

OWL 2

Pascal Hitzler Markus Krötzsch Sebastian Rudolph

Institut AIFB · Universität Karlsruhe

Semantic Web Technologies 1 (WS08/09)

21. Januar 2009

<http://semantic-web-grundlagen.de>

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



Rückblick: Erweiterungen von OWL

OWL für viele Aufgaben noch zu schwach (siehe → letzte Vorlesung).

- OWL als Anfragesprache ungenügend
↪ Konjunktive Anfragen, SPARQL für OWL
- OWL als Ontologiesprache ungenügend
↪ Prädikatenlogische Regelerweiterungen, SWRL
- OWL als Programmiersprache ungenügend
↪ Logikprogrammierung im Semantic Web
(→ Vorlesung SWebT2)

Sollte auch der OWL-Standard selbst erweitert werden?

↪ OWL 2



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
- 12 **OWL 2** (→ Webseite)
- 13 Bericht aus der Praxis
- 14 Regeln für OWL
- 15 Semantic Web – Anwendungen

Literaturhinweise siehe → Webseite dieser Vorlesung

Entwicklung von OWL 2

OWL 2 als „nächste Version“ von OWL

Erweiterungen aufgrund von Praxiserfahrung mit OWL 1:

- zusätzliche Ausdrucksstärke durch neue ontologische Axiome
- nicht-logische Erweiterungen (Syntax, Metadaten, ...)
- Überarbeitung der OWL-Varianten (Lite/DL/Full)

Zielstellungen:

- weitestgehende Kompatibilität zum existierenden OWL-Standard
- Erhaltung der Entscheidbarkeit von OWL DL
- Behebung von Problemen im OWL-1-Standard



OWL DL basiert auf Beschreibungslogik $SHOIN(D)$:

- Axiome:
 - Tbox: Subklassenbeziehungen $C \sqsubseteq D$
 - Rbox: Subrollenbeziehungen $R \sqsubseteq S(\mathcal{H})$, Inverse Rollen $R^-(\mathcal{I})$, Transitivität
 - Abox: Fakten zu Klassen $C(a)$, Rollen $R(a, b)$, und Gleichheit $a \approx b$ bzw. $a \not\approx b$
- Klassenkonstruktoren:
 - Konjunktion $C \sqcap D$, Disjunktion $C \sqcup D$, Negation $\neg C$ von Klassen
 - Rollenrestriktionen: universell $\forall R.C$ und existenziell $\exists R.C$
 - Zahlenrestriktionen (\mathcal{N}): $\leq n R$ und $\geq n R$ (n nicht-negative Zahl)
 - Nominale (\mathcal{O}): $\{a\}$
- Datentypen (D)



Erweiterung in OWL 2 zu SROIQ(D)

Zahlenrestriktionen

$SHOIN$ unterstützt nur einfache Zahlenrestriktionen (\mathcal{N}):

$\text{Person} \sqcap \geq 3 \text{ hatKind}$

„Klasse aller Personen mit 3 oder mehr Kindern.“

$\rightsquigarrow SROIQ$ erlaubt auch **qualifizierte Zahlenrestriktionen** (\mathcal{Q}):

$\text{Person} \sqcap \geq 3 \text{ hatKind} . (\text{Frau} \sqcap \text{Professor})$

„Klasse aller Personen mit 3 oder mehr Töchtern, die Professoren sind.“



$SHOIN$ unterstützt verschiedene Abox-Fakten:

- Klassenzugehörigkeit $C(a)$ (C komplexe Klasse),
- Sonderfall: negierte Klassenzugehörigkeit $\neg C(a)$ (C komplexe Klasse),
- Gleichheit $a \approx b$,
- Ungleichheit $a \not\approx b$
- Rollenbeziehungen $R(a, b)$
- *negierte Rollenbeziehungen?*

$\rightsquigarrow SROIQ$ erlaubt auch **negierte Rollen** in der Abox: $\neg R(a, b)$



Das Konzept Self

Modellierungsaufgabe: „Jeder Mensch kennt sich selbst.“

- $SHOIN$:

$\text{kennt}(\text{tom}, \text{tom}) \quad \text{kennt}(\text{tina}, \text{tina}) \quad \text{kennt}(\text{udo}, \text{udo}) \quad \dots$

\rightsquigarrow nicht allgemein anwendbar

- $SROIQ$: spezieller Ausdruck Self

$\text{Mensch} \sqsubseteq \exists \text{kennt} . \text{Self}$



Rollenaxiome und die universelle Rolle

SRIOIQ bietet zusätzliche Aussagen über Rollen:

- Tra(R): R ist **transitiv** (definiert wie in SHOIN)
Beispiel: Tra(liegtIn)
- Sym(R): R ist **symmetrisch** (definiert wie in SHOIN)
Beispiel: Sym(verwandtMit)
- Ref(R): R ist **reflexiv**, $(x, x) \in R^I$ für alle Domänenindividuen x
Beispiel: Ref(kennt)
- Irr(R): R ist **irreflexiv**, $(x, x) \notin R^I$ für alle Domänenindividuen x
Beispiel: Irr(hatKind)
- Dis(R, S): R und S sind **disjunkt**, $(x, y) \notin R^I \cap S^I$ für alle x, y
Beispiel: Dis(hatVater, hatSohn)
- **Universelle Rolle U**: $(x, y) \in U^I$ für alle x, y
Beispiel: $\top \sqsubseteq \leq 7000000000 U.Menschen$ (nicht empfohlen!)
~> U ist vor allem als Gegenstück zu \top sinnvoll, z.B. als Wurzel der Rollenhierarchie in grafischen Editoren



Ausdrucksstärke der Rolleninklusion

Wie kompliziert ist Rolleninklusion?

Mit Rboxen kann man formale Sprachen kodieren: **(Skizze!)**

Grammatik für Sprache der Wörter ab, aabb, aaabbb, ...:

$$L ::= ab \quad \text{wird zu Rbox} \quad R_a \circ R_b \sqsubseteq L$$

$$L ::= aLb \quad R_a \circ L \circ R_b \sqsubseteq L$$

- ~> $\exists L. \top \neq \perp$ ("∃L.⊤ notwendig nicht-leer") bedeutet*:
„Es gibt eine Kette aus R_a und R_b , die zur Sprache gehört.“
- ~> $\exists L_1. \exists L_2. \perp$ für zwei kodierte Sprachen L_1 und L_2 bedeutet:
„Es gibt ein Wort, das zu L_1 und zu L_2 gehört.“

*) bei entsprechender Tbox!

Leider gilt: Leerheit der Überschneidung kontextfreier Sprachen ist unentscheidbar.

~> **OWL mit Rolleninklusionen ist unentscheidbar**



Allgemeine Rolleninklusion

„Die Freunde meiner Freunde sind auch meine Freunde.“

~> Kann in SHOIN ausgedrückt werden: hatFreund ist transitiv.

„Die Feinde meiner Freunde sind auch meine Feinde.“

~> Kann nicht in SHOIN ausgedrückt werden!

Rolleninklusion

- Rbox-Ausdrücke der Form $R_1 \circ R_2 \circ \dots \circ R_n \sqsubseteq S$,
Beispiel: $\text{hatFreund} \circ \text{hatFeind} \sqsubseteq \text{hatFeind}$
- Semantik: wenn $(x_0, x_1) \in R_1^I, (x_1, x_2) \in R_2^I, \dots, (x_{n-1}, x_n) \in R_n^I$,
dann gilt auch $(x_0, x_n) \in S^I$
Beispiel: wenn $(x, y) \in \text{hatFreund}^I$ und $(y, z) \in \text{hatFeind}^I$,
dann gilt auch $(x, z) \in \text{hatFeind}^I$

Weitere Beispiele:

$\text{teilVon} \circ \text{gehört} \sqsubseteq \text{gehört}$ $\text{hatBruder} \circ \text{hatKind} \sqsubseteq \text{istOnkelVon}$



Reguläre Rboxen

Kann man Rolleninklusion zwecks Entscheidbarkeit einschränken?

- Rboxen sind wie Grammatiken für kontextfreie formale Sprachen
 - Überschneidungen von kontextfreien Sprachen problematisch
- ~> Einschränkung auf reguläre Sprachen!

Reguläre Rboxen

Rollennamen werden mit \prec geordnet (strenge totale Ordnung).
Jede Rbox-Inklusion muss eine der folgenden Formen haben:

- $R \circ R \sqsubseteq R$
- $R^- \sqsubseteq R$
- $S_1 \circ S_2 \circ \dots \circ S_n \sqsubseteq R$
- $R \circ S_1 \circ S_2 \circ \dots \circ S_n \sqsubseteq R$
- $S_1 \circ S_2 \circ \dots \circ S_n \circ R \sqsubseteq R$

Dabei gilt: $S_i \prec R$ für alle $i = 1, 2, \dots, n$.

Rbox ist regulär, wenn es so eine Ordnung \prec gibt.



Reguläre Rboxen – Beispiel

Beispiel:

$$R \circ S \sqsubseteq R \quad S \circ S \sqsubseteq S \quad R \circ S \circ R \sqsubseteq T$$

\rightsquigarrow ist regulär mit Ordnung $S \prec R \prec T$

Beispiel:

$$R \circ T \circ S \sqsubseteq T$$

\rightsquigarrow ist nicht regulär (unzulässige Inklusions-Form)

Beispiel:

$$R \circ S \sqsubseteq S \quad S \circ R \sqsubseteq R$$

\rightsquigarrow ist nicht regulär (keine gültige Ordnung möglich)



Beschränkung einfacher Rollen

- Einfache Rollen in *SHOIN* = Rollen ohne transitive Unterrollen
- In *SROIQ*: Beachtung der Rolleninklusionen nötig!

Einfache Rollen sind alle Rollen . . .

- die nicht auf der rechten Seite einer Rolleninklusion vorkommen,
- die Inverse von anderen einfachen Rollen sind,
- die nur auf der rechten Seite von Rolleninklusionen $R \sqsubseteq S$ vorkommen, bei denen links einzelne einfache Rollen stehen.

(Achtung: induktive Definition)

\rightsquigarrow nicht-einfach sind Rollen, die direkt oder indirekt von sich selbst abhängen (und deren Überrollen)

Warum ist das wichtig?

Ausdrücke $\leq n R.C, \geq n R.C, \text{Irr}(R), \text{Dis}(R, S), \exists R.\text{Self} \neg R(a, b)$
nur für einfache Rollen R und S erlaubt!

(Grund: Sicherstellung von Entscheidbarkeit)



Überblick *SROIQ*

Klassenausdrücke

Klassennamen	A, B
Konjunktion	$C \sqcap D$
Disjunktion	$C \sqcup D$
Negation	$\neg C$
Exist. Rollenrestr.	$\exists R.C$
Univ. Rollenrestr.	$\forall R.C$
Self	$\exists S.\text{Self}$
Größer-als	$\geq n S.C$
Kleiner-als	$\leq n S.C$
Nominale	$\{a\}$

Rollen

Rollennamen	R, S, T
einfache Rollen	S, T
Inverse Rollen	R^-
Universelle Rolle	U

Tbox (Klassenaxiome)

Inklusion	$C \sqsubseteq D$
Äquivalenz	$C \equiv D$

Rbox (Rollenaxiome)

Inklusion	$R_1 \sqsubseteq R_2$
Allgemeine Inkl.	$R_1^{(-)} \circ \dots \circ R_n^{(-)} \sqsubseteq R$
Transitivität	$\text{Tra}(R)$
Symmetrie	$\text{Sym}(R)$
Reflexivität	$\text{Ref}(R)$
Irreflexivität	$\text{Irr}(S)$
Disjunktheit	$\text{Dis}(S, T)$

Abox (Fakten)

Klassenzugehörigkeit	$C(a)$
Rollenbeziehung	$R(a, b)$
Neg. Rollenbeziehung	$\neg S(a, b)$
Gleichheit	$a \approx b$
Ungleichheit	$a \not\approx b$



Wie kompliziert ist *SROIQ*?

Rückblick: *SHOIN* (OWL DL) ist sehr komplex (NEXPTIME)

Wie komplex ist *SROIQ*?

Beobachtung: einige Ausdrucksmittel sind nicht wirklich nötig!

- $\text{Tra}(R)$ durch $R \circ R \sqsubseteq R$ ausdrückbar
- $\text{Sym}(R)$ durch $R^- \sqsubseteq R$ ausdrückbar
- $\text{Irr}(S)$ durch $\top \sqsubseteq \neg \exists S.\text{Self}$ ausdrückbar
- Universelle Rolle durch transitive, reflexive Überrolle aller Rollen ersetzbar (hier nicht vertieft)
- Abox durch Nominale darstellbar, z.B. $R(a, b)$ durch $\{a\} \sqsubseteq \exists R.\{b\}$

Qualifizierte Zahlenrestriktionen kaum problematisch (bekannt und implementiert, siehe Vorlesung zu OWL)

\rightsquigarrow Hauptproblem Rollenaxiome (Rbox)



Wie geht man mit Rboxen um?

- Rbox-Regeln ähneln formalen Grammatiken
- jede Rolle R definiert eine reguläre Sprache: die Sprache der Rollen-Ketten, aus denen R folgt
- reguläre Sprachen \equiv reguläre Ausdrücke \equiv endliche Automaten

↔ Ansatz: Tableauverfahren werden mit „Rbox-Automaten“ erweitert

Details siehe Literaturangaben zu SROIQ



OWL 2 DL: Weitere Aspekte

SROIQ ist „nur“ logische Grundlage von OWL 2 DL

Weitere nicht-logische Aspekte:

- Syntax (Erweiterung nötig)
- Datentypdeklaration und Datentypfunktionen, neue Datentypen?
- Metamodellierung: „Punning“
- Kommentarfunktionen und ontologische Metadaten
- Invers-funktionale konkrete Rollen (DatatypeProperties): „Simple Keys“
- Mechanismen zu Ontologieimport
- ...

Hier: Übersicht einiger Kernaspekte



Tableauverfahren von *SROIQ* verfügbar:

SROIQ ist entscheidbar.

- Tableau-Verfahren ungeeignet für enge Komplexitätsabschätzungen
- Komplexitätsresultat (2008): *SROIQ* ist N2EXPTIME-vollständig!
- Aber: Tableau-Algorithmus hat gute Anpassungseigenschaften: ungenutzte Merkmale belasten die Abarbeitung kaum („pay as you go“)



Metamodellierung

Metamodellierung

Spezifikation ontologischen Wissens *über* einzelne Elemente der Ontologie (einschließlich Klassen, Rollen, Axiome).

Beispiele:

- „Die Klasse *Person* wurde am 3.1.2009 von *MarkusK* angelegt.“
- „Für die Klasse *Stadt* wird die Property *Einwohnerzahl* empfohlen.“
- „Die Aussage ‚Dresden wurde 1206 gegründet‘ wurde maschinell ermittelt mit einer Sicherheit von 85%.“

(Vergleich auch Reifikation in RDF Schema)



Wortspiele in OWL: Punning

Metamodellierung in ausdrucksstarken Logiken ist gefährlich und teuer!

OWL 2 unterstützt zurzeit einfachste Form von Metamodellierung:

Punning

- Bezeichner für Klassen, Rollen, Individuen müssen nicht disjunkt sein (Ausnahme: ObjectProperty und DataProperty)
- keine *logische* Beziehung zwischen Klasse, Individuum und Rolle gleichen Namens
- Beziehung nur relevant für pragmatische Interpretation

Beispiel:

Person(Sebastian) klasseErstelltVon(Person, Markus)



Syntax von OWL 2

OWL 2 kann in verschiedenen Syntaktischen Formen ausgedrückt werden:

- Funktionale Syntax: ersetzt „Abstrakte Syntax“ von OWL 1
- RDF-Syntax: Erweiterung der bestehenden OWL/RDF-Abbildung
- XML-Syntax: Eigenständige XML-Serialisierung

↪ funktionale Syntax einfacher zu definieren

(keine RDF-Beschränkungen), kompakter

↪ RDF-Syntax für Abwärtskompatibilität wichtig



Kommentare und Metadaten

Punning unterstützt einfache Metadaten mit (schwacher) semantischer Bedeutung

Wie kann man rein „syntaktische“ Kommentare zu einer Ontologie machen?

- Kommentare in XML-Dateien: `<!-- Kommentar -->`
↪ kein Bezug auf OWL-Axiome dieser Datei
- nicht-logische Annotationen in OWL 1:
`owl:AnnotationProperty`
↪ fest verknüpft mit (semantischem) ontologischem Element, kein syntaktischer Bezug

OWL 2 verändert die Bedeutung von Annotationen: keine semantische Interaktion, aber struktureller Teil von OWL-Ontologien.

Außerdem: Annotationen ganzer Axiome möglich, nicht nur von Individuen



Quo vadis, OWL Lite?

OWL Lite als Fehlschlag:

- beinahe so komplex wie OWL DL
- komplizierte Syntax gibt keinen direkten Zugang zu wahrer Ausdrucksstärke
- Verwendung in Ontologien heute praktisch nur „zufällig“, nicht bewusst

Ursprüngliches Ziel:

einfach und effizient implementierbarer Teil von OWL

↪ neuer Ansatz in OWL 2: mehrere einfache Sprachprofile



OWL-Profil basierend auf Beschreibungslogik \mathcal{EL}^{++} :

Beschreibungslogik \mathcal{EL}^{++}

- Konzepte nur mit Konjunktion $C \sqcap D$, Existenz $\exists R.C$, \top und \perp
- Nominale, eingeschränkte Property-Ranges
- allgemeine Rolleninklusionen (Rbox), Transitivität

Vorteile:

- polynomielle Komplexität
- schnelle Implementierungen verfügbar
- unterstützt praktisch relevante Ontologien (z.B. SNOMED-CT)



OWL-Profil basierend auf Horn-Regel-Fragment von OWL 2:

Horn-Regel-Fragment von OWL 2

- Oberklassen (rechte Seite von \sqsubseteq): \top , $\exists R.\{a\}$, $\forall R.C$, $\leq 1R.C$
- Unterklassen (linke Seite von \sqsubseteq): \top , \perp , $\exists R.C$, $\exists R.\{a\}$
- Keine negierten Fakten, keine Reflexivität, sonst alle Rbox-Features

Vorteile:

- polynomielle Komplexität
- relativ einfache Implementierung (OWL-Axiome als Regeln)
- verwandt mit Regelsprachen



OWL-Profil basierend auf Beschreibungslogik DL Lite:

Beschreibungslogik DL Lite

- Oberklassen (rechte Seite von \sqsubseteq): \top , \neg , $\exists R.C$
- Unterklassen (linke Seite von \sqsubseteq): $\exists R.\top$
- Inverse Rollen, einfache Rollenhierarchien
- Abox wie in *SROIQ*

Vorteile:

- sub-polynomielle Komplexität (verwandt mit relationalen Datenbanken)
- schnelle Implementierungen verfügbar
- geeignet für besonders große Datenmengen



- Erweiterung von OWL Full um neue OWL-2-Konstrukte
- Semantik (größtenteils) als Erweiterung der OWL-Full-Semantik
- gedacht eher als konzeptionelle Modellierungssprache, zurzeit wenig Softwareunterstützung für automatische Ableitungen
- logische Konsistenz der Spezifikation weiter offen (wie bei OWL Full)

viele OWL-Full-Ontologien nunmehr auch als OWL 2 DL interpretierbar (siehe z.B. Punning)



Aktueller Status:

- OWL-Arbeitsgruppe besteht seit September 2007 (erstes Treffen Dezember 2007)
- inzwischen letzte Entwürfe für alle Kerndokumente veröffentlicht
- Softwareunterstützung für Inferenz und Ontologieerstellung zum Teil bereits verfügbar

Offizieller Zeitplan:

- Dezember 2008: fertig ausgearbeitete Standardisierungsempfehlung (*Candidate Recommendation*)
- März 2009: Verabschiedung des neuen Standards



OWL 2 als erste Weiterentwicklung des OWL-Standards

- logische Erweiterung: Beschreibungslogik *SROIQ* als Grundlage
- neue Ausdrucksmittel vor allem Rollenaxiome, qualifizierte Zahlenrestriktionen
- nicht-logische Erweiterungen: Punning, Annotationen, Datentypen, u.a.
- OWL-Profile als Ersatz von OWL Lite
- OWL 2 Full im Sinne von OWL Full definiert
- Verabschiedung bis März 2009 geplant



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

