Semantic Web Technologies I

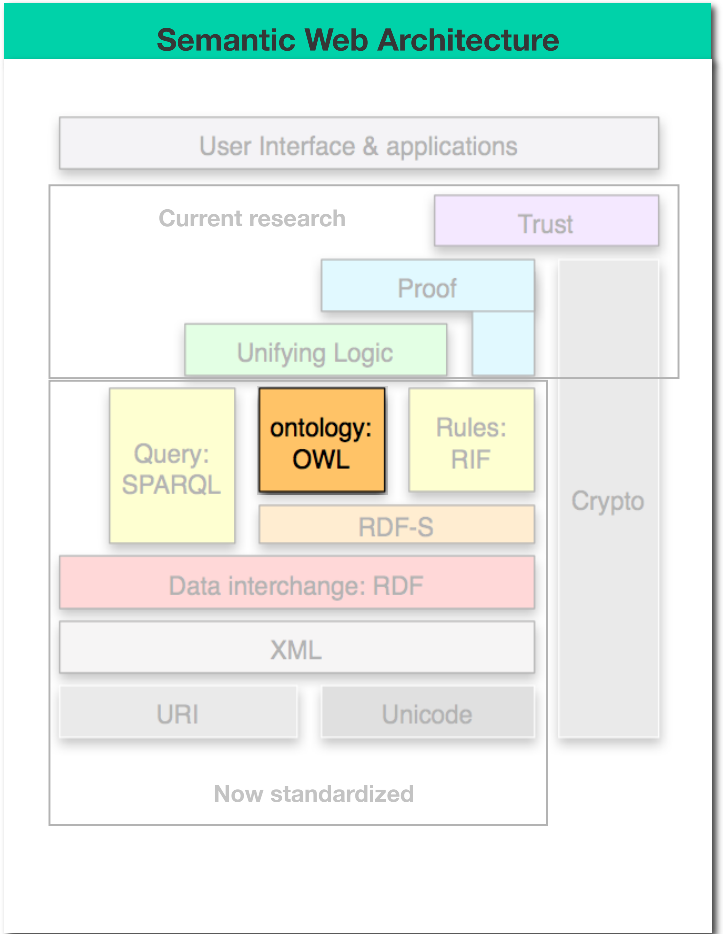
Lehrveranstaltung im WS10/11

Dr. Andreas Harth
Dr. Sebastian Rudolph

AIFB

OWL - Semantik & Reasoning

Dr. Sebastian Rudolph



AIFB

Agenda

- Beschreibungslogiken
- $\mathcal{ALC}$
- OWL als $\mathcal{SROIQ}$(D)
- Inferenzprobleme
- Tableau-Beweiser

Agenda

- Beschreibungslogiken
- $\mathcal{ALC}$
- OWL als $\mathcal{SROIQ}$(D)
- Inferenzprobleme
- Tableau-Beweiser

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 $\mathcal{SROIQ}$ (D)
- wir besprechen zunächst $\mathcal{ALC}$ (Basis für komplexere DLs)

Semantic Web TECHNOLOGIES

DLs - Aufbau

AIFB

- DLs sind eine Familie logikbasierter Formalismen zur Wissensrepräsentation
- Spezielle Sprachen v.a. charakterisiert durch:
  - Konstruktoren für komplexe Klassen und Rollen aus einfacheren.
  - Menge von Axiomen um Fakten über Klassen, Rollen und Individuen auszudrücken.

- $\mathcal{ALC}$ ist die kleinste DL, die aussagenlogisch abgeschlossen ist
  - Konjunktion, Disjunktion, Negation sind Konstruktoren, geschrieben $\sqcap$, $\sqcup$, $\neg$.
  - Quantoren schränken Rollenbereiche ein, z.B.:

- Man $\sqcap$ $\exists$hasChild.Female $\sqcap$ $\exists$hasChild.Male $\sqcap$ $\forall$hasChild.(Rich $\sqcup$ Happy)

DLs - weitere Sprachmittel

- Andere Konstruktoren sind z.B.
  - number restrictions (Kardinalitätseinschränkungen) für Rollen:
    $\geq$3 hasChild, $\leq$1hasMother
  - qualified number restrictions (klassenspezifische Kardinalitäts-einschränkungen) für Rollen:
    $\geq$2 hasChild.Female, $\leq$1 hasParent.Male
  - abgeschlossene Klassen (Definition durch Aufzählung):
    {Italy, France, Spain}
  - konkrete Bereiche (Datentypen):
    hasAge.($\geq$21)

  - inverse Rollen: hasChild^{-}
  - transitive Rollen: Trans(hasAncestor)
  - Rollenverknüpfung: Sibling $\circ$ Parent

Agenda

- Beschreibungslogiken
- $\mathcal{ALC}$
- OWL als $\mathcal{SROIQ}$(D)
- Inferenzprobleme
- Tableau-Beweiser

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 $\sqsubseteq$ Fakultaetsmitglied

- „Jeder Professor ist ein Fakultätsmitglied.“
- entspricht $(\forall x)(\text{Professor}(x) \rightarrow \text{Fakultaetsmitglied}(x))$
- entspricht owl:subClassOf

Professor $\equiv$ Fakultaetsmitglied

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

ALC - komplexe Klassen

AIFB

Konjunktion $\sqcap$ entspricht owl:intersectionOf

Disjunktion $\sqcup$ entspricht owl:unionOf

Negation $\neg$ entspricht owl:complementOf

Beispiel:

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

$$(\forall x)(\text{Professor}(x) \rightarrow ((\text{Person}(x) \land \text{Universitaetsangehoeriger}(x)) \lor (\text{Person}(x) \land \neg\text{Doktorand}(x)))$$

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: $\bot \equiv C \sqcap \neg C$
- owl:thing: $\top \equiv C \sqcup \neg C$
- owl:disjointWith: $C \sqcap D \equiv \bot$
  (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 $\mathcal{ALC}$. Dabei ist A eine atomare Klasse und R eine Rolle.

$$C,D \to A \mid \top \mid \bot \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \forall R.C \mid \exists R.C$$

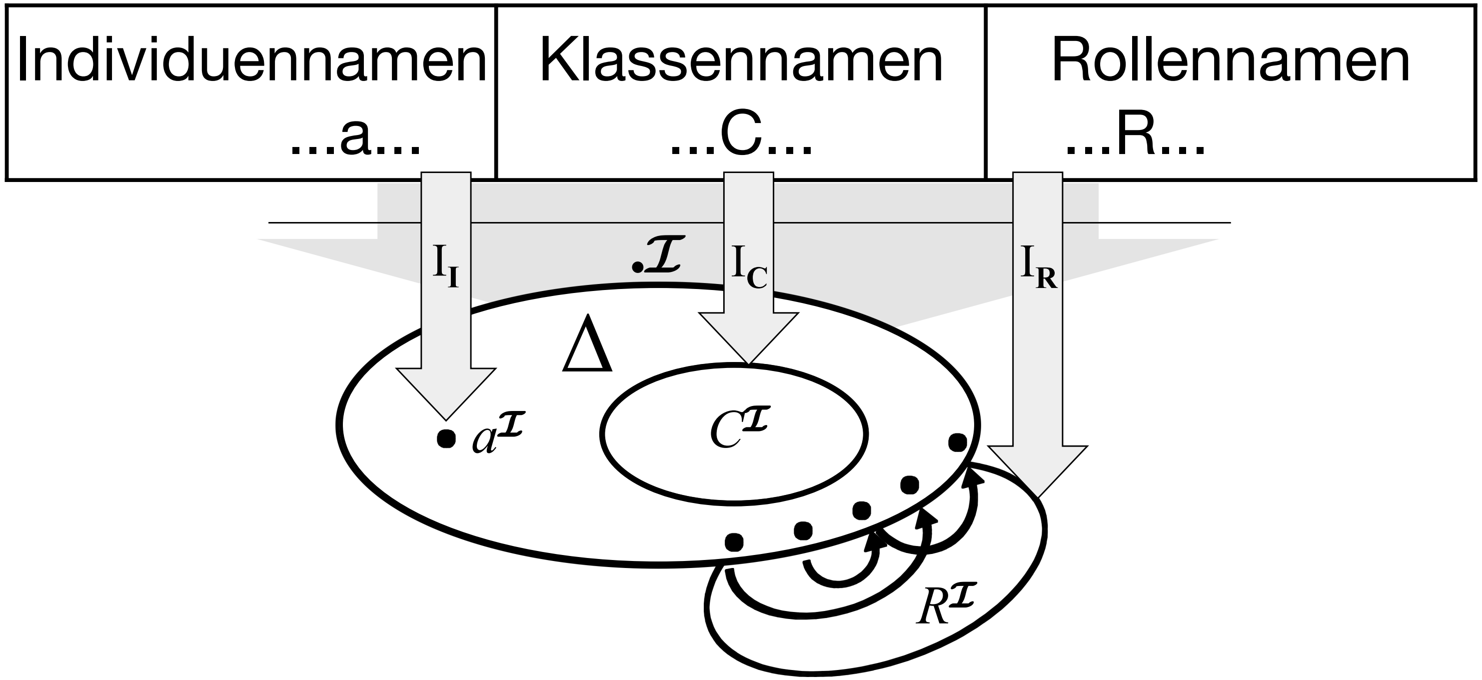
- Eine $\mathcal{ALC}$-*TBox* besteht aus Aussagen der Form $C \sqsubseteq D$ und $C \equiv D$, wobei C, D Klassen sind.
- Eine $\mathcal{ALC}$-*ABox* besteht aus Aussagen der Form C(a) und R(a,b), wobei C eine komplexe Klasse, R eine Rolle und a,b Individuen sind.
- Eine $\mathcal{ALC}$-*Wissensbasis* besteht aus einer ABox und einer TBox.

ALC - Semantik (Interpretationen)

- wir definieren modelltheoretische Semantik für $\mathcal{ALC}$ (d.h. Folgerung wird über Interpretationen definiert)
- eine Interpretation $\mathcal{I}=(\Delta^{\mathcal{I}},\cdot^{\mathcal{I}})$ besteht aus
  - einer Menge $\Delta^{\mathcal{I}}$, genannt Domäne und
  - einer Funktion $\cdot^{\mathcal{I}}$, die abbildet von
    - Individuennamen a auf Domänenelemente $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
    - Klassennamen C auf Mengen von Domänenelementen $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
    - Rollennamen R auf Mengen von Paaren von Domänenelementen $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$

ALC - Semantik (Interpretationen)

- schematisch:



ALC-Semantik (komplexe Klassen)

- wird auf komplexe Klassen erweitert:
  - $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$ $\quad\quad$ $\bot^{\mathcal{I}} = \emptyset$
  - $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$ $\quad\quad$ $(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$
  - $(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
  - $\forall R.C = \{ x \mid \forall (x,y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}} \}$
  - $\exists R.C = \{ x \mid \exists (x,y) \in R^{\mathcal{I}}$ mit $y \in C^{\mathcal{I}} \}$
- ...und schließlich auf Axiome:
  - $C(a)$ gilt, wenn $a^{\mathcal{I}} \in C^{\mathcal{I}}$
  - $R(a,b)$ gilt, wenn $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$
  - $C \sqsubseteq D$ gilt, wenn $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
  - $C \equiv D$ gilt, wenn $C^{\mathcal{I}} = D^{\mathcal{I}}$

ALC - alternative Semantik

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

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

DL-Wissensbasen

- DL Wissensbasen bestehen aus 2 Teilen:
  - TBox: Axiome, die die Struktur der zu modellierenden Domäne beschreiben (konzeptionelles Schema):
    - **HappyFather $\equiv$ Man $\sqcap$ $\exists$hasChild.Female $\sqcap$ …**
    - **Elephant $\sqsubseteq$ Animal $\sqcap$ Large $\sqcap$ 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

Semantic Web TECHNOLOGIES

AGENDA

AIFB

- Beschreibungslogiken
- $\mathcal{ALC}$
- OWL als $\mathcal{SROIQ}$(D)
- Inferenzprobleme
- Tableau-Beweiser

Semantic Web TECHNOLOGIES

OWL als SROIQ(D) - Individuen

AIFB

owl:sameAs

- gibt an dass zwei Individuennamen dasselbe Domänenelement bezeichnen
- DL: $a=b$
- FOL: Erweiterung durch Gleichheitsprädikat

owl:differentFrom

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

OWL ALS SHROIQ(D) – ABGESCHLOSSENE KLASSEN

Abgeschlossene Klassen

- owl:oneOf
  - definiert eine Klasse durch vollständige Aufzählung ihrer Instanzen
  - DL: $C \equiv \{a,b,c\}$
  - FOL: $(\forall x)\ (C(x) \leftrightarrow (x{=}a \lor x{=}b \lor x{=}c))$

- owl:hasValue
  - „erzwingt" Rolle zu einem bestimmten Individuum
  - darstellbar mittels owl:someValuesFrom und owl:oneOf

OWL ALS SROIQ(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… Prüfung, h…hatPruefer)

$$(\forall x)(P(x) \rightarrow \neg(\exists x1)(\exists x2)(\exists x3)(x1 \neq x2 \wedge x2 \neq x3 \wedge x1 \neq x3 \wedge h(x,x1) \wedge h(x,x2) \wedge h(x,x3)))$$

Entsprechend für die anderen Zahlenrestriktionen

OWL als SROIQ(D) – Self-Loops

Self-Restriktionen

- owl:hasSelf
  - definiert eine Klasse der mit sich selbst verknüpften Instanzen
  - DL: $C \equiv \exists R.\mathrm{Self}$
  - FOL: $(\forall x)\ (C(x) \leftrightarrow R(x,x))$

OWL ALS SROIQ(D) - ROLLEN

rdfs:subPropertyOf

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

Entsprechend Rollenäquivalenz

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

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

Rollenketten: $R_1 \circ \ldots \circ R_n \sqsubseteq S$

- FOL: $(\forall x_1 \ldots x_{n+1})\ (R_1(x_1,x_2) \wedge \ldots \wedge R_n(x_n,x_{n+1}) \rightarrow S(x_1,x_{n+1}))$

Rollendisjunktheit: Disj(R,S)

- FOL: $(\forall x)(\forall y)(R(x,y) \rightarrow \neg S(y,x))$

OWL als SROIQ(D) - Rollen

transitive Rollen: Trans(R)

- gibt an, dass eine Rolle transitiv ist
- Ausdrückbar durch Rollenketten: $R \circ R \sqsubseteq R$
- FOL: $(\forall x)(\forall y)(\forall z)(R(x,y) \land R(y,z) \to R(x,z))$

Symmetrie: $R \equiv R^{-}$

- ausdrückbar als Rollenäquivalenz mit der Inversen

Funktionalität: $\top \sqsubseteq {\leq}1\ R$

- ausdrückbar durch Klasseninklusion und Kardinalitätsrestriktion

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

- ausdrückbar wie Funktionalität zuzüglich inverser Rolle

OWL als SROIQ(D) - Überblick

Erlaubt sind:

- $\mathcal{ALC}$
- Gleichheit und Ungleichheit zwischen Individuen
- Abgeschlossene Klassen
- Zahlenrestriktionen
- Subrollen und Rollenäquivalenz
- Inverse und transitive Rollen
- Datentypen

Semantic Web TECHNOLOGIES

AIFB

DLs - Nomenklatur

- $\mathcal{ALC}$: Attribute Language with Complement
- $\mathcal{S}$: $\mathcal{ALC}$ + Rollentransitivität
- $\mathcal{H}$: Subrollenbeziehung
- $\mathcal{O}$: abgeschlossene Klassen
- $\mathcal{I}$: inverse Rollen
- $\mathcal{N}$: Zahlenrestriktionen $\leq$n R etc.
- $\mathcal{Q}$: Qualifizierende Zahlenrestriktionen $\leq$n R.C etc.
- (D): Datentypen
- $\mathcal{F}$: Funktionale Rollen
- $\mathcal{R}$: Rollenverkettung, Self-Loops, Rollendisjunktheit

- OWL DL ist $\mathcal{SROIQ}$ (D)

DL-Syntax - Übersicht

Concepts

$\mathcal{ALC}$	Atomic	A, B
	Not	¬C
	And	C ⊓ D
	Or	C ⊔ D
	Exists	∃R.C
	For all	∀R.C
$\mathcal{Q}$ ($\mathcal{N}$)	At least	≥n R.C (≥n R)
	At most	≤n R.C (≤n R)
$\mathcal{O}$	Closed class	$\{i_1,\ldots,i_n\}$
$\mathcal{R}$	Self	∃R.Self

Roles

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

Ontology (=Knowledge Base)

Concept Axioms (TBox)

Subclass	C ⊑ D
Equivalent	C ≡ D

Role Axioms (RBox)

$\mathcal{H}$	Subrole	R ⊑ S
$\mathcal{S}$	Transitivity	Trans(S)
$\mathcal{R}$	Role Chain	R ∘ R′ ⊑ S
	R. Disjointness	Disj(S)

Assertional Axioms (ABox)

Instance	C(a)
Role	R(a,b)
Same	a = b
Different	a ≠ b

$\mathcal{S}$ = $\mathcal{ALC}$ + Transitivity **OWL DL = $\mathcal{SROIQ}$(D)** (D: concrete domain)

DL-Syntax - Klassenkonstruktoren

Semantic Web TECHNOLOGIES

AIFB

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

Beliebig komplexes Schachteln von Konstruktoren erlaubt:

Person $\sqcap$ $\forall$hasChild.(Doctor $\sqcup$ $\exists$hasChild.Doctor)

Semantic Web TECHNOLOGIES

DL-Syntax - Axiome

AIFB

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
sameAs	$\{x_1\} \equiv \{x_2\}$	{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:

$$C \equiv D \text{ gdw } (C \sqsubseteq D \text{ und } D \sqsubseteq C)$$

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.

Are all children of Bill male?

No idea, since we do not know all children of Bill.

If we assume that we know everything about Bill, then all of his children are male.

		DL answers	Prolog
child(Bill,Bob)			
Man(Bob)	**? $\models \forall$child.Man(Bill)**	**don't know**	**yes**
$\leq$ 1 child.T(Bill)	**? $\models \forall$child.Man(Bill)**	**yes**	

Now we know everything about Bill's children.

Agenda

- Beschreibungslogiken
- $\mathcal{ALC}$
- OWL als $\mathcal{SROIQ}$(D)
- Inferenzprobleme
- Tableau-Beweiser

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.

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 $\mathcal{ALC}$ behandeln.
- Tableau- und Resolutionsverfahren zeigen Unerfüllbarkeit einer Theorie.

- $\rightarrow$ Rückführung der Inferenzprobleme auf das Finden von Inkonsistenten in der Wissensbasis,
  d.h. zeigen der Unerfüllbarkeit der Wissensbasis!

Semantic Web TECHNOLOGIES

Rückführung auf Unerfüllbarkeit

AIFB

- Klassenkonsistenz $C \equiv \bot$ gdw
  - KB $\cup$ {C(a)} unerfüllbar (a neu)
- Klasseninklusion (Subsumption) $C \sqsubseteq D$ gdw
  - KB $\cup$ {C $\sqcap$ ¬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 $\cup$ {¬C(a)} unerfüllbar (a neu)
- Instanzgenerierung (Retrieval) alle C(X) finden
  - Prüfe Klassenzugehörigkeit für alle Individuen.
  - Schwerer, dies gut zu implementieren!

Agenda

- Beschreibungslogiken
- $\mathcal{ALC}$
- OWL als $\mathcal{SROIQ}$(D)
- Inferenzprobleme
- Tableau-Beweiser

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.

Tableau - Transformation in NNF

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

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

$NNF(\neg\neg C) = NNF(C)$

$NNF(C \sqcup D) = NNF(C) \sqcup NNF(D)$

$NNF(C \sqcap D) = NNF(C) \sqcap NNF(D)$

$NNF(\neg(C \sqcup D)) = NNF(\neg C) \sqcap NNF(\neg D)$

$NNF(\neg(C \sqcap D)) = NNF(\neg C) \sqcup NNF(\neg D)$

$NNF(\forall R.C) = \forall R.NNF(C)$

$NNF(\exists R.C) = \exists R.NNF(C)$

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

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

W und NNF(W) sind logisch äquivalent.

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 ¬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: $\neg C(a) \land D(a)$

daraus folgt u.a. $\neg C(a)$

Widerspruch ist gefunden.

Tableau - weiteres Beispiel

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

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:
  $\{\neg P \sqcup (E \sqcap U) \sqcup (E \sqcap \neg D), (P \sqcap \neg E)(a)\}$

Semantic Web TECHNOLOGIES

AIFB

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: $\neg\text{Person} \sqcup \exists\text{hasParent.Person}$

Abzuleiten: $\neg\text{Person(Bill)}$

Person(Bill)

$\neg\text{Person} \sqcup \exists\text{hasParent.Person(Bill)}$

$\neg\text{Person(Bill)}$ | $\exists\text{hasParent.Person(Bill)}$ ← $\sqcup$

$\text{hasParent}(\text{Bill},x_1)$
$\text{Person}(x_1)$ ← $\exists$

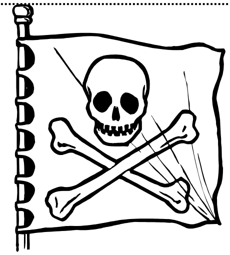
$\neg\text{Person} \sqcup \exists\text{hasParent.Person}(x_1)$

$\neg\text{Person}(x_1)$ | $\exists\text{hasParent.Person}(x_1)$ ← $\sqcup$

$\text{hasParent}(x_1,x_2)$
$\text{Person}(x_2)$ ← $\exists$

$\neg\text{Person} \sqcup \exists\text{hasParent.Person}(x_2)$

$\neg\text{Person}(x_2)$ | $\exists\text{hasParent.Person}(x_2)$ ← $\sqcup$

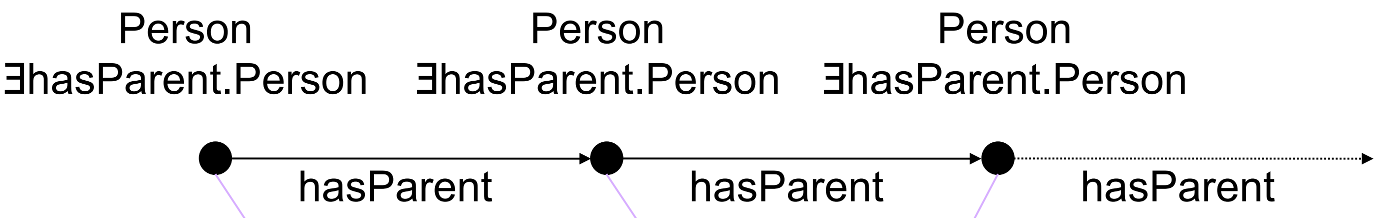


etc.

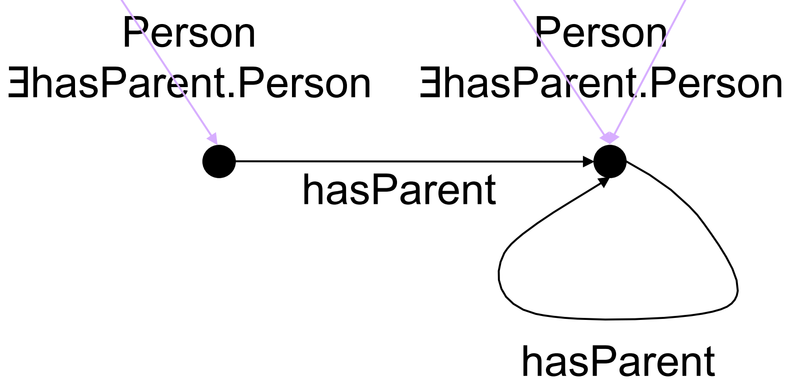
Problem tritt auf durch Existenzquantoren (und minCardinality)

Tableau - Blocking - Idee

Wir haben folgendes konstruiert:



Folgendes wäre aber auch denkbar:



Blocking!

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

Tableau mit Blocking

- Einziges Axiom: $\neg$Person $\sqcup$ $\exists$hasParent.Person

  Abzuleiten: $\neg$Person(Bill)

Person(Bill)

$\neg$Person $\sqcup$ $\exists$hasParent.Person(Bill) ← $\sqsubseteq$

$\neg$Person(Bill) | **$\exists$hasParent.Person(Bill)** ← $\sqcup$

STOP

hasParent(Bill,x_1)
Person(x_1) ← $\exists$

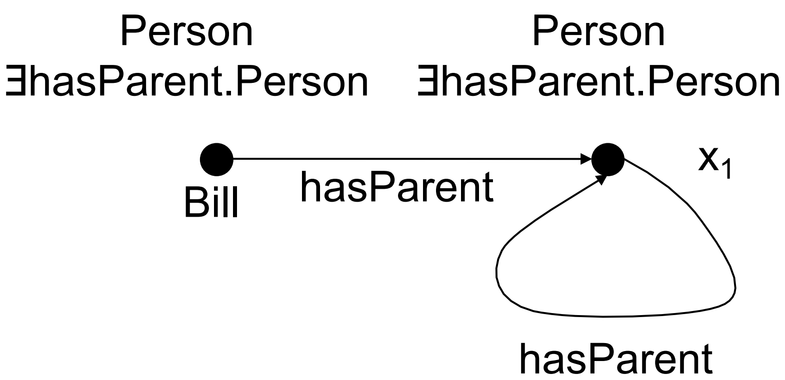
$\neg$Person $\sqcup$ $\exists$hasParent.Person(x_1) ← $\sqsubseteq$

$\neg$Person(x_1) | **$\exists$hasParent.Person(x_1)** ← $\sqcup$

STOP

σ(Bill) = { Person,
Person $\sqcup$ $\exists$hasParent.Person,
$\exists$hasParent.Person }
$\sigma(x_1)$ = { Person,
Person $\sqcup$ $\exists$hasParent.Person,
$\exists$hasParent.Person }
$\sigma(x_1) \subseteq \sigma$(Bill), so Bill blocks x_1

STOP



Semantic Web TECHNOLOGIES

Tableau - Blocking - Definition

AIFB

Die Auswahl von $(\exists R.C)(a)$ im Tableauzweig A ist *blockiert*, falls es ein Individuum b gibt, so dass

$\{C \mid C(a) \in A\} \subseteq \{C \mid C(b) \in A\}$ ist.

Zwei Möglichkeiten der Terminierung:

1. Abschluss des Tableaus.
   Dann Wissensbasis unerfüllbar.
2. Keine ungeblockte Auswahl führt zu Erweiterung.
   Dann Wissensbasis erfüllbar.

Tableau für OWL DL

- Die Grundidee ist dieselbe!
- Kompliziertere Blockingregeln müssen verwendet werden.
- Schlechte Unterstützung von Instanzgenerierung.

Tableau-Beweiser

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