Konjunktive Anfragen und Regelsprachen

Pascal Hitzler Markus Krötzsch Sebastian Rudolph

Institut AIFB · Universität Karlsruhe

Semantic Web Technologies 1 (WS08/09)
14. Januar 2009
http://semantic-web-grundlagen.de

Semantic Web Technologies 1

1. Einleitung und Ausblick
2. XML und URIs
3. Einführung in RDF
4. RDF Schema
5. Logik – Grundlagen
6. Semantik von RDF(S)
7. OWL – Syntax und Intuition
8. OWL – Semantik und Reasoning
9. SPARQL – Syntax und Intuition
10. Semantik von SPARQL
11. **Konjunktive Anfragen/Einführung Regelsprachen** (→ Webseite)
12. OWL 2
13. Bericht aus der Praxis
14. Regeln für OWL
15. Semantic Web – Anwendungen

Literaturhinweise siehe → Semantic Web – Grundlagen, Kapitel 7

Die Grenzen von OWL

OWL-Konzepte als Anfragesprache ungenügend:

- „Welche Paare von Personen haben ein gemeinsames Elternteil?“
- „Welche Personen wohnen bei einem ihrer Eltern?“
- „Welche Paare (direkter oder indirekter) Nachkommen gibt es?“

Relevante Informationen nicht in OWL-Ontologie darstellbar:

- „$(\forall x)(\forall y)(\forall z)\ (\text{bruder}(y,z) \wedge \text{vater}(x,y) \rightarrow \text{onkel}(x,z))$“
- „$(\forall x)\ (\text{liebt}(x,x) \rightarrow \text{Narzist}(x))$“

OWL ungeeignet zur Programmierung:

- OWL ist entscheidbar: es kann grundsätzlich nicht alles Programmierbare ausdrücken (*Halteproblem*).
- OWL wird nicht „abgearbeitet“, es ist *nicht prozedural*: Bestimmte Erweiterungen (Built-ins) sind nur schwer zu realisieren.

Übersicht über nächsten Vorlesungen

Vorlesung 11:

- Ausdrucksstarke Anfragen für OWL
  ⇝ **Konjunktive Anfragen**
- Erweiterung von OWL zur Wissensrepräsentation
  ⇝ **Regelsprachen, SWRL**

Vorlesung 12:

- Weiterentwicklung von OWL
  ⇝ **OWL 2**

Vorlesung 14:

- Regelerweiterungen für OWL
  ⇝ **Regeln sinnvoll einschränken**

Semantic Web Technologies 2:

- Regel austauschen im Semantic Web
  ⇝ **Rule Interchange Format (RIF)**
- Logikprogrammierung als semantische Technologie
  ⇝ **F-Logik**

Ausdrucksstarke Anfragen für OWL?

SPARQL als Anfragesprache für RDF
$\rightsquigarrow$ keine direkte Unterstützung für OWL

Anfrageformalismus für OWL DL: **konjunktive Anfragen**

- keine offizielle Spezifikation, keine normative Syntax
  $\rightsquigarrow$ uns reichen hier Kurzschreibweisen anstelle von vollen URIs
- Ziel: ausdrucksstärkere Anfragen nach *Individuen*
- keine Betrachtung von Formatierung oder Nachbearbeitung der Ergebnisse
- praktische Bedeutung für Anwendungen
- verschiedene Implementationen verfügbar

Syntax und Intuition

Konjunktive Anfragen sind recht einfach!

Beispiel:

$$\texttt{Buch}(x) \wedge \texttt{VerlegtBei}(x, \texttt{Springer}) \wedge \texttt{Autor}(x, y)$$

„Welche Bücher sind bei Springer erschienen und wer hat sie geschrieben?“

- Syntax angelehnt an Prädikatenlogik
- Hauptelemente: Bezeichner von Rollen/Klassen/Individuen, Variablen, Konjunktion $\wedge$

Semantik konjunktiver Anfragen

Konjunktive Anfragen ähneln logischen Formeln
$\rightsquigarrow$ Anfragen ohne Variablen können aus einer Ontologie *folgen*
$\rightsquigarrow$ Variablen als Platzhalter für Bezeichner von Individuen

Funktion μ is **Lösung einer konjunktiven Anfrage** q für eine Ontologie O, falls gilt:

1. Domäne von μ ist die Menge der freien Variablen in q
2. Wertebereich von μ ist die Menge der Individuenbezeichner in O
3. $O \models \mu(q)$, d.h. q mit dieser Variablenbelegung folgt aus O

$\rightsquigarrow$ keine partielle Funktion – alle Variablen müssen belegt sein
$\rightsquigarrow$ Literale (Datentypen) hier zur Vereinfachung nicht betrachtet

Unbenannte Elemente

Variablen bisher als Platzhalter für (benannte) Individuen,

aber

OWL-Ontologien können auch die **Existenz unbenannter Elemente** implizieren

Beispiel:

$\texttt{Buch}(a)$	(a ist ein Buch)
$\texttt{Buch} \sqsubseteq \exists \texttt{Autor}.\top$	(jedes Buch hat einen Autor)

$\rightsquigarrow$ Anfrage $\texttt{Buch}(x) \wedge \texttt{Autor}(x, y)$ hat keine Lösung!

Unbestimmte Variablen

Wie können Anfragen auch unbenannte Individuen berücksichtigen?

- unbenannte Elemente können kaum als Teil der Lösung ausgegeben werden
- wir wollen nur die *Existenz* geeigneter Elemente fordern

$\rightsquigarrow$ **Unbestimmte Variablen** werden durch Existenzquantoren gebunden

Beispiel:

$\texttt{Buch}(a)$	(a ist ein Buch)
$\texttt{Buch} \sqsubseteq \exists \texttt{Autor}.\top$	(jedes Buch hat einen Autor)

Anfrage $\exists y.(\texttt{Buch}(x) \wedge \texttt{Autor}(x, y))$

$\rightsquigarrow$ Lösung $\{x \mapsto a\}$, aber y nicht Teil der Lösung

Vergleich mit SPARQL

SPARQL	**konjunktive Anfragen**
Muster in Graphen	logische Konjunktionen
ein kanonisches Modell	viele mögliche Modelle
Optionen, Alternativen, Filter	—
Abfrage beliebiger Elemente, z.B. von Property-Bezeichnern	nur Abfrage von Instanzen (strikte Typung)
Variablen für beliebige Elemente	Bestimmte und unbestimmte Variablen

„SPARQL für OWL“ ist möglich:

- Darstellung logischer Konsequenzen als Graph
- Typung wie bei OWL DL, ev. Erweiterung um in OWL übliche Anfragen (z.B. Klassenhierarchie)
- Inkompatibilitäten bei Variablensemantik müssen akzeptiert werden

Variablen in SPARQL und in konjunktiven Anfragen

Kennzeichen verschiedener Arten von Anfragevariablen:

- Unbenannte Werte: Sind Werte möglich, die keine Bezeichner (URI/Literal) haben?
- Ausgabe: Erscheint die Variable in Lösungen der Anfrage?

Unterschiedliche Formen von Variablen:

	Unbenannte Werte	Ausgabe
Bestimmte Variable	—	Ja
Unbestimmte Variable	Ja	—
SPARQL-Variable	Ja	Ja
Leerer Knoten in SPARQL	Ja	—
Nicht ausgewählte SPARQL-Variable	Ja	—

$\rightsquigarrow$ „SPARQL für OWL“:
leere Knoten der Ontologie: Individuen oder unbenannte Elemente?

Erweiterungen konjunktiver Anfragen

Mögliche Erweiterungen konjunktiver Anfragen:

- **Filter, Modifikatoren, Ergebnisformate:** Definition wie in SPARQL möglich, unabhängig von Anfrage (Filter unproblematisch wenn kein `OPTIONAL`)
- **Negation:** Zulassung von $\neg$ vor Anfrageausdrücken
- **Disjunktionen:** Zulassung von $\vee$, entspricht `UNION` in SPARQL, „disjunktive Anfragen“
- **Komplexe Pfadausdrücke:** Reguläre Ausdrücke zur Beschreibung von Mustern aus Rollen, z.B. Anfrage nach allen Vorfahren einer Person (enthält beliebig lange Kette aus Rolle „KindVon“)

Komplexität und Entscheidbarkeit

Schlussfolgern mit OWL DL ist sehr komplex (NExpTime-vollständig)
⇝ Wie schwierig sind dann konjunktive Anfragen?

Bisher noch nicht abschließend geklärt!

- Konjunktive Anfragen für $\mathcal{SHIQ}$ (und für OWL Lite): 2ExpTime-vollständig!
- Konjunktive Anfragen für $\mathcal{SHOQ}$: entscheidbar in 2ExpTime
- Konjunktive Anfragen für $\mathcal{SHOIQ}$ (und für OWL DL): Entscheidbarkeit nicht bekannt! (13.01.2009)

⇝ konjunktive Anfragen für OWL sind äußerst kompliziert

Implementationen

Implementationen von konjunktiven Anfragen für OWL verfügbar (13.01.2009)

- **KAON2**: konjunktive Anfragen ohne unbestimmte Variablen, eingeschränkte Negation zulässig (→ Webseite)
- **Pellet**: konjunktive Anfragen mit unbestimmten Variablen und Negationen, nicht vollständig für OWL DL (→ Webseite)
- weitere Systeme mit speziellen Anfragesprachen (**RacerPro**) oder Beschränkung auf einfachere DLs (**QuOnto** für „OWL QL“)

⇝ Einschränkung des Problems für bessere Implementierbarkeit

Was sind Regeln?

1. Logische Regeln (Implikationen in Prädikatenlogik):
   - „$F \rightarrow G$“ (≡ „$\neg F \vee G$“)
   - Logische Erweiterung der Wissensbasis ⇝ **statisch**
   - Open World
   - **Deklarativ** (beschreibend)
2. Prozedurale Regeln (z.B. Production Rules):
   - „*If X then Y else Z*“
   - Ausführbare Maschinen-Anweisungen ⇝ **dynamisch**
   - **Operational** (Bedeutung = Effekt bei Ausführung)
3. Logikprogrammierung (z.B. Prolog, F-Logik):
   - „`mann(X) <- person(X) AND NOT frau(X)`“
   - Approximation logischer Semantik mit operationalen Aspekten, Built-ins möglich
   - häufig Closed World
   - **„Semi-deklarativ“**
4. Ableitungsregeln eines Kalküls (z.B. Regeln zur RDF-Semantik)
   - Regeln nicht als Teil der Wissensbasis, „Meta-Regeln“
     ⇝ nicht Thema dieser Vorlesung

Welche Regelsprache?

Regelsprachen sind untereinander kaum kompatibel!
⇝ Wahl der geeigneten Regelsprache sehr wichtig

Mögliche Kriterien:

- Klare Spezifikation von Syntax und Semantik?
- Unterstützung durch Software-Tools?
- Welche Ausdrucksmittel werden benötigt?
- Komplexität der Implementierung? Performanz?
- Kompatibilität mit bestehenden Formaten wie OWL?
- Deklarativ (Beschreiben) oder operational (Programmieren)?
- . . .

Welche Regelsprache?

1. Logische Regeln (Implikationen in Prädikatenlogik):
   - klar definiert, umfassend erforscht, gut verstanden
   - sehr gut kompatibel mit OWL DL und RDF
   - ohne Einschränkungen nicht entscheidbar
2. Prozedurale Regeln (z.B. Production Rules):
   - viele unabhängige Ansätze, oft nur vage definiert
   - Verwendung oft wie Programmiersprachen, Beziehung zu OWL und RDF unklar
   - effiziente Abarbeitung möglich
3. Logikprogrammierung (z.B. Prolog, F-Logik):
   - klar definiert, aber viele unterschiedliche Ansätze
   - teilweise kompatibel mit OWL und RDF
   - Entscheidbarkeit/Komplexität stark vom gewählten Ansatz abhängig

$\rightsquigarrow$ Schwerpunkt dieser Vorlesung: prädikatenlogische Regeln
(die aber auch die Grundlage der Logikprogrammierung sind)

Arten von Regeln

Bezeichnungen für „Regeln“ der Prädikatenlogik:

- **Klausel:** Disjunktion von atomaren Aussagen oder negierten atomaren Aussagen
- **Hornklausel:** Klausel mit *höchstens* einem nicht-negiertem Atom
- **Definite Klausel:** Klausel mit *genau einem* nicht negiertem Atom
- **Fakt:** Klausel aus einem einzigen nicht-negiertem Atom

Beispiele:

$\text{Person}(x)$	$\rightarrow \text{Frau}(x) \vee \text{Mann}(x)$	(Klausel)
$\text{Mann}(x) \wedge \text{hatKind}(x,y)$	$\rightarrow \text{Vater}(x)$	(definite Klausel)
$\text{hatBruder}(\text{mutter}(x),y)$	$\rightarrow \text{OnkelVon}(x,y)$	(Funktionsymbol)
$\text{Mann}(x) \wedge \text{Frau}(x)$	$\rightarrow$	(Hornklausel, „Integritätsbed.“)
	$\text{Frau}(\text{gisela})$	(Fakt)

Prädikatenlogik als Regelsprache

- Regeln als Implikationsformeln der Prädikatenlogik:

$$\underbrace{A_1 \wedge A_2 \wedge \ldots \wedge A_n}_{\text{Rumpf}} \rightarrow \underbrace{H}_{\text{Kopf}}$$

  $\rightsquigarrow$ Semantisch äquivalent zu Disjunktion:

$$H \vee \neg A_1 \vee \neg A_2 \vee \ldots \vee \neg A_n$$

- Konstanten, Variablen und Funktionssymbole erlaubt
- Quantoren für Variablen werden oft weggelassen:
  freie Variablen als universell quatifiziert verstanden
  (d.h. Regel gilt für alle Belegungen)
- Disjunktion mit mehreren nicht-negierten Atomen
  $\rightsquigarrow$ disjunktive Regel:

$$\underbrace{A_1 \wedge A_2 \wedge \ldots \wedge A_n}_{\text{Rumpf}} \rightarrow \underbrace{H_1 \vee H_2 \vee \ldots \vee H_m}_{\text{Kopf}}$$

Datalog

Einschränkung auf Horn-Regeln *ohne* Funktionssymbole $\rightsquigarrow$ **Datalog-Regeln**

Datalog

- logische Regelsprache, ursprünglich Grundlage *deduktiver Datenbanken*
- Wissensbasen („Datalog-Programme“) aus Horn-Klauseln ohne Funktionssymbole
- entscheidbar
- effizient für große *Daten*mengen, Gesamtkomplexität wie OWL Lite (ExpTime)

Semantik von Regeln

Semantik von Regeln:
Standardsemantik der Prädikatenlogik!

- Semantik weithin bekannt und gut verstanden
- mit anderen prädikatenlogischen Ansätzen kompatibel (z.B. Beschreibungslogik)

Semantik von Datalog (Auffrischung)

Semantik definiert über über **logische Modelle:**

- Interpretation $\mathcal{I}$ mit Domäne $\Delta_\mathcal{I}$
- Auswertung von Variablen: Variablenzuweisung $\mathcal{Z}$ (Abbildung von Variablen auf $\Delta_\mathcal{I}$)
- Interpretation von Formeln und Termen unter $\mathcal{I}$ (und $\mathcal{Z}$):
  - Interpretation einer Konstante: $a^{\mathcal{I},\mathcal{Z}} = a^\mathcal{I} \in \Delta_\mathcal{I}$
  - Interpretation einer Variable: $x^{\mathcal{I},\mathcal{Z}} = \mathcal{Z}(x) \in \Delta_\mathcal{I}$
  - Interpretation eines n-stelligen Prädikats: $p^\mathcal{I} \in \Delta_\mathcal{I}^n$
  - $\mathcal{I}, \mathcal{Z} \models p(t_1, \ldots, t_n)$ genau dann wenn $(t_1^{\mathcal{I},\mathcal{Z}}, \ldots, t_n^{\mathcal{I},\mathcal{Z}}) \in p^\mathcal{I}$,
  - $\mathcal{I} \models B \rightarrow H$ genau dann wenn für jede Variablenzuweisung $\mathcal{Z}$ gilt: entweder $\mathcal{I}, \mathcal{Z} \models H$ oder $\mathcal{I}, \mathcal{Z} \not\models B$.
- $\mathcal{I}$ ist ein Modell für eine Regelmenge, wenn gilt: $\mathcal{I} \models B \rightarrow H$ für alle Regeln $B \rightarrow H$ dieser Menge

Logische Folgerung wie in „Obstlogik“ (vgl. Vorlesung 5)

Datalog in der Praxis

Datalog in der Praxis:

- verschiedene Implementierungen verfügbar
- Anpassungen für das Semantic Web: Datentypen aus XML Schema, URIs (z.B. → IRIS)

Erweiterungen von Datalog:

- *disjunktives Datalog* erlaubt Disjunktionen in Köpfen
- nichtmonotone Negation (keine prädikatenlogische Semantik)
- Einbindung von Informationen aus OWL-Ontologien (z.B. → dl-programs, → dlvhex)
  ⇝ lose Kopplung von OWL und Datalog (nicht über gemeinsame prädikatenlogische Semantik)

SWRL

Wie kann man Datalog und OWL DL kombinieren?

SWRL – „Semantic Web Rule Language“

- Vorschlag einer OWL-Regelerweiterung (W3C-Einreichung)
- Idee: Datalog-Regeln mit Bezug zu OWL-Ontologie
- Symbole in Regeln können OWL-Bezeichner sein, oder neue Datalog-Bezeichner
- Zusätzliche *Built-Ins* zur Verarbeitung von Datentypen
- mehrere syntaktische Darstellungen

Semantik von SWRL

OWL DL (Beschreibungslogik) und Datalog verwenden die gleichen Interpretationen:

- OWL-Individuen sind Datalog-Konstanten
- OWL-Klassen sind einstellige Datalog-Prädikate
- OWL-Rollen sind zweistellige Datalog-Prädikate

$\rightsquigarrow \mathcal{I}$ kann gleichzeitig Modell sein für OWL-Ontologie und Menge von Datalog-Regeln
$\rightsquigarrow$ Schlussfolgerung über OWL-Datalog-Kombination möglich

Beispiel

Kombinierte SWRL-Wissensbasis (Datalog + Beschreibungslogik):

(1) $\texttt{Vegetarier}(x) \wedge \texttt{Fischprodukt}(y) \rightarrow \texttt{magNicht}(x,y)$
(2) $\texttt{hatBestellt}(x,y) \wedge \texttt{magNicht}(x,y) \rightarrow \texttt{Unglücklich}(x)$
(3) $\texttt{hatBestellt}(x,y) \rightarrow \texttt{Gericht}(y)$
(4) $\texttt{magNicht}(x,z) \wedge \texttt{Gericht}(y) \wedge \texttt{enthält}(y,z) \rightarrow \texttt{magNicht}(x,y)$
(5) $\rightarrow \texttt{Vegetarier}(\texttt{markus})$
(6) $\texttt{Glücklich}(x) \wedge \texttt{Unglücklich}(x) \rightarrow$
(7) $\exists \texttt{hatBestellt}.\texttt{ThaiCurry}(\texttt{markus})$
(8) $\texttt{ThaiCurry} \sqsubseteq \exists \texttt{enthält}.\texttt{Fischprodukt}$

Wir können folgern: $\texttt{Unglücklich}(\texttt{markus})$

- Komplexität?
- Praktische Umsetzbarkeit?
- Verfügbare Implementierungen?

Weitere Details in Vorlesung 14 . . .

Was haben konjunktive Anfragen mit Regeln zu tun?

Jede konjunktive Anfrage kann als Regel ausgedrückt werden:

$$\exists y.(\texttt{Buch}(x) \wedge \texttt{Autor}(x,y))$$

entspricht

$$\text{Buch}(x) \wedge \text{Autor}(x,y) \rightarrow \text{Ergebnis}(x)$$

$\rightsquigarrow$ Kopf enthält Bindungen für bestimmte Variablen

Zusätzliche Schwierigkeit von Regeln:

- Ergebnisse können in anderen Regeln/Ontologieaxiomen weiterverwendet werden (Rekursion!)
- Variablen nicht immer auf benannte Individuen beschränkt (außer für Ausgabe und in DL-safe Rules)

Zusammenfassung

Konjunktive Anfragen für OWL DL

- kein offizieller Standard, aber große Verbreitung
- Anfrage basierend auf logischer Beschreibung
- Diverse Erweiterungen (SPARQL-Features, $\neg$, $\vee$, Pfadausdrücke)
- Keine normierte Sysntax (manche Implementationen verwenden SPARQL-Syntax)
- Semantik durch Erweiterung der beschreibungslogischen Interpretation von OWL

Prädikatenlogische Regelerweiterungen für OWL DL

- Datalog als gut bekannter Formalismus
- Kombination mit OWL möglich: SWRL
- Semantik durch Erweiterung der beschreibungslogischen Interpretation von OWL

Vorlesung 12: OWL 2 (neue Features für OWL)
Vorlesung 14: Regeln für OWL (OWL 2 und SWRL praktisch nutzen)

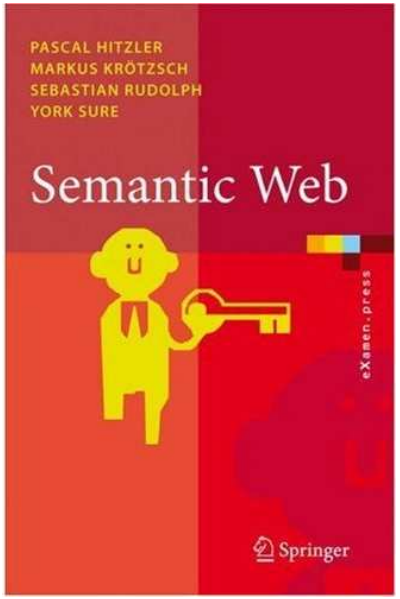
AIFB

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:
Kapitel 7 (Anfragen) & Vorlesung 11



AIFB