuen (ABox):**
  Orphan(harrypotter)
  hasParent(harrypotter,jamespotter)

- Semantik und logische Konsequenzen klar, da übersetzbar nach FOL.

OWL und ALC

Folgende OWL DL Sprachelemente sind in ALC repräsentierbar:

- Klassen, Rollen, Individuen
- Klassenzugehörigkeit, Rolleninstanzen
- owl:Thing und owl:Nothing
- Klasseninklusion, -äquivalenz, -disjunktheit
- owl:intersectionOf, owl:unionOf
- owl:complementOf
- owl:allValuesFrom, owl:someValuesFrom
- rdfs:range und rdfs:domain

Outline

OWL als SHOIN(D) - Individuen

owl:sameAs

- gibt an dass zwei Individuennamen dasselbe Domanenelement bezeichnen
- DL: a=b
- FOL: Erweiterung durch Gleichheitspradikat

owl:differentFrom

- gibt an dass zwei Individuennamen unterschiedliche Domanenelemente bezeichnen
- DL: $a \neq b$
- FOL: $\neg(a = b)$

OWL als SHOIN(D) - abgeschlossene Klassen

Abgeschlossene Klassen

- owl:oneOf
  - definiert eine Klasse durch vollständige Aufzählung ihrer Instanzen
  - DL: $C \equiv \{a,b,c\}$
  - FOL: $\forall x\ (C(x) \leftrightarrow (x{=}a \vee x{=}b \vee x{=}c))$
- owl:hasValue
  - erzwingt Rolle zu einem bestimmten Individuum
  - darstellbar mittels owl:someValuesFrom und owl:oneOf

OWL als SHOIN(D) - Kardinalität

Zahlenrestriktionen mittels Gleichheitsprädikat

```
<owl:Class rdf:about="#Pruefung">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hatPruefer"/>
      <owl:maxCardinality rdf:datatype="&xsd;nonNegativeInteger">2</owl:maxcardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Eine Prüfung kann höchstens zwei Prüfer haben.

- DL: Pruefung $\sqsubseteq \leq$ 2 hatPruefer
- In FOL: (P...Pruefung, h...hatPruefer)
  $\forall$x (P(x) $\rightarrow \neg\exists x_1 \exists x_2 \exists x_3$ (x1 $\neq$ x2 $\wedge$ x2 $\neq$ x3 $\wedge$ x1 $\neq$ x3 $\wedge$ h(x,x_1) $\wedge$ h(x,x_2) $\wedge$ h(x,x_3)))

Entsprechend für die anderen Zahlenrestriktionen

OWL als SHOIN(D) - Rollen

rdfs:subPropertyOf

- spezifiziert Unterrolle-Oberrolle-Beziehung
- DL: $R \sqsubseteq S$
- FOL: $\forall x \forall y\ (R(x,y) \rightarrow S(x,y))$

Rollenäquivalenz: entsprechend
inverse Rollen: $R \equiv S^{-}$

- Konstruktor für Rollen zur Bildung der Inversen
- FOL: $\forall x \forall y\ (R(x,y) \leftrightarrow S(y,x))$

transitive Rollen: Trans(R)

- gibt an, dass eine Rolle transitiv ist
- FOL: $\forall x \forall y \forall z\ ((R(x,y) \wedge R(y,z)) \rightarrow R(x,z))$

OWL als SHOIN(D) - Rollen

Symmetrie: R $\equiv$ R-

- ausdrückbar als Rollenäquivalenz mit der Inversen

Funktionalitat: $\top \sqsubseteq \leq 1$ R

- ausdrückbar durch Klasseninklusion und Kardinalitätsrestriktion

Inverse Funktionalitat: $\top \sqsubseteq \leq 1$ R^{-}

- ausdrückbar wie Funktionalität zuzüglich inverser Rolle

OWL als SHOIN(D) - Überblick

Erlaubt sind:

- ALC
- Gleichheit und Ungleichheit zwischen Individuen
- Abgeschlossene Klassen
- Zahlenrestriktionen
- Subrollen und Rollenäquivalenz
- Inverse und transitive Rollen
- Datentypen

DLs - Nomenklatur

- ALC: Attribute Language with Complement
- S: ALC + Rollentransitivitat
- H: Subrollenbeziehung
- O: abgeschlossene Klassen
- I: inverse Rollen
- N: Zahlenrestriktionen $\leq n$ R etc.
- Q: Qualifizierende Zahlenrestriktionen $\leq n$ R.C etc.
- (D): Datentypen
- F: Funktionale Rollen

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

DL-Syntax - Übersicht

	Concepts	
$\mathcal{ALC}$	Atomic	A, B
$\mathcal{ALC}$	Not	$\neg$C
$\mathcal{ALC}$	And	C $\sqcap$ D
$\mathcal{ALC}$	Or	C $\sqcup$ D
$\mathcal{ALC}$	Exists	$\exists$R.C
$\mathcal{ALC}$	For all	$\forall$R.C
$\mathcal{Q}$ ($\mathcal{N}$)	At least	$\geq$n R.C ($\geq$n R)
$\mathcal{Q}$ ($\mathcal{N}$)	At most	$\leq$n R.C ($\leq$n R)
$\mathcal{O}$	Closed class	$\{i_1, \ldots, i_n\}$

	Roles	
	Atomic	R
$\mathcal{I}$	Inverse	R^-

Ontology (=Knowledge Base)

Concept Axioms (TBox)	
Subclass	C $\sqsubseteq$ D
Equivalent	C $\equiv$ D

	Role Axioms (RBox)	
$\mathcal{H}$	Subrole	R $\sqsubseteq$ S
$\mathcal{S}$	Transitivity	Trans(S)

Assertional Axioms (ABox)	
Instance	C(a)
Role	R(a,b)
Same	a = b
Different	a $\neq$ b

S = ALC + Transitivity

OWL DL = SHOIN(D) (D: concrete domain)

DL-Syntax - Klassenkonstruktoren

Constructor	DL Syntax	Example	FOL Syntax
intersectionOf	$C_1 \sqcap \ldots \sqcap C_n$	Human ⊓ Male	$C_1(x) \wedge \ldots \wedge C_n(x)$
unionOf	$C_1 \sqcup \ldots \sqcup C_n$	Doctor ⊔ Lawyer	$C_1(x) \vee \ldots \vee C_n(x)$
complementOf	$\neg C$	¬Male	$\neg C(x)$
oneOf	$\{x_1\} \sqcup \ldots \sqcup \{x_n\}$	{john} ⊔ {mary}	$x = x_1 \vee \ldots \vee x = x_n$
allValuesFrom	$\forall P.C$	∀hasChild.Doctor	$\forall y.P(x,y) \rightarrow C(y)$
someValuesFrom	$\exists P.C$	∃hasChild.Lawyer	$\exists y.P(x,y) \wedge C(y)$
maxCardinality	$\leqslant nP$	⩽1hasChild	$\exists^{\leqslant n} y.P(x,y)$
minCardinality	$\geqslant nP$	⩾2hasChild	$\exists^{\geqslant n} y.P(x,y)$

Beliebig komplexes Schachteln von Konstruktoren erlaubt:
Person ⊓ ∀hasChild.(Doctor ⊔ ∃hasChild.Doctor)

DL-Syntax - Axiome

Axiom	DL Syntax	Example
subClassOf	$C_1 \sqsubseteq C_2$	Human $\sqsubseteq$ Animal $\sqcap$ Biped
equivalentClass	$C_1 \equiv C_2$	Man $\equiv$ Human $\sqcap$ Male
disjointWith	$C_1 \sqsubseteq \neg C_2$	Male $\sqsubseteq \neg$Female
sameIndividualAs	$\{x_1\} \equiv \{x_2\}$	{President_Bush} $\equiv$ {G_W_Bush}
differentFrom	$\{x_1\} \sqsubseteq \neg\{x_2\}$	{john} $\sqsubseteq \neg${peter}
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter $\sqsubseteq$ hasChild
equivalentProperty	$P_1 \equiv P_2$	cost $\equiv$ price
inverseOf	$P_1 \equiv P_2^-$	hasChild $\equiv$ hasParent$^-$
transitiveProperty	$P^+ \sqsubseteq P$	ancestor$^+$ $\sqsubseteq$ ancestor
functionalProperty	$\top \sqsubseteq \leqslant 1 P$	$\top \sqsubseteq \leqslant$1hasMother
inverseFunctionalProperty	$\top \sqsubseteq \leqslant 1 P^-$	$\top \sqsubseteq \leqslant$1hasSSN$^-$

General Class Inclusion ($\sqsubseteq$) genügt: C $\equiv$ D gdw (C $\sqsubseteq$ D und D $\sqsubseteq$ C)

Offensichtliche FOL-Äquivalenzen:

- $C \equiv D \Leftrightarrow \forall x\ (C(x) \leftrightarrow D(x))$
- $C \sqsubseteq D \Leftrightarrow \forall x\ (C(x) \rightarrow D(x))$

Wissensmodellierung OWA vs. CWA

- OWA: Open World Assumption
  Die Existenz von weiteren Individuen ist möglich, sofern sie nicht explizit ausgeschlossen wird.

OWL verwendet OWA!

- CWA: Closed World Assumption
  Es wird angenommen, dass die Wissensbasis alle Individuen enthält. Datenbanken verwenden typischerweise die CWA.

Beispiel:

- T1: child(Bill,Bob), Man(Bob)
- T2: child(Bill,Bob), Man(Bob), ≤ 1 child.$\top$(Bill)

Anfrage: $\models \forall$child.Man(Bill) ?

Outline

Wichtige Inferenzprobleme

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

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 für OWL DL abwandeln. Genauer: Wir werden nur ALC behandeln.
- Tableau- und Resolutionsverfahren zeigen Unerfüllbarkeit einer Theorie.
- Rückführung der Inferenzprobleme auf das Finden von Inkonsistenten in der Wissensbasis, d.h. zeigen der Unerfüllbarkeit der Wissensbasis!

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:
  $C \sqcap D \equiv \bot$ 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.
  - Schwerer, dies gut zu implementieren!

Outline

Tableau - Transformation in NNF

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.
- W und NNF(W) sind logisch äquivalent.

Tableau - Transformation in NNF

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

Tableau - NNF - Beispiel

- $P \sqsubseteq (E \sqcap U) \sqcup \neg(\neg E \sqcup D)$.
- In Negationsnormalform:
- $\neg P \sqcup (E \sqcap U) \sqcup (E \sqcap \neg D)$.

Naives Tableauverfahren

Rückführung auf Unerfüllbarkeit/Widerspruch

Idee:

- Gegeben Wissensbasis W.
- Erzeugen von Konsequenzen der Form C(a) und $\neg$C(a), bis Widerspruch gefunden.

Tableau - Einfaches Beispiel

$C(a)$
$(\neg\, C \sqcap D)(a)$

$\neg C(a)$ ist logische Konsequenz:
2. Formel in FOL: $\neg C(a) \wedge D(a)$ daraus folgt u.a. $\neg C(a)$

Widerspruch ist gefunden.

Tableau - Weiteres Beispiel

$C(a)$ $\quad$ $\neg C \sqcup D$ $\quad$ $\neg D(a)$

Ableitung von Konsequenzen:
$C(a)$
$\neg D(a)$
$(\neg C \sqcup D)(a)$

Nun Fallunterscheidung:

1. $\neg C(a)$
   Widerspruch
2. $D(a)$
   Widerspruch

Teilen des Tableaus in zwei Zweige.

Tableau - Definitionen

- Tableauzweig: Endliche Menge von Aussagen der Form C(a), ¬C(a), R(a,b).
- Tableau: Endliche Menge von Tableauzweigen.
- Tableauzweig ist abgeschlossen wenn er ein Paar widersprüchlicher Aussgen C(a) und ¬C(a) enthält.
- Tableau ist abgeschlossen, wenn jeder Zweig von ihm abgeschlossen ist.

Tableau - Erzeugung

Auswahl	Aktion
$C(a) \in W$ (ABox)	Füge $C(a)$ hinzu.
$R(a,b) \in W$ (ABox)	Füge $R(a,b)$ hinzu.
$C \in W$ (TBox)	Füge $C(a)$ fr ein bekanntes Individuum a hinzu.
$(C \sqcap D)(a) \in A$	Füge $C(a)$ und $D(a)$ hinzu.
$(C \sqcup D)(a) \in A$	Dupliziere den Zweig. Füge zum einen Zweig $C(a)$ und zum anderen Zweig $D(a)$ hinzu.
$(\exists R.C)(a) \in A$	Füge $R(a,b)$ und $C(b)$ für neues Individuum b hinzu.
$(\forall R.C)(a) \in A$	Falls $R(a,b) \in A$, so füge $C(b)$ hinzu.

- Ist das resultierende Tableau abgeschlossen, so ist die ursprüngliche Wissensbasis unerfüllbar.
- Man wählt dabei immer nur solche Elemente aus, die auch wirklich zu neuen Elementen im Tableau führen. Ist dies nicht möglich, so terminiert der Algorithmus und die Wissensbasis ist erfüllbar.

Tableau - Beispiel (1/2)

- P...Professor
  E...Person
  U...Universitätsangehöriger
  D...Doktorand

- Wissensbasis: $P \sqsubseteq (E \sqcap U) \sqcup (E \sqcap \neg D)$
  Ist $P \sqsubseteq E$ logische Konsequenz?

- Wissensbasis (mit Anfrage) in NNF:
  $\{\neg P \sqcup (E \sqcap U) \sqcup (E \sqcap \neg D), (P \sqcap \neg E)(a)\}$

Tableau - Beispiel (2/2)

TBox: $\neg P \sqcup (E \sqcap U) \sqcup (E \sqcap \neg D)$

Tableau:

$(P \sqcap \neg E)(a)$ (aus Wissensbasis)

$P(a)$

$\neg E(a)$

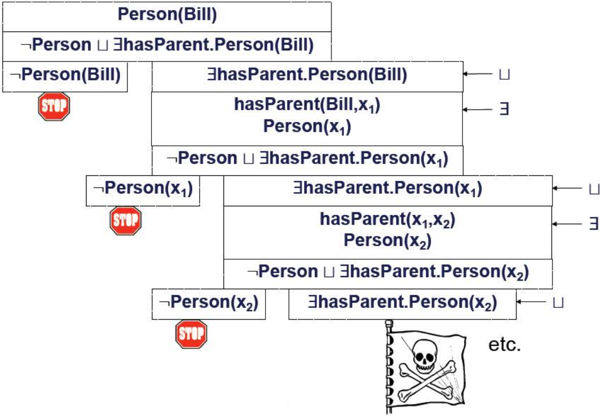
$(\neg P \sqcup (E \sqcap U) \sqcup (E \sqcap \neg D))(a)$

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

D.h. Wissensbasis ist unerfüllbar, d.h. $P \sqsubseteq E$.

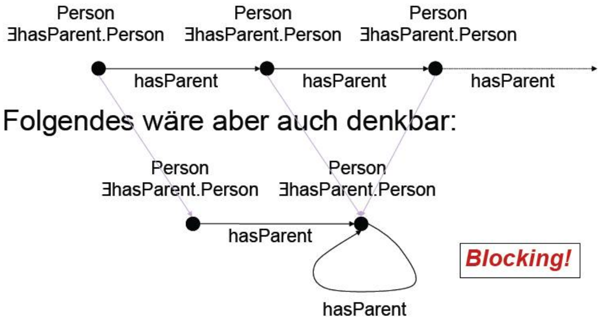
Tableau - Terminierungsproblem

Einziges Axiom: ¬Person ⊔ ∃hasParent.Person
Abzuleiten: ¬Person(Bill)



Problem tritt auf durch Existenzquantoren (und minCardinality)!

Wir haben folgendes konstruiert:



D.h. Wiederverwendung alter Knoten!
Es muss natürlich formal nachgewiesen werden, dass das ausreicht!

Tableau mit Blocking

Einziges Axiom: ¬Person ⊔ ∃hasParent.Person
Abzuleiten: ¬Person(Bill)

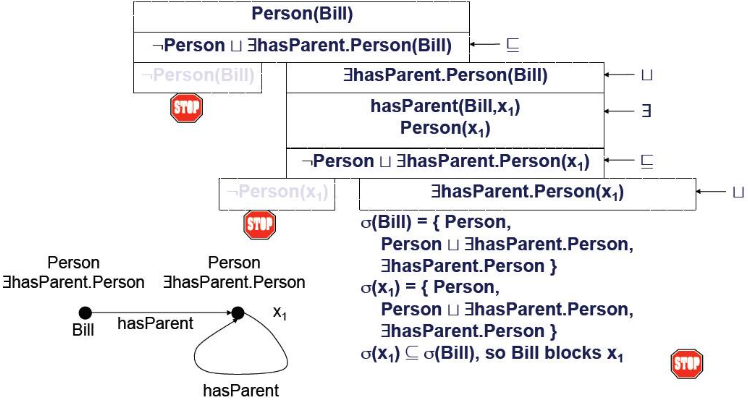


Tableau - Blocking - Definition

Die Auswahl von $(\exists R.C)(a)$ im Tableauzweig A ist blockiert, falls es ein Individuum b gibt, so dass $\{C \mid C(a) \in A\} \subseteq \{C \mid C(b) \in A\}$ gilt.

Zwei Moglichkeiten der Terminierung:

1. Abschluss des Tableaus. Dann Wissensbasis unerfüllbar.
2. Keine ungeblockte Auswahl fuhrt zu Erweiterung. Dann Wissensbasis erfullbar.

Tableau für OWL DL

- Die Grundidee ist dieselbe!
- Kompliziertere Blockingregeln müssen verwendet werden.
- Schlechte Unterstützung von Instanzgenerierung.
- Tableau mit Blocking ist 2NExptime!
  $\rightarrow$ schlechter als nötig!

Literatur

Pascal Hitzler
Markus Krötzsch
Sebastian Rudolph
York Sure

Semantic Web
Grundlagen

Springer 2008, 277 S., Softcover
ISBN: 978-3-540-33993-9
Aktuelle Literaturhinweise online