

SEMANTIC WEB TECHNOLOGIES I

Lehrveranstaltung im WS09/10

M.Sc. Markus Krötzsch
Dr. Sebastian Rudolph

OWL - SEMANTIK & REASONING

M.Sc. Markus Krötzsch

Einleitung und XML

Einführung in RDF

RDF Schema

Logik - Grundlagen

Semantik von RDF(S)

OWL - Syntax und Intuition

OWL - Semantik und Reasoning

OWL 2

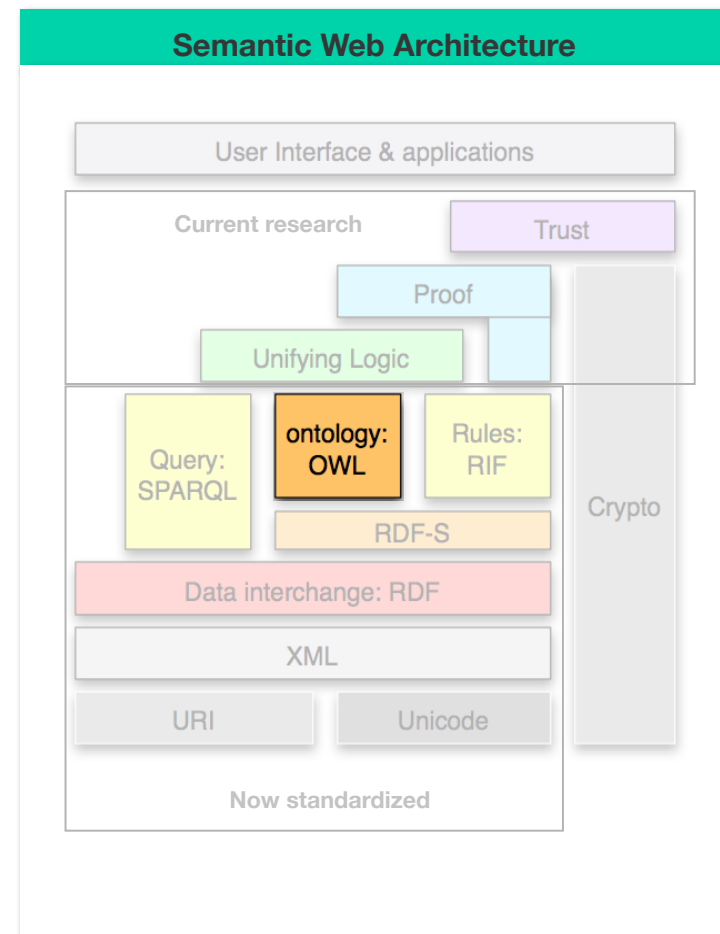
SPARQL - Syntax und Intuition

Konjunktive Anfragen / Einführung Regelsprachen

Regeln für OWL

Ontology Engineering

Semantic Web - Anwendungen



AGENDA

AIFB 

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

AGENDA

AIFB 

- **Beschreibungslogiken**
- \mathcal{ALC}
- OWL als $\mathcal{SHOIN}(\mathcal{D})$
- 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 *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:
 - 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.:
- $\text{Man} \sqcap \exists \text{hasChild.Female} \sqcap \exists \text{hasChild.Male} \sqcap \forall \text{hasChild.}(\text{Rich} \sqcup \text{Happy})$

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

AGENDA

AIFB 

- Beschreibungslogiken
- *ALC*
- OWL als *SHOIN(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



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)

$(\forall x)(\text{Professor}(x) \rightarrow$
 $((\text{Person}(x) \wedge \text{Unversitaetsangehoeriger}(x))$
 $\vee (\text{Person}(x) \wedge \neg \text{Doktorand}(x)))$

ALC - QUANTOREN AUF ROLLEN

Pruefung $\sqsubseteq \forall \text{hatPruefer. Professor}$

- „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 $\sqsubseteq \exists \text{hatPruefer. Person}$

- „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

AIFB 

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

ALC - FORMALE SYNTAX



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

$$C, D \rightarrow A \mid \top \mid \perp \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \forall R.C \mid \exists R.C$$

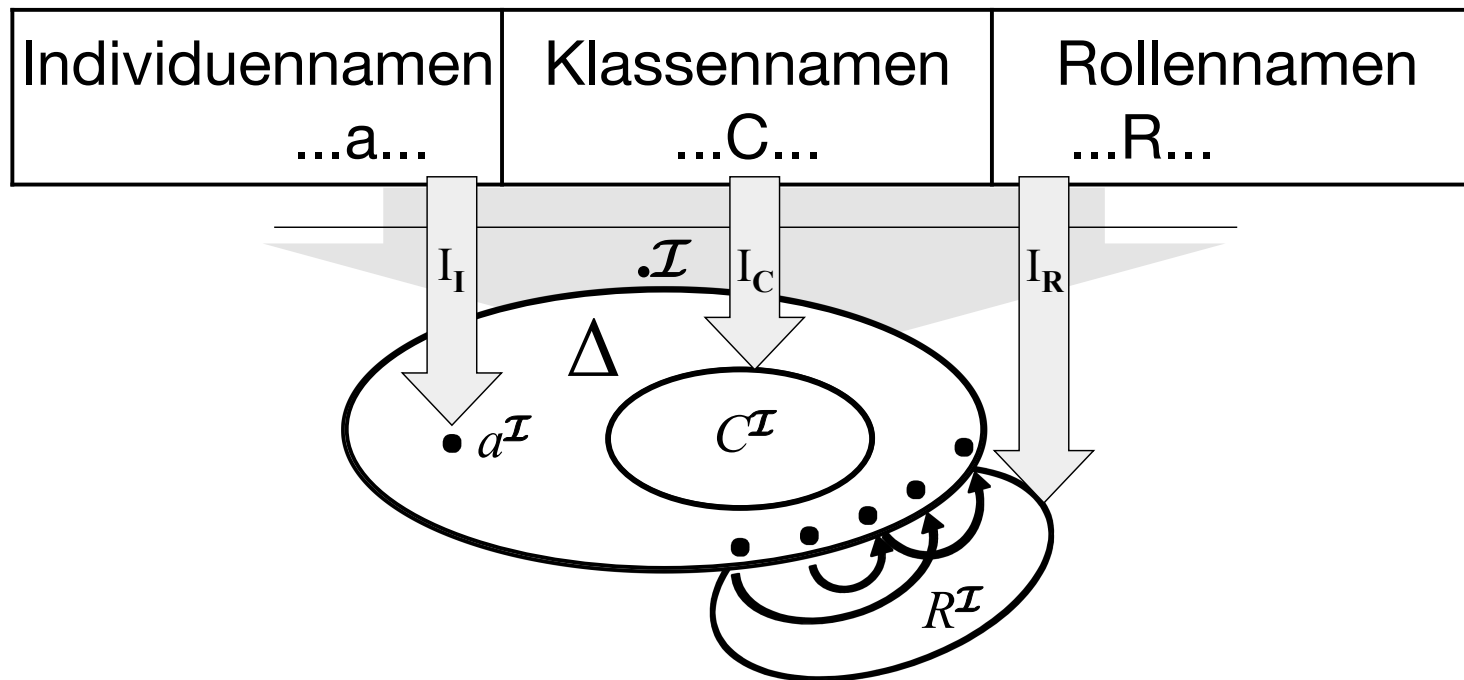
- Eine \mathcal{ALC} -TBox besteht aus Aussagen der Form $C \sqsubseteq D$ und $C \equiv D$, wobei C, D Klassen sind.
- Eine \mathcal{ALC} -ABox besteht aus Aussagen der Form $C(a)$ und $R(a,b)$, wobei C eine komplexe Klasse, R eine Rolle und a, b Individuen sind.
- Eine \mathcal{ALC} -Wissensbasis besteht 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:
 - $\top^I = \Delta^I$ $\perp^I = \emptyset$
 - $(C \sqcap D)^I = C^I \cap D^I$ $(C \sqcup D)^I = C^I \cup D^I$
 - $(\neg C)^I = \Delta^I \setminus C^I$
 - $\forall R.C = \{x \mid \forall (x,y) \in R^I \rightarrow y \in C^I\}$
 - $\exists R.C = \{x \mid \exists (x,y) \in R^I \text{ mit } y \in C^I\}$
- ...und schließlich auf Axiome:
 - $C(a)$ gilt, wenn $a^I \in C^I$
 - $R(a,b)$ gilt, wenn $(a^I, b^I) \in R^I$
 - $C \sqsubseteq D$ gilt, wenn $C^I \subseteq D^I$
 - $C \equiv D$ gilt, wenn $C^I = D^I$

ALC - ALTERNATIVE SEMANTIK

AIFB 

- 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.

$$\pi(C \sqsubseteq D) = (\forall x)(\pi_x(C) \rightarrow \pi_x(D))$$

$$\pi(C \equiv D) = (\forall x)(\pi_x(C) \leftrightarrow \pi_x(D))$$

$$\pi_x(A) = A(x)$$

$$\pi_x(\neg C) = \neg \pi_x(C)$$

$$\pi_x(C \sqcap D) = \pi_x(C) \wedge \pi_x(D)$$

$$\pi_x(C \sqcup D) = \pi_x(C) \vee \pi_x(D)$$

$$\pi_x(\forall R.C) = (\forall y)(R(x, y) \rightarrow \pi_y(C))$$

$$\pi_x(\exists R.C) = (\exists y)(R(x, y) \wedge \pi_y(C))$$

$$\pi_y(A) = A(y)$$

$$\pi_y(\neg C) = \neg \pi_y(C)$$

$$\pi_y(C \sqcap D) = \pi_y(C) \wedge \pi_y(D)$$

$$\pi_y(C \sqcup D) = \pi_y(C) \vee \pi_y(D)$$

$$\pi_y(\forall R.C) = (\forall x)(R(y, x) \rightarrow \pi_x(C))$$

$$\pi_y(\exists R.C) = (\exists x)(R(y, x) \wedge \pi_x(C))$$

DL-WISSENSBASSEN

AIFB 

- 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*):

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

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

AGENDA

AIFB 

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

OWL ALS SHOIN(D) - INDIVIDUEN



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 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: $\text{Pruefung} \sqsubseteq \leq 2 \text{ hatPruefer}$

In FOL: (P... Prüfung, h...hatPruefer)

$$(\forall x)(P(x) \rightarrow \neg(\exists x_1)(\exists x_2)(\exists x_3)(x_1 \neq x_2 \wedge x_2 \neq x_3 \wedge x_1 \neq x_3 \wedge h(x,x_1) \wedge h(x,x_2) \wedge h(x,x_3)))$$

Entsprechend für die anderen Zahlenrestriktionen

OWL ALS SHOIN(D) - ROLLEN



`rdfs:subPropertyOf`

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

Entsprechend Rollenäquivalenz

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

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

transitive Rollen: $\text{Trans}(R)$

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

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 SHOIN(D) - ÜBERBLICK



Erlaubt sind:

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

DLs - NOMENKLATUR

AIFB 

- \mathcal{ALC} : Attribute Language with Complement
- S : \mathcal{ALC} + Rollentransitivität
- \mathcal{H} : Subrollenbeziehung
- O : abgeschlossene Klassen
- 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

- OWL DL ist $S\mathcal{H}OIN(D)$
- OWL Lite ist $S\mathcal{H}IF(D)$

DL-SYNTAX - ÜBERSICHT

AIFB 

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

Roles		
I	Atomic	R
	Inverse	R^-

Ontology (=Knowledge Base)		
Concept Axioms (TBox)		
Subclass	$C \sqsubseteq D$	
Equivalent	$C \equiv D$	
Role Axioms (RBox)		
SH	Subrole	$R \sqsubseteq S$
	Transitivity	$\text{Trans}(S)$
Assertional Axioms (ABox)		
Instance	$C(a)$	
Role	$R(a, b)$	
Same	$a = b$	
Different	$a \neq b$	

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

DL-SYNTAX - KLASSENKONSTRUKTOREN



Constructor	DL Syntax	Example	FOL Syntax
intersectionOf	$C_1 \sqcap \dots \sqcap C_n$	Human \sqcap Male	$C_1(x) \wedge \dots \wedge C_n(x)$
unionOf	$C_1 \sqcup \dots \sqcup C_n$	Doctor \sqcup Lawyer	$C_1(x) \vee \dots \vee C_n(x)$
complementOf	$\neg C$	\neg Male	$\neg C(x)$
oneOf	$\{x_1\} \sqcup \dots \sqcup \{x_n\}$	{john} \sqcup {mary}	$x = x_1 \vee \dots \vee x = x_n$
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	$\leq nP$	≤ 1 hasChild	$\exists \leq n y.P(x, y)$
minCardinality	$\geq nP$	≥ 2 hasChild	$\exists \geq n y.P(x, y)$

Beliebig komplexes Schachteln von Konstruktoren erlaubt:

Person $\sqcap \forall$ hasChild.(Doctor $\sqcup \exists$ 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
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	$\top \sqsubseteq \leq 1P$	$\top \sqsubseteq \leq 1$ hasMother
inverseFunctionalProperty	$\top \sqsubseteq \leq 1P^-$	$\top \sqsubseteq \leq 1$ hasSSN ⁻

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

AIFB 

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.

	<i>Are all children of Bill male?</i>	<i>No idea, since we do not know all children of Bill.</i>	<i>If we assume that we know everything about Bill, then all of his children are male.</i>
child(Bill,Bob)		DL answers	Prolog
Man(Bob)	? $\models \forall \text{child.Man(Bill)}$	don't know	yes
$\leq 1 \text{ child.T(Bill)}$? $\models \forall \text{child.Man(Bill)}$	yes	<i>Now we know everything about Bill's children.</i>

AGENDA

AIFB 

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

WICHTIGE INFERENZPROBLEME



- Globale Konsistenz der Wissensbasis $KB \models \text{false?}$
 - Ist Wissensbasis sinnvoll?
- Klassenkonsistenz $C \equiv \perp?$
 - Muss Klasse C leer sein?
- Klasseninklusion (Subsumption) $C \sqsubseteq D?$
 - Strukturierung der Wissensbasis
- Klassenäquivalenz $C \equiv D?$
 - Sind zwei Klassen eigentlich dieselbe?
- Klassendisjunktheit $C \sqcap D = \perp?$
 - Sind zwei Klassen disjunkt?
- Klassenzugehörigkeit $C(a)?$
 - Ist Individuum a in der Klasse C?
- Instanzgenerierung (Retrieval) „alle X mit C(X) finden“
 - Finde alle (bekanntesten!) 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.
- → Rückführung der Inferenzprobleme auf das Finden von Inkonsistenten in der Wissensbasis, d.h. zeigen der Unerfüllbarkeit der Wissensbasis!

RÜCKFÜHRUNG AUF UNERFÜLLBARKEIT

AIFB

- **Klassenkonsistenz** $C \equiv \perp$ 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** $C \sqcap D = \perp$ 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!

AGENDA

AIFB 

- Beschreibungslogiken
- \mathcal{ALC}
- OWL als $\mathcal{SHOIN}(\mathcal{D})$
- 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

AIFB 

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

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

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

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

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

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

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

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

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

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

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

W und $\text{NNF}(W)$ sind logisch äquivalent.

TABLEAU - NNF - BEISPIEL

AIFB 

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

NAIVES TABLEAUFERFAHREN



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

AIFB 

$C(a)$

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

$\neg C(a)$ 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

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

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
 $C(a)$, $\neg C(a)$, $R(a,b)$.
- *Tableau*: Endliche Menge von Tableauzweigen.
- Tableauzweig ist *abgeschlossen* wenn er ein Paar widersprüchlicher Aussagen $C(a)$ und $\neg C(a)$ enthält.
- Tableau ist *abgeschlossen*, wenn jeder Zweig von ihm abgeschlossen ist.

TABLEAU - ERZEUGUNG

AIFB 

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 $C(a)$ für 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)

AIFB 

- 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)

AIFB 

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

Tableau:

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

$P(a)$

$\neg E(a)$

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

$\neg P(a)$

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

$(E \sqcap U)(a)$

$(E \sqcap \neg D)(a)$

$E(a)$

$E(a)$

$U(a)$

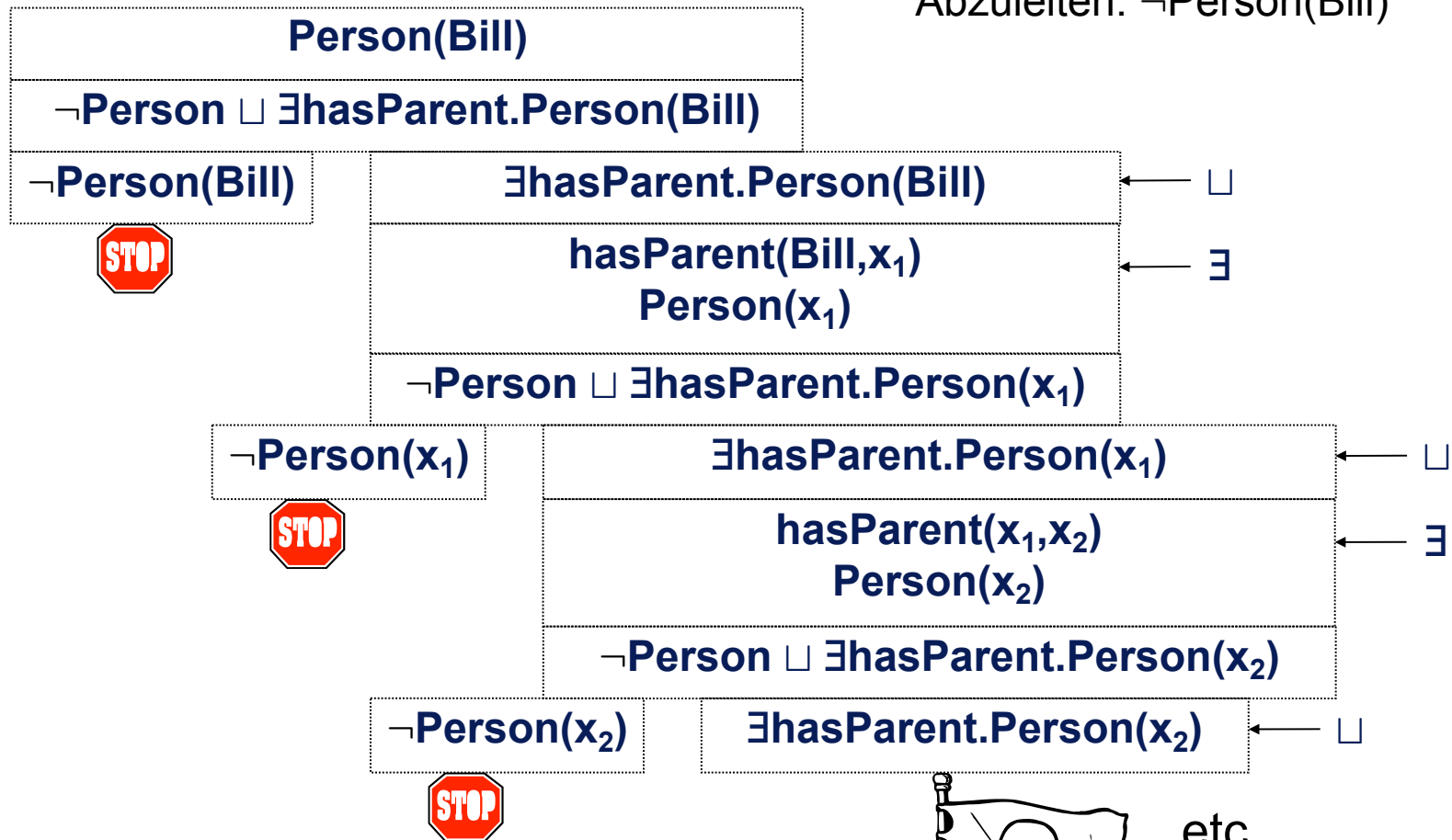
$\neg D(a)$

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

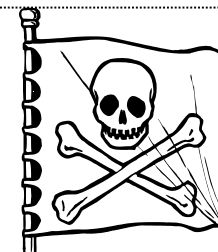
TABLEAU - TERMINIERUNGSPROBLEM

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

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



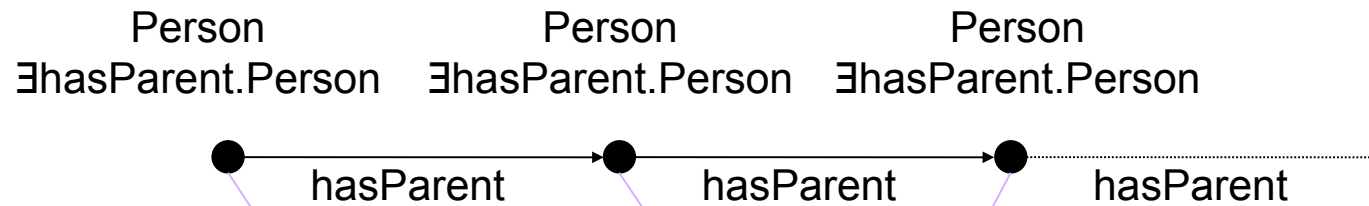
**Problem tritt auf durch
Existenzquantoren (und minCardinality)**



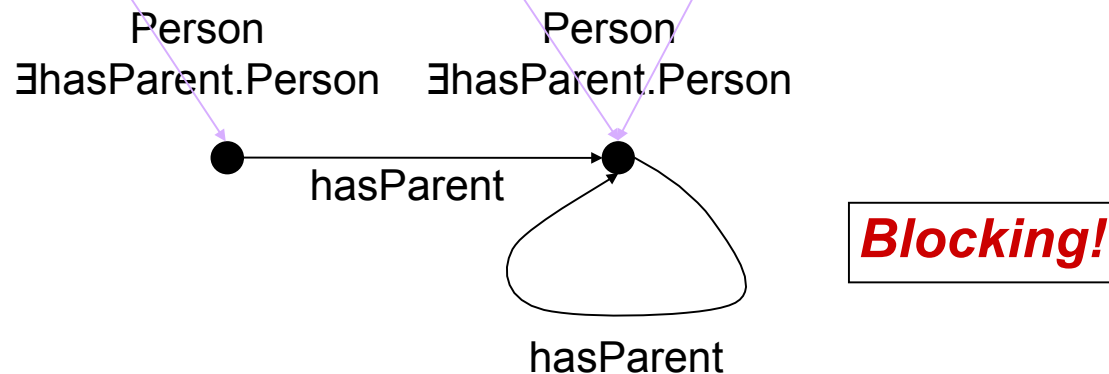
etc.

TABLEAU - BLOCKING - IDEE

AIFB  Wir haben folgendes konstruiert:



Folgendes wäre aber auch denkbar:

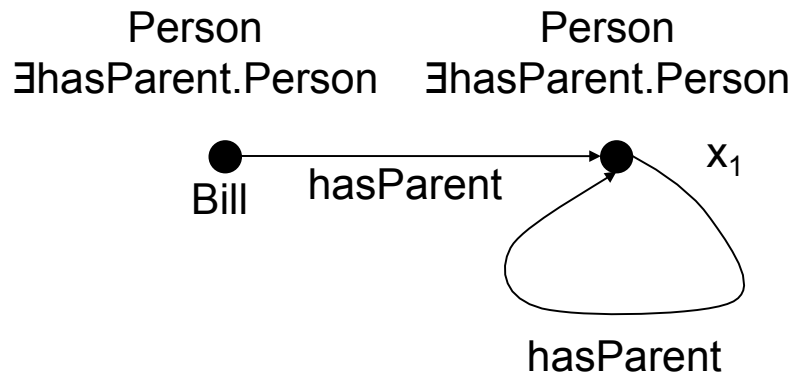
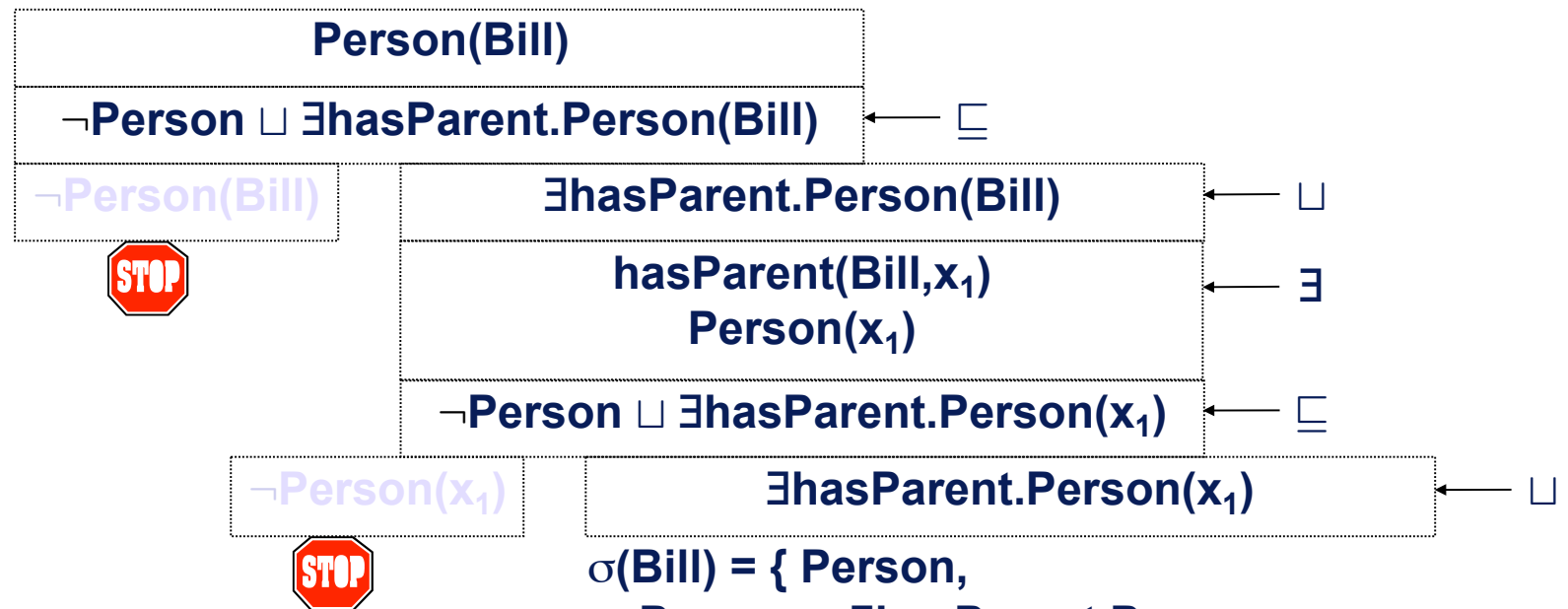


D.h. Wiederverwendung alter Knoten!

Es muss natürlich formal nachgewiesen werden, dass das ausreicht!

TABLEAU MIT BLOCKING

- Einziges Axiom: $\neg \text{Person} \sqcup \exists \text{hasParent. Person}$
Abzuleiten: $\neg \text{Person}(\text{Bill})$



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



TABLEAU - BLOCKING - DEFINITION



Die Auswahl von $(\exists R.C)(a)$ im Tableauxzweig 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 mit Blocking ist $2NExptime!$
→ schlechter als nötig!

TABLEAU-BEWEISER

AIFB

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