ulm university universität ulm

Semantic Web Grundlagen

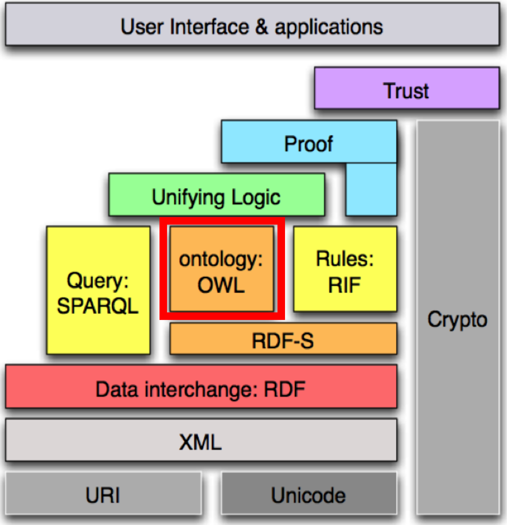
OWL & Beschreibungslogiken

Birte Glimm
Institut für Künstliche Intelligenz | 21. Nov 2011

Organisatorisches: Inhalt

Einleitung und XML	17. Okt	SPARQL Syntax	12. Dez
Einführung in RDF	20. Okt	Übung 4	15. Dez
RDF Schema	24. Okt	SPARQL Semantik	19. Dez
fällt aus	27. Okt	SPARQL 1.1	22. Dez
Logik – Grundlagen	31. Okt	Übung 5	9. Jan
Übung 1	3. Nov	SPARQL Entailment	12. Jan
Semantik von RDF(S)	7. Nov	SPARQL Implemetierung	16. Jan
RDF(S) & Datalog Regeln	10. Nov	Abfragen & RIF	19. Jan
OWL Syntax & Intuition	14. Nov	Übung 6	23. Jan
Übung 2	17. Nov	Ontology Editing	26. Jan
OWL & BLs	21. Nov	Ontology Engineering	30. Jan
OWL 2	24. Nov	Linked Data	2. Feb
Tableau	28. Nov	Übung 7	6. Feb
Übung 3	1. Dez	SemWeb Anwendungen	9. Feb
Blocking & Unravelling	5. Dez	Wiederholung	13. Feb
Hypertableau	8. Dez	Übung 8	16. Feb

OWL & Beschreibungslogiken



Agenda

- Motivation
- Einführung Beschreibungslogiken
- Die Beschreibungslogik $\mathcal{ALC}$
- Erweiterungen von $\mathcal{ALC}$
- Inferenzprobleme

Beschreibungslogiken

- Beschreibungslogiken (BLs, engl. Description Logics) sind einer der aktuellen KR Paradigmen
- Beeinflussten wesentlich die Standardisierung von Semantic Web Sprachen
  - OWL ist im Wesentlichen basierent auf BLs
- Zahlreiche Inferenzmaschinen

Quonto	JFact	FaCT++	RacerPro
Owlgres	Pellet	SHER	snorocket
OWLIM	Jena	Oracle Prime	QuOnto
Trowl	HermiT	condor	CB
	ELK	konclude	RScale

OWL Werkzeuge

Guter Support durch Editoren

- Protégé, `http://protege.stanford.edu`
- SWOOP, `http://code.google.com/p/swoop/`
- OWL Tools, `http://owltools.ontoware.org/`
- Neon Toolkit, `http://neon-toolkit.org/`

Beschreibungslogiken

- Ursprung: Semantic Networks und Frame-basierte Systeme
- Nachteile der Ursprungssysteme: nur intuitive Semantik – abweichende Interpretationsmöglichkeiten
- Beschreibungslogiken geben eine formale und logik-basierte Semantik
- Können als entscheidbare Fragmente der Prädikatenlogik erster Stufe verstanden werden, verwandt zu Modallogiken
- Viel Forschung zur Bestimmung der worst-case Komplexität von typischen Problemen des Automatischen Schlussfolgerns (z.B. Konsistenz von Ontologien)
- Trotz hoher Komplexität gibt es optimierte Algorithmen mit gutem average-case Verhalten

Agenda

- Motivation
- Einführung Beschreibungslogiken
- Die Beschreibungslogik $\mathcal{ALC}$
- Erweiterungen von $\mathcal{ALC}$
- Inferenzprobleme

BL Konstruktionsblöcke

- Individuen: `birte`, `cs63.800`, `sebastian`, etc.
  - $\leadsto$ Konstanten in der Prädikatenlogik, Ressourcen in RDF
- Konzeptnamen: `Person`, `Kurs`, `Student`, etc.
  - $\leadsto$ Unäre Prädikate in der Prädikatenlogik, Klassen in RDF
- Rollennamen: `hatVater`, `besucht`, `arbeitetMit`, etc.
  - $\leadsto$ Binäre Prädikate in der Prädikatenlogik, Propertys in RDF
  - Können in abstrakte und konkrete Rollen unterteilt werden (object und data properties)

Die Menge aller Individuen-, Konzept- und Rollennamen wird als Signatur oder Vokabular bezeichnet

Teile einer BL Wissensbasis

TBox $\mathcal{T}$	Informationen über Konzepte und ihre taxonomischen Abhängigkeiten
ABox $\mathcal{A}$	Informationen über Individuen, ihre Konzepte und Rollenverbindungen

Bei ausdrucksstärkeren BLs auch:

RBox $\mathcal{R}$	Informationen über Rollen und ihre Abhängigkeiten

Agenda

- Motivation
- Einführung Beschreibungslogiken
- Die Beschreibungslogik $\mathcal{ALC}$
- Erweiterungen von $\mathcal{ALC}$
- Inferenzprobleme

Komplexe Konzepte

$\mathcal{ALC}$, Attribute Language with Complement, ist die einfachste BL, die aussagenlogisch abgeschlossen ist

Wir definieren (komplexe) $\mathcal{ALC}$-Konzepte induktiv wie folgt:

- jeder Konzeptname ist eine Konzept,
- $\top$ und $\bot$ sind Konzepte,
- für C und D Konzepte, $\neg C$, $C \sqcap D$, und $C \sqcup D$ sind Konzepte,
- für r eine Rolle und C ein Konzept, $\exists r.C$ und $\forall r.C$ sind Konzepte

Beispiel: $\texttt{Student} \sqcap \forall \texttt{besuchtKurs}.\texttt{Masterkurs}$
Intuitiv: Beschreibt die Klasse der Studenten, due nur Masterkurse besuchen

Konzeptkonstruktoren und OWL

- $\top$ entspricht `owl:Thing`
- $\bot$ entspricht `owl:Nothing`
- $\sqcap$ entspricht `owl:intersectionOf`
- $\sqcup$ entspricht `owl:unionOf`
- $\neg$ entspricht `owl:complementOf`
- $\forall$ entspricht `owl:allValuesFrom`
- $\exists$ entspricht `owl:someValuesFrom`

Konzept Axiome

Für C, D Konzepte, eine generelles Konzeptinklusions-Axiom (engl. general concept inclusion – GCI) hat die Form

$$C \sqsubseteq D$$

- $C \equiv D$ ist eine Abkürzung für $C \sqsubseteq D$ und $D \sqsubseteq C$
- Eine TBox (terminologische Box) besteht aus einer Menge von Konzept Axiomen

TBox $\mathcal{T}$

ABox

Ein ABox Fakt kann eine der folgenden Formen haben

- $C(a)$, genannt Konzept Fakt (concept assertion)
- $r(a, b)$, genannt Rollen Fakt (role assertion)

Eine ABox besteht aus einer Menge von ABox Fakten.

ABox $\mathcal{A}$

Eine Beispiel Wissensbasis

TBox $\mathcal{T}$

$$\texttt{Healthy} \sqsubseteq \neg\,\texttt{Dead}$$

“Gesunde sind nicht tot. ”

$$\texttt{Cat} \sqsubseteq \texttt{Dead} \sqcup \texttt{Alive}$$

“Jede Katze ist entweder tot oder lebendig. ”

$$\texttt{HappyCatOwner} \sqsubseteq \exists\,\texttt{owns.Cat} \sqcap \forall\,\texttt{owns.Alive}$$

“Ein glücklicher Katzenbesitzer besitzt eine Katze und alle Dinge, die er/sie besitzt sind gesund. ”

ABox $\mathcal{A}$

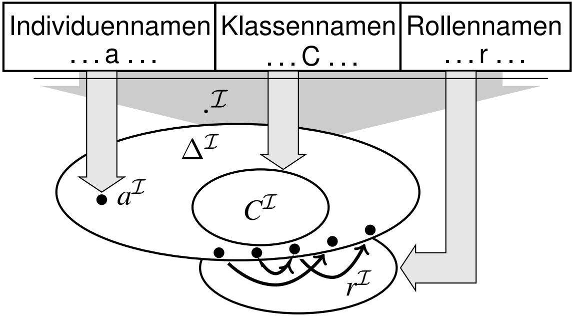
$$\texttt{HappyCatOwner}\ (\texttt{schrödinger})$$

“Schrödinger ist ein glücklicher Katzenbesitzer. ”

Die Beschreibungslogik $\mathcal{ALC}$

- $\mathcal{ALC}$ ist eine syntaktische Variante der Modallogik **K**
- Die Semantik ist modelltheoretisch definiert, d.h., wieder über Interpretationen für die logischen Formeln
- Entspricht einer Übersetzung in die Prädikatenlogik erster Stufe
- Eine BL Interpretation $\mathcal{I}$ besteht aus einer Domäne $\Delta^{\mathcal{I}}$ und einer Funktion (mapping) $\cdot^{\mathcal{I}}$, 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 $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$

Schematische Darstellung einer Interpretation



Interpretation von Komplexen Konzepten

Die Interpretation komplexer Konzepte ist induktiv definiert:

Name	Syntax	Semantik
top	$\top$	$\Delta^\mathcal{I}$
bottom	$\bot$	$\emptyset$
Negation	$\neg C$	$\Delta^\mathcal{I} \setminus C^\mathcal{I}$
Konjunktion	$C \sqcap D$	$C^\mathcal{I} \cap D^\mathcal{I}$
Disjunktion	$C \sqcup D$	$C^\mathcal{I} \cup D^\mathcal{I}$
Allquantor	$\forall r.C$	$\{x \in \Delta^\mathcal{I} \mid (x,y) \in r^\mathcal{I}$ impliziert $y \in C^\mathcal{I}\}$
Existenzquantor	$\exists r.C$	$\{x \in \Delta^\mathcal{I} \mid$ es existiert $y \in \Delta^\mathcal{I}$, so dass $(x,y) \in r^\mathcal{I}$ und $y \in C^\mathcal{I}\}$

Interpretation von Axiomen

Die Interpretation kann nun auf Axiome erweitert werden:

Name	Syntax	Semantik	geschrieben
Inklusion	$C \sqsubseteq D$	gilt wenn $C^\mathcal{I} \subseteq D^\mathcal{I}$	$\mathcal{I} \models C \sqsubseteq D$
Äquivalenz	$C \equiv D$	gilt wenn $C^\mathcal{I} = D^\mathcal{I}$	$\mathcal{I} \models C \equiv D$
Konzept Fakt	$C(a)$	gilt wenn $a^\mathcal{I} \in C^\mathcal{I}$	$\mathcal{I} \models C(a)$
Rollen Fakt	$r(a, b)$	gilt wenn $(a^\mathcal{I}, b^\mathcal{I}\mathcal{I}) \in r^\mathcal{I}$	$\mathcal{I} \models r(a, b)$

Logisches Folgern in Wissensbasen

- Sei $\mathcal{I}$ eine Interpretation, $\mathcal{T}$ eine TBox, $\mathcal{A}$ eine Abox und $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ eine Wissensbasis
- $\mathcal{I}$ ist ein Modell für eine $\mathcal{T}$, wenn $\mathcal{I} \models$ ax für jedes Axiom ax in $\mathcal{T}$, geschrieben $\mathcal{I} \models \mathcal{T}$
- $\mathcal{I}$ ist ein Modell für $\mathcal{A}$, wenn $\mathcal{I} \models$ ax für jedes Fakt ax in $\mathcal{A}$, geschrieben $\mathcal{I} \models \mathcal{A}$
- $\mathcal{I}$ ist ein Modell für $\mathcal{K}$, wenn $\mathcal{I} \models \mathcal{T}$ und $\mathcal{I} \models \mathcal{A}$
- Aus $\mathcal{K}$ folgt ein Axiom ax, geschrieben $\mathcal{K} \models$ ax, wenn jedes Modell $\mathcal{I}$ von $\mathcal{K}$ auch ein Modell für ax ist

Semantik durch Übersetzung in FOL

Übersetzung von TBox-Aussagen in die Prädikatenlogik mittels der Abbildung π mit C, D komplexe Klassen, r eine Rolle und A eine atomare Klasse:

$$\pi(C \sqsubseteq D) = \forall x.(\pi_x(C) \rightarrow \pi_x(D)) \qquad \pi(C \equiv D) = \forall x.(\pi_x(C) \leftrightarrow \pi_x(D))$$

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

Semantik durch Übersetzung in FOL

- Übersetzung braucht nur zwei Variables

$\rightsquigarrow$ $\mathcal{ALC}$ ist ein Fragment der Prädikatenlogik erster Stufe mit zwei Variablen $\mathcal{L}_2$

$\rightsquigarrow$ Prüfen ob eine Menge von $\mathcal{ALC}$-Axiomen erfüllbar ist, ist entscheidbar

Agenda

- Motivation
- Einführung Beschreibungslogiken
- Die Beschreibungslogik $\mathcal{ALC}$
- Erweiterungen von $\mathcal{ALC}$
- Inferenzprobleme

Inverse Rollen

- Eine Rolle kann
  - ein Rollenname r, oder
  - eine inverse Rolle r^- sein.
- Die Semantik von inversen Rollen ist definiert als:

$$(r^-)^{\mathcal{I}} = \{(y, x) \mid (x, y) \in r^{\mathcal{I}}\}$$

- Die Erweiterung von $\mathcal{ALC}$ mit inversen Rollen wird als $\mathcal{ALCI}$ bezeichnet
- Entspricht `owl:inverseOf`

Teile einer Wissensbasis

TBox $\mathcal{T}$	Informationen über Konzepte und ihre taxonomischen Abhängigkeiten
ABox $\mathcal{A}$	Informationen über Individuen, ihre Konzepte und Rollenverbindungen

Bei ausdrucksstärkeren BLs auch:

RBox $\mathcal{R}$	Informationen über Rollen und ihre Abhängigkeiten

Rollen Axiome

- Für r, s Rollen, ein Rollen Axiom (engl. role inclusion axiom – RIA) hat die Form $r \sqsubseteq s$
- $r \equiv s$ ist eine Abkürzung für $r \sqsubseteq s$ und $s \sqsubseteq r$
- Eine RBox (Rollen Box) oder Rollen Hierarchie besteht aus einer Menge von Rollen Axiomen
- $r \sqsubseteq s$ gilt in einer Interpretation $\mathcal{I}$ wenn $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$, geschrieben $\mathcal{I} \models r \sqsubseteq s$
- Die Erweiterung von $\mathcal{ALC}$ mit Rollen Hierarchien wird als $\mathcal{ALCH}$ bezeichnet plus inverse Rollen: $\mathcal{ALCHI}$
- Entspricht `owl:subPropertyOf`

RBox $\mathcal{R}$

Eine Beispiel Wissensbasis

RBox $\mathcal{R}$

$$\mathtt{own} \sqsubseteq \mathtt{careFor}$$

“Wenn man etwas besitzt, dann kümmert man sich darum. ”

TBox $\mathcal{T}$

$$\mathtt{Healthy} \sqsubseteq \neg\,\mathtt{Dead}$$

“Gesunde sind nicht tot. ”

$$\mathtt{Cat} \sqsubseteq \mathtt{Dead} \sqcup \mathtt{Alive}$$

“Jede Katze ist entweder tot oder lebendig. ”

$$\mathtt{HappyCatOwner} \sqsubseteq \exists\,\mathtt{owns.Cat} \sqcap \forall\,\mathtt{caresFor.Alive}$$

“Ein glücklicher Katzenbesitzer besitzt eine Katze und alle Dinge, um die er/sie sich kümmert sind lebendig. ”

ABox $\mathcal{A}$

$$\mathtt{HappyCatOwner}\,(\mathtt{schrödinger})$$

“Schrödinger ist ein glücklicher Katzenbesitzer. ”

Transitivität von Rollen

- Für r eine Rolle, ein Transitivitäts-Axiom hat die Form Trans(r)
- Trans(r) gilt in einer Interpretation $\mathcal{I}$ wenn $r^{\mathcal{I}}$ eine transitive Relation ist, d.h., $(x, y) \in r^{\mathcal{I}}$ und $(y, z) \in r^{\mathcal{I}}$ impliziert $(x, z) \in r^{\mathcal{I}}$, geschrieben $\mathcal{I} \models$ Trans(r)
- Die Erweiterung von $\mathcal{ALC}$ mit Transitivitäts-Axiomen wird als $\mathcal{S}$ bezeichnet (von der Modallogik S_5)
- Entspricht `owl:TransitiveProperty`

Funktionalität von Rollen

- Für r eine Rolle, ein Funktionalitäts-Axiom hat die Form Func(r)
- Func(r) gilt in einer Interpretation $\mathcal{I}$ wenn $(x, y_1) \in r^{\mathcal{I}}$ und $(x, y_2) \in r^{\mathcal{I}}$ impliziert $y_1 = y_2$, geschrieben $\mathcal{I} \models$ Func(r)
- Übersetzung in FOL braucht nun Gleichheit (=)
- Die Erweiterung von $\mathcal{ALC}$ mit Funktionalitäts-Axiomen wird als $\mathcal{ALCF}$ bezeichnet
- Entspricht `owl:FunctionalProperty`

Einfache und Komplexe Rollen

- Gegeben eine Rollenhierarchie $\mathcal{R}$, $\sqsubseteq^*_{\mathcal{R}}$ ist der reflexive und transitive Abschluss bzgl. $\sqsubseteq$
- Gegeben eine Rollenhierarchie $\mathcal{R}$, können die Rollen in $\mathcal{R}$ in einfache und komplexe unterteilt werden
- Eine Rolle r ist *komplex* bzgl. $\mathcal{R}$, wenn es eine Rolle t gibt, so dass $\mathsf{Trans}(t) \in \mathcal{R}$ und $t \sqsubseteq^*_{\mathcal{R}} r$ gilt
- Alle anderen Rollen sind *einfach*
- Beispiel: $\mathcal{R} = \{u \sqsubseteq t, \quad t \sqsubseteq s, \quad s \sqsubseteq r, \quad q \sqsubseteq r, \quad \mathsf{Trans}(t)\}$

$$\begin{tikzcd} & & q \arrow[dr] & \\ u \arrow[r] & t \arrow[r] & s \arrow[r] & r \end{tikzcd}$$

Komplex: t, s, r Einfach: q, u

(Unqualifizierten) Zahlenrestriktionen

- Für eine einfache Rolle s, und eine natürliche Zahl n, $\leqslant n\, s$, $\geqslant n\, s$ und $= n\, s$ sind Konzepte
- Die Semantik ist definiert als:

$$(\leqslant n\, s)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \#\{y \in \Delta^{\mathcal{I}} \mid (x,y) \in s^{\mathcal{I}}\} \leq n\}$$
$$(\geqslant n\, s)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \#\{y \in \Delta^{\mathcal{I}} \mid (x,y) \in s^{\mathcal{I}}\} \geq n\}$$
$$(= n\, s)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \#\{y \in \Delta^{\mathcal{I}} \mid (x,y) \in s^{\mathcal{I}}\} = n\}$$

- Die Erweiterung von $\mathcal{ALC}$ mit (unqualifizierten) Zahlenrestriktionen wird als $\mathcal{ALCN}$ bezeichnet
- Entspricht `owl:maxCardinality`, `owl:minCardinality` und `owl:cardinality`
- Beschränkung auf einfache Rollen sichert Entscheidbarkeit z.B. für das Testen der Erfüllbarkeit einer Wissensbasis
- Definition einer TBox baut nun bereits auf einer RBox auf

(Unqualifizierten) Zahlenrestriktionen in FOL

- Übersetzung in die Prädikatenlogik erster Stufe erfordert Gleichheit bzw. Zahlquantoren (counting quantifiers)
- Die übersetzung ist definiert als (analog für π_y):

$$\pi_x(\leqslant n\, s) = \exists^{\leqslant n} y.(s(x,y))$$
$$\pi_x(\geqslant n\, s) = \exists^{\geqslant n} y.(s(x,y))$$
$$\pi_x(= n\, s) = \exists^{\leqslant n} y.(s(x,y)) \land \exists^{\geqslant n} y.(s(x,y))$$

- Die folgenden Äquivalenzen gelten:

$$\neg(\leqslant n\, s) = \geqslant n+1\, s \qquad \neg(\geqslant n\, s) = \leqslant n-1\, s, \quad n \geq 1$$
$$\neg(\geqslant 0\, s) = \bot \qquad \geqslant 1\, s = \exists s.\top$$
$$\leqslant 0\, s = \forall s.\bot \qquad \top \sqsubseteq \leqslant 1 s = \mathsf{Func}(s)$$

Nominale oder Abgeschlossene Klassen

- Definiert eine Klasse durch vollständige Aufzählung ihrer Instanzen
- Für $a_1, \ldots, a_n$ Individuen, $\{a_1, \ldots, a_n\}$ ist ein Konzept
- Die Semantik ist definiert als:

$$\text{DL: } (\{a_1, \ldots, a_n\})^{\mathcal{I}} = \{a_1^{\mathcal{I}}, \ldots, a_n^{\mathcal{I}}\}$$

$$\text{FOL: } \pi_x(\{a_1, \ldots, a_n\}) = \forall x.(x = a_1 \vee \ldots \vee x = a_n)$$

- Die Erweiterung von $\mathcal{ALC}$ mit Nominalen wird als $\mathcal{ALCO}$ bezeichnet
- Entspricht `owl:oneOf`

Nominale zum Kodieren weiterer OWL Konstruktoren

- `owl:hasValue` "erzwingt" Rolle zu einem bestimmten Individuum

```
<owl:Class rdf:ID="Woman">
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasGender"/>
      <owl:hasValue rdf:resource="#female"
                    rdf:type="#Gender"/>
    </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>
```

- In Beschreibungslogik:

$$\text{Woman} \equiv \exists \text{hasGender}.\{\text{female}\}$$

$$\text{Gender}(\text{female})$$

Weitere Formen für ABox Fakten

Ein ABox Fakt kann eine der folgenden Formen haben

- $C(a)$, genannt Konzept Fakt (concept assertion)
- $r(a, b)$, genannt Rollen Fakt (role assertion)
- $\neg r(a, b)$, genannt negativer Rollen Fakt (negative role assertion)
- $a \approx b$, genannt Gleichheits Fakt (equality assertion)
- $a \not\approx b$, genannt Ungleichheits Fakt (inequality assertion)

Internalisierung von ABox Fakten

Wenn Nominale unterstützt werden, kann jede Wissensbasis mit einer ABox in eine äquivalente Wissensbasis ohne ABox transformiert werden:

$$\begin{aligned} C(a) &= \{a\} \sqsubseteq C \\ r(a,b) &= \{a\} \sqsubseteq \exists r.\{b\} \\ \neg r(a,b) &= \{a\} \sqsubseteq \forall r.(\neg\{b\}) \\ a \approx b &= \{a\} \equiv \{b\} \\ a \not\approx b &= \{a\} \equiv \neg\{b\} \end{aligned}$$

Übersicht 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

(D) Datentypen

$\mathcal{F}$ Funktionale Rollen

OWL DL ist $\mathcal{SHOIN}(\mathcal{D})$ und OWL Lite ist $\mathcal{SHIF}(\mathcal{D})$

Begriffe in BLs und in OWL

OWL	**BL**
Klasse	Konzept
Property	Rolle
object property	abstrakte Rolle
data property	konkrete Rolle
oneOf	Nominal
Ontologie	Wissensbasis (knowledge base)
–	TBox, RBox, ABox

Eine komplexere Wissensbasis

$$\begin{aligned}
\text{Human} &\sqsubseteq \text{Animal} \sqcap \text{Biped} \\
\text{Man} &\equiv \text{Human} \sqcap \text{Male} \\
\text{Male} &\sqsubseteq \neg\text{Female} \\
\{\text{President_Obama}\} &\equiv \{\text{Barack_Obama}\} \\
\{\text{john}\} &\sqsubseteq \neg\{\text{peter}\} \\
\text{hasDaughter} &\sqsubseteq \text{hasChild} \\
\text{hasChild} &\equiv \text{hasParent}^{-} \\
\text{cost} &\equiv \text{price}
\end{aligned}$$

$$\text{Trans}(\text{ancestor})$$

$$\text{Func}(\text{hasMother})$$

$$\text{Func}(\text{hasSSN}^{-})$$

Open versus Closed World Assumption

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 und Fakten enthält

	Sind alle Kinder von Bill männlich?	Keine Ahnung, wenn wir annehmen nicht alles über Bill zu wissen	Wenn wir alles wissen, dann sind alle Kinder von Bill männlich
child(bill, bob) Man(bob)	$\models^?$ $(\forall$ child.Man)(bill)	DL answers don't know	Prolog yes
$(\leqslant 1$ child)(bill)	$\models^?$ $(\forall$ child.Man)(bill)	yes	yes

Agenda

- Motivation
- Einführung Beschreibungslogiken
- Die Beschreibungslogik $\mathcal{ALC}$
- Erweiterungen von $\mathcal{ALC}$
- Inferenzprobleme

Wichtige Inferenzprobleme für eine Wissensbasis $\mathcal{K}$

- Globale Konsistenz der Wissensbasis: $\mathcal{K} \models^? \text{false}$?
  $\mathcal{K} \models^? \top \sqsubseteq \bot$?
  - Ist die Wissensbasis sinnvoll?
- Klassenkonsistenz: $\mathcal{K} \models^? C \sqsubseteq \bot$?
  - Muss die Klasse C leer sein?
- Klasseninklusion (Subsumption): $\mathcal{K} \models^? C \sqsubseteq D$?
  - Strukturierung der Wissensbasis
- Klassenäquivalenz: $\mathcal{K} \models^? C \equiv D$?
  - Sind zwei Klassen eigentlich dieselbe?
- Klassendisjunktheit: $\mathcal{K} \models^? C \sqcap D \sqsubseteq \bot$?
  - Sind zwei Klassen disjunkt?
- Klassenzugehörigkeit: $\mathcal{K} \models^? C(a)$?
  - Ist Individuum a in der Klasse C?
- Instanzgenerierung (Retrieval): “Finde alle x, so dass $\mathcal{K} \models C(x)$ gilt”
  - 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!

Zusammenfassung

- Wir haben den Zusammenhang zwischen OWL und Beschreibungslogik kennengelernt
- Nicht jede OWL Ontologie ist eine BL Wissensbasis
  - Mapping existiert nur für wohlgeformte Ontologien
- Enger Zusammenhand zwischen BLs und (Fragmenten) der Prädikatenlogik erster Stufe
- Semantik von OWL intuitiv für Logiker
- Heisst Direct Semantics in OWL
- Unterschiede zur RDF basierten Semantik

Ausblick OWL 2

- OWL 2 erweitert die hier vorgestellten Fragmente um weitere Konstruktoren
- OWL 2 definiert aber auch einfachere Fragmente (PTime für Standard-Inferenzprobleme)
- Diverse Tools für das automatische Schlussfolgern vorhanden
- Editoren unterstützem beim Erstellen von Ontologien/Wissensbasen