

Konjunktive Anfragen und Regelsprachen

Foliensatz: Markus Krötzsch, Sebastian Rudolph

Webbasierte Informationssysteme
15. November 2010

Die nichtkommerzielle Vervielfältigung, Verbreitung und Bearbeitung dieser Folien ist gestattet (→ Lizenzbestimmungen CC-BY-NC).

Semantic Web Technologies 1

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

Literaturhinweise siehe → Semantic Web – Grundlagen, Kapitel 7

OWL-Konzepte als Anfragesprache ungenügend

Beispiele:

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

Zudem fehlen OWL die nicht-logischen Features einer Anfragesprache (z.B. LIMIT).

Ausdrucksstarke Anfragen für OWL?

SPARQL als Anfragesprache für RDF

↪ keine direkte Unterstützung für OWL

Anfrageformalismus für OWL DL: **konjunktive Anfragen**

- keine offizielle Spezifikation, keine normative Syntax
 - ↪ 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 Implementierungen verfügbar

Konjunktive Anfragen sind recht einfach!

Beispiel:

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

„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

↪ Anfragen ohne Variablen können aus einer Ontologie *folgen*

↪ Variablen als Platzhalter für Bezeichner von Individuen

Funktion μ ist **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

↪ keine partielle Funktion – alle Variablen müssen belegt sein

↪ 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:

$\text{Buch}(a)$ (a ist ein Buch)

$\text{Buch} \sqsubseteq \exists \text{Autor.T}$ (jedes Buch hat einen Autor)

\rightsquigarrow **Anfrage** $\text{Buch}(x) \wedge \text{Autor}(x, y)$ 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

↪ **Unbestimmte Variablen** werden durch Existenzquantoren gebunden

Beispiel:

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

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

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

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	—

⇒ „SPARQL für OWL“:

leere Knoten der Ontologie: Individuen oder unbenannte Elemente?

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

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“)

Schlussfolgern mit OWL DL ist sehr komplex (NEXPTIME-vollständig)

↪ Wie schwierig sind dann konjunktive Anfragen?

Bisher noch nicht abschließend geklärt!

- Konjunktive Anfragen für *SHIQ* (und für OWL Lite):
2EXPTIME-vollständig!
- Konjunktive Anfragen für *SHOQ*: entscheidbar in 2EXPTIME
- Konjunktive Anfragen für *SHOIQ* (und für OWL DL):
Entscheidbarkeit nicht bekannt! (19.01.2010)
(Neuestes Ergebnis: entscheidbar für *ALCHOIQ* (Rudolph und Glimm, 2009))

↪ konjunktive Anfragen für OWL sind äußerst kompliziert

Implementationen von konjunktiven Anfragen für OWL verfügbar
(19.01.2010)

- **KAON2**: konjunktive Anfragen ohne unbestimmte Variablen, eingeschränkte Negation zulässig (→ Webseite)
- **Pellet**: konjunktive Anfragen mit unbestimmten Variablen und Negationen, nicht vollständig für OWL DL (→ Webseite)
- **Hermit**: konjunktive Anfragen ohne unbestimmte Variablen werden unterstützt (→ 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):
 - „ $F \rightarrow G$ “ (\equiv „ $\neg F \vee G$ “)
 - Logische Erweiterung der Wissensbasis \rightsquigarrow **statisch**
 - Open World
 - **Deklarativ** (beschreibend)
- ② Prozedurale Regeln (z.B. Production Rules):
 - „*If X then Y else Z*“
 - Ausführbare Maschinen-Anweisungen \rightsquigarrow **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“
 \rightsquigarrow 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

↪ Schwerpunkt dieser Vorlesung: prädikatenlogische Regeln
(die aber auch die Grundlage der Logikprogrammierung sind)

Prädikatenlogik als Regelsprache

- Regeln als Implikationsformeln der Prädikatenlogik:

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

↔ Semantisch äquivalent zu Disjunktion:

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

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

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

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)

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:

Standardsemantik der Prädikatenlogik!

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

Semantik von Datalog (Auffrischung)

Semantik definiert 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: $p^{\mathcal{I}} \in \Delta_{\mathcal{I}}^n$
 - $\mathcal{I}, \mathcal{Z} \models p(t_1, \dots, t_n)$ genau dann wenn $(t_1^{\mathcal{I},\mathcal{Z}}, \dots, t_n^{\mathcal{I},\mathcal{Z}}) \in p^{\mathcal{I}}$,
 - $\mathcal{I} \models B \rightarrow H$ 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 Regelmengde, wenn gilt: $\mathcal{I} \models B \rightarrow H$ für alle Regeln $B \rightarrow H$ dieser Menge

Datalog in der Praxis:

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

Erweiterungen von Datalog:

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

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

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

↔ \mathcal{I} kann gleichzeitig Modell sein für OWL-Ontologie und Menge von Datalog-Regeln

↔ Schlussfolgerung über OWL-Datalog-Kombination möglich

Beispiel

Kombinierte SWRL-Wissensbasis (Datalog + Beschreibungslogik):

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

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

- Komplexität?
- Praktische Umsetzbarkeit?
- Verfügbare Implementierungen?

Weitere Details in Vorlesung 13 ...

Was haben konjunktive Anfragen mit Regeln zu tun?

Jede konjunktive Anfrage kann als Regel ausgedrückt werden:

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

entspricht

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

↪ 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)

Konjunktive Anfragen für OWL DL

- kein offizieller Standard, aber große Verbreitung; eventuell Teil des nächsten SPARQL-Standards
- Anfrage basierend auf logischer Beschreibung
- Diverse Erweiterungen (SPARQL-Features, \neg , \vee , Pfadausdrücke)
- Keine normierte Syntax (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 13: Regeln für OWL (OWL 2 und SWRL praktisch nutzen)

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