
Informatik-Seminar zum Thema
Service-orientierte Architektur (SOA)

**Monotone und nichtmonotone Regeln im Semantic Web:
Problemstellung und Lösungsansätze**

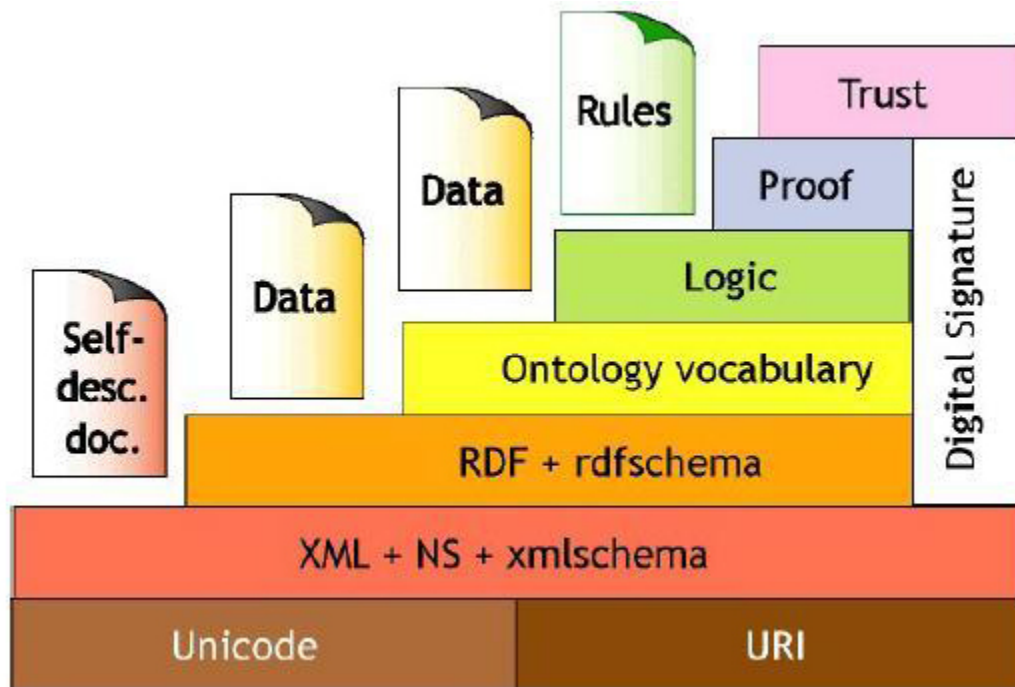
Louisa Bechtel
(mi2249)

Überblick

- Einführung
- Prädikatenlogik / First Order Logik (FOL)
- Horn-Logik
- Monotone Regeln
- Description Logic Programs (DLP)
- Nichtmonotone Regeln
- Umfassendes Beispiel zu nichtmonotonen Regeln
- Rule Markup Language

Einführung

Die Architektur des Semantic Web



Semantic Web Tower

Logikebene:

- Regelbasierte Systeme
- Regeln zur Verarbeitung der in der Ontologieschicht beschriebenen Inhalte definiert
- Ausgeführt in der darüber liegenden Schicht (Proof-Layer)

Einführung

Beispiel:

Eine Suchanfrage nach einem Autor eines Buches richtet sich an zwei Datenbanken. In der einen Datenbank wird das Wort 'Autor' zur Bezeichnung des Autors verwendet, in der anderen das Wort 'creator'.

Im Programm selbst gibt es keine Regel die besagt, dass 'Autor' und 'creator' die selbe Information enthält.

Aufgabe der Logik:

- Verknüpfung von Aussagen
- Schlussfolgerung aus gegebenen Aussagen
- Führen formaler Beweise anhand von Schlussregeln

} Repräsentation von Wissen

Prädikatenlogik / First Order Logic (FOL)

Syntax:

- Variablen: x, y
- Prädikatensymbole: $P(x,y)$
- Funktionen: $f(x,y)$
- Konstanten (nullstellige Funktionen): a,b
- Quantoren:
 - Existenzquantor: $\exists x(\dots)$
 - Allquantor: $\forall x(\dots)$
- Prädikatenlogische Formeln:
 - Verknüpfung von Variablen, Funktionen und Prädikaten mit aussagenlogischen Operatoren oder Quantoren
 - Es gibt freie und durch Quantoren gebundene Variablen
 - $\exists x \forall y P(y, f(x, z)) \rightarrow x, y$ sind gebunden, z ist frei

Prädikatenlogik / First Order Logic (FOL)

Beispiele:

- Eine Person ist entweder eine Frau oder ein Mann
 - $\text{Person}(x) \rightarrow (\text{Frau}(x) \vee \text{Mann}(x))$
- Wenn Alex eine Person ist, dann ist Alex keine Frau
 - $\text{Person}(\text{Alex}) \rightarrow \neg \text{Frau}(\text{Alex})$
- Erika ist Mutter von Alex
 - $\text{Mutter}(\text{Erika}, \text{Alex})$

Prädikatenlogik / First Order Logic (FOL)

Problem der Prädikatenlogik:

Die Prädikatenlogik ist zwar auf Objektebene sehr ausdruckskräftig, allerdings auch sehr allgemein, was dazu führt, dass sie nicht entscheidbar ist.

Das heißt, dass es nicht möglich ist zu schließen, ob ein Faktum eines Sachverhaltes gilt, nicht gilt oder gelten kann.

→ Einführung der Horn-Logik als Teilmenge der Prädikatenlogik

Horn Logik

Syntax:

Aufbau einer Regel:

$$B_1, \dots, B_n \rightarrow A$$

- A, B_1, \dots, B_n : atomare Formeln
- A : Kopf
- $\{B_1, \dots, B_n\}$: Körper

- Die Kommas im Körper der Regel repräsentieren Konjunktionen
 - Wenn B_1 und B_2 und ... und B_n wahr sind, ist auch A wahr.
- Variablen dürfen in A, B_1, \dots, B_n auftreten

$\text{studies}(X, Y), \text{lives}(X, Z), \text{loc}(Y, U), \text{loc}(Z, U) \rightarrow \text{homeStudent}(X)$

Horn Logik

Vergleich mit Beschreibungssprachen (OWL):

Beschreibungssprachen (OWL) und Horn-Logik sind orthogonal zueinander.

in Horn-Logik möglich, in OWL nicht:

$\text{studies}(X,Y), \text{lives}(X,Z), \text{loc}(Y,U), \text{loc}(Z,U) \rightarrow \text{homeStudent}(X)$

in OWL möglich, in Horn-Logik nicht:

z.B. die Bildung von Gegensätzen, bestimmte Arten von Bewertungen

Monotone Regeln

Syntax:

Rules:

$$B_1, \dots, B_n \rightarrow A$$

- A, B_1, \dots, B_n : atomare Formeln
- A : Kopf
- $\{B_1, \dots, B_n\}$: Körper

Regeln können als folgende Formel dargestellt werden:

$$\forall x_1 \dots \forall x_k ((B_1 \wedge \dots \wedge B_n) \rightarrow A)$$

oder

$$\forall x_1 \dots \forall x_k (A \vee \neg B_1 \vee \dots \vee \neg B_n)$$

x_1, \dots, x_k : Variablen die in A, B_1, \dots, B_n auftreten können

Monotone Regeln

Syntax:

Facts:

atomare Formel

Logic Programs:

begrenzte Menge von Facts und Rules

Goals:

Anfrage $(B_1, \dots, B_n \rightarrow)$ an ein Logic Program

Darstellung als Formel:

$$\forall x_1 \dots \forall x_k (\neg B_1 \vee \dots \vee \neg B_n)$$

Monotone Regeln

Beispiel:

$treuerKunde(X), Alter(X) > 60 \rightarrow Preisnachlass(X)$

Bestandteile:

- Variable: X
- Konstante: 60
- Prädikat: $treuerKunde$
- Funktionssymbol: $Alter$

Beispiel für ein Fact:

$treuerKunde(a12345)$

Bedeutung: Der Kunde mit der Kundennummer a12345 ist ein treuer Kunde

Description Logic Programs (DLP)

Description Logic:

- Untermenge der Prädikatenlogik
- Definitionen und Eigenschaften leicht beschreibbar
- Beschreibung der Zusammenhänge zwischen Konzepten

Horn-Logik und DL sind orthogonal.

Die Schnittmenge von OWL und Horn-Logik bilden Description Logic Programs.

Nutzen von DLP:

aus Entwicklersicht: es ist egal, ob mit OWL oder mit Logikregeln gearbeitet wird

aus Implementationssicht: Flexibilität und Gewährleistung von dem Bestehen gemeinsamer Standards

Description Logic Programs (DLP)

RDF in Horn-Logik ausgedrückt:

Instanz: $\text{type}(a,C) \quad C(a)$

C als subclass von D: $C(X) \rightarrow D(X)$

OWL in Horn-Logik ausgedrückt:

sameClassAs(C,D) $C(X) \rightarrow D(X)$

$D(X) \rightarrow C(X)$

Transitivität

samePropertyAs(X,Z) $P(X,Y), P(Y,Z) \rightarrow P(X,Z)$

Nichtmonotone Regeln

Beispiel:

r1: Wenn Geburtstag, dann besonderer Rabatt

r2: Wenn nicht Geburtstag, dann kein besonderer Rabatt

r1: Wenn Geburtstag, dann besonderer Rabatt

r2': Wenn Geburtstag nicht bekannt, dann kein besonderer Rabatt

Nichtmonotone Regeln: Regeln die verletzt werden können
Menschen schließen nichtmonoton!

Nichtmonotone Regeln

Monotone Schlussfolgerung:

1)

Alle Hühner sind Vögel,

Alle Vögel legen Eier,

Tweety ist ein Huhn.

↳ Tweety legt Eier

$\text{Huhn}(x) \rightarrow \text{Vogel}(x)$

$\text{Vogel}(x) \rightarrow \text{legtEier}(x)$

$\text{Huhn}(\text{Tweety})$

↳ $\text{legt_Eier}(\text{Tweety})$

2)

Tweety ist ein Vogel,

Vögel können fliegen,

↳ Tweety kann fliegen

$\forall x: \text{Vogel}(x) \rightarrow \text{fliegt}(x)$

Weitere Fakten:

Pinguine sind Vögel.

Pinguine können nicht fliegen.

Tweety ist ein Pinguin.

Nichtmonotone Regeln

Syntax:

$r: L_1, \dots, L_n \Rightarrow L$

- \Rightarrow : zur Hervorhebung von nichtmonotonen Regeln
- r : label
- $\{L_1, \dots, L_n\}$: Körper / Prämissen
- L : Kopf
- L, L_1, \dots, L_n : positive oder negative Literale
- keine Funktionssymbole

Nichtmonotone Regeln

Syntax:

Prioritäten zwischen den Regeln:

$r_1 > r_2$
 r_1 hat eine höhere Priorität als r_2 ;
es dürfen keine Zirkelschlüsse auftreten

Prioritäten werden z.B. nach folgenden Gesichtspunkten vergeben:

- Sicherheit der Quellen
- Aktualität der Regeln
- spezielle Regeln vorgezogen vor allgemeineren Regeln

Logikprogramm nichtmonotoner Regeln: Darstellung durch ein Tripel $(F,R,>)$

F: Menge von Facts

R: begrenzte Menge von nichtmonotonen Regeln

>: binäre, nicht zyklische Relation der Regeln

Umfassendes Beispiel: nichtmonotone Regeln

Beispiel: Wohnungsvermietung

Ansprüche an die Wohnung:

- mindestens 45qm Größe
- mindestens 2 Schlafzimmer
- Wenn die Wohnung im dritten Stock oder höher liegt, muss ein Fahrstuhl vorhanden sein
- Haustiere müssen erlaubt sein
- 300€ würden für eine zentral liegende Wohnung ausgegeben werden,
- 250€ für eine dezentral liegende Wohnung
- 5€ pro Quadratmeter würden gezahlt werden, wenn die Wohnung größer wäre,
- 2€ pro Quadratmeter für einen Garten
- Es können nicht mehr als 400€ gesamt gezahlt werden

Prioritäten:

- 1) Preisgünstigste Wohnung
- 2) Garten ist vorhanden
- 3) Zusätzlicher Platz

Umfassendes Beispiel: nichtmonotone Regeln

Prädikate um die Eigenschaften der Wohnung zu beschreiben:

- $\text{Groesse}(x,y)$ y ist die Größe von x in qm
- $\text{Schlafzimmer}(x,y)$ x hat y Schlafzimmer
- $\text{Preis}(x,y)$ y ist der Preis für x
- $\text{Stockwerk}(x,y)$ x ist im y ten Stock
- $\text{Garten}(x,y)$ x hat einen Garten der Größe y
- $\text{Fahrstuhl}(x)$ x hat einen Fahrstuhl
- $\text{Haustiere}(x)$ Haustiere sind in x erlaubt
- $\text{zentral}(x)$ x liegt zentral

zusätzliche Prädikate:

- $\text{akzeptabel}(x)$ x genügt den Ansprüchen
- $\text{Angebot}(x,y)$ y € werden für die Wohnung x bezahlt

Umfassendes Beispiel: nichtmonotone Regeln

Anforderungen:

r1: \Rightarrow akzeptabel(X)

r2: Schlafzimmer(X,Y), $Y < 2 \Rightarrow \neg$ akzeptabel(X)

r3: Groesse(X,Y), $Y < 45 \Rightarrow \neg$ akzeptabel(X)

r4: \neg Haustiere(X) $\Rightarrow \neg$ akzeptabel(X)

r5: Stockwerk(X,Y), $Y > 2$, \neg Fahrstuhl(X) $\Rightarrow \neg$ akzeptabel(X)

r6: Preis(X,Y), $Y > 400 \Rightarrow \neg$ akzeptabel(X)

Prioritäten festlegen:

$r2 > r1$, $r3 > r1$, $r4 > r1$, $r5 > r1$, $r6 > r1$

Umfassendes Beispiel: nichtmonotone Regeln

Weitere Einschränkungen:

r7: $\text{Groesse}(X,Y), Y \geq 45, \text{Garten}(X,Z), \text{zentral}(X) \Rightarrow \text{Angebot}(X, 300 + 2Z + 5(Y - 45))$

r8: $\text{Groesse}(X,Y), Y \geq 45, \text{Garten}(X,Z), \neg\text{zentral}(X) \Rightarrow \text{Angebot}(X, 250 + 2Z + 5(Y - 45))$

r9: $\text{Angebot}(X,Y), \text{Preis}(X,Z), Y < Z \Rightarrow \neg\text{akzeptabel}(X)$

r9 > r1

Umfassendes Beispiel: nichtmonotone Regeln

Mögliche Wohnungen:

Wohnung	Schlafzimmer	Größe	Zentral	Stockwerk	Fahrstuhl	Haustiere	Garten	Preis
A1	1	50	Ja	1	Nein	Ja	0	300
A2	2	45	Ja	0	Nein	Ja	0	335
A3	2	65	Nein	2	Nein	Ja	0	350
A4	2	55	Nein	1	Ja	Nein	15	330
A5	3	55	Ja	0	Nein	Ja	15	350
A6	2	60	Ja	3	Nein	Nein	0	370
A7	3	65	Ja	1	Nein	Ja	12	375

Umfassendes Beispiel: nichtmonotone Regeln

Repräsentation der Wohnungen:

Beispiel A1:

- Schlafzimmer(A1,1)
- Groesse(A1,50)
- zentral(A1)
- Stockwerk(A1,1)
- \neg Fahrrad(A1)
- Haustiere(A1)
- Garten(A1,0)
- Preis(A1,300)

Die Wohnung A1 entspricht nicht den Anforderungen, weil sie nur ein Schlafzimmer hat.

[Verstoß gegen

r2: Schlafzimmer(X,Y), $Y < 2 \Rightarrow \neg$ akzeptabel(X)]

Umfassendes Beispiel: nichtmonotone Regeln

Mögliche Wohnungen:

Wohnung	Schlafzimmer	Größe	Zentral	Stockwerk	Fahrstuhl	Haustiere	Garten	Preis
A1	1	50	Ja	1	Nein	Ja	0	300
A2	2	45	Ja	0	Nein	Ja	0	335
A3	2	65	Nein	2	Nein	Ja	0	350
A4	2	55	Nein	1	Ja	Nein	15	330
A5	3	55	Ja	0	Nein	Ja	15	350
A6	2	60	Ja	3	Nein	Nein	0	370
A7	3	65	Ja	1	Nein	Ja	12	375

Ausgrenzung:

- Die Wohnungen A4 und A6 sind nicht akzeptabel, da keine Haustiere erlaubt sind (Verstoß gegen r4)
- A2 ist nicht akzeptabel, da der Preis zu hoch ist (Verstoß gegen r7 und r9)

Die Wohnungen A3, A5 und A7 genügen den Anforderungen.

Umfassendes Beispiel: nichtmonotone Regeln

Präferenzen:

r10: akzeptabel(X), Preis(X,Z), not(akzeptabel(Y), Preis(Y,W), $W < Z$)

⇒ guenstigste(X)

r11: guenstigste(X), Gartengroesse(X,Z), not(guenstigste(Y), Gartengroesse(Y,W), $W > Z$)

⇒ groessterGarten(X)

r12: groessterGarten(X), Groesse(X,Z), not(groessterGarten(Y), Groesse(Y,W), $W > Z$)

⇒ Mieten(X)

Umfassendes Beispiel: nichtmonotone Regeln

Mögliche Wohnungen:

Wohnung	Schlafzimmer	Größe	Zentral	Stockwerk	Fahrstuhl	Haustiere	Garten	Preis
A1	1	50	Ja	1	Nein	Ja	0	300
A2	2	45	Ja	0	Nein	Ja	0	335
A3	2	65	Nein	2	Nein	Ja	0	350
A4	2	55	Nein	1	Ja	Nein	15	330
A5	3	55	Ja	0	Nein	Ja	15	350
A6	2	60	Ja	3	Nein	Nein	0	370
A7	3	65	Ja	1	Nein	Ja	12	375

- guenstigste(A3)
- guenstigste(A5)
- groesste(A3)
- groesste(A7)
- groessterGarten(A5)

A5 wird gemietet:

- r11 (groessterGarten) gilt für A5
- r10 (guenstigste) gilt für A3 und A5

Rule Markup Language (RuleML)

Beschreibung:

- bisher kein W3C Standard
- soll als standardisiertes Dokumentenformat für Regeln dienen
- unter der Benutzung können monotone Regeln in XML beschrieben werden
- verbindet verschiedene Typen von Regelsprachen miteinander
- Der Kern basiert auf Horn Logik ohne Verwendung von Funktionen

Rule Markup Language (RuleML)

Beispiel:

„Für einen Premium-Kunden
der ein Luxus-Produkt kauft
beträgt der Preisnachlass 7,5 Prozent.“

Horn Logik Bestandteil	RuleML
Faktum	Fact
Regel	Implies
Kopf	Head
Prämissen	Body
Atomare Formel	Atom
Konjunktion von atomaren Formeln	And
Prädikat	Rel
Konstante	Ind
Variable	Var

```
<Implies>
  <head>
    <Atom>
      <Rel>Preisnachlass</Rel>
      <Var>Kunde</Var>
      <Var>Produkt</Var>
      <Ind>7.5 Prozent</Ind>
    </Atom>
  </head>
  <body>
    <And>
      <Atom>
        <Rel>premium</Rel>
        <Var>Kunde</Var>
      </Atom>
      <Atom>
        <Rel>luxus</Rel>
        <Var>Produkt</Var>
      </Atom>
    </And>
  </body>
</Implies>
```

Fazit

- Mit Hilfe des Semantic Web soll das Wissen in einer „maschinen-erreichbaren“ Art und Weise unter der Benutzung von Web Languages dargestellt werden
- Logik als Grundlage zur Darstellung Information
- Horn Logik (eine Teilmenge der Prädikatenlogik) bildet die Basis von monotonen Regeln
- In Description Logik Programs wird OWL mit Horn Logik (monotonen Regeln) verbunden
- Nichtmonotone Regeln sind nötig, wenn die Informationen unvollständig sind
 - sie können von anderen Regeln bei Bedarf überschrieben werden
 - das Überschreiben wird durch die Vergabe von Prioritäten geregelt

Vielen Dank für die Aufmerksamkeit!