Regeln für OWL

Pascal Hitzler Markus Krötzsch Sebastian Rudolph

Institut AIFB · Universität Karlsruhe

Semantic Web Technologies 1 (WS07/08)

4. Februar 2009

http://semantic-web-grundlagen.de

Outline

1. Einleitung und Wiederholung
2. Logisches Schließen in SWRL
3. Description Logic Rules
4. DL-safe Rules
5. Zusammenfassung

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
13. Bericht aus der Praxis
14. Regeln für OWL (→ Webseite)
15. Semantic Web – Anwendungen

Literaturhinweise siehe → Webseite dieser Vorlesung

Was bisher geschah . . .

Problem

Ontologiesprache OWL DL für verschiedene Anwendungen zu schwach

- Konjunktive Anfragen: Abfrage von Instanzen in OWL-DL-Wissensbasen
- SWRL: Erweiterung von OWL DL mit Datalog-Regeln
  $\rightsquigarrow$ Konjunktive Anfragen in SWRL darstellbar durch einzelne Regeln
- OWL 2: Erweiterung von OWL, logische und nicht-logische Änderungen
  $\rightsquigarrow$ OWL 2 DL ebenfalls mit konjunktiven Anfragen und SWRL erweiterbar

SWRL-Beispiel (Wiederholung)

Kombinierte SWRL-Wissensbasis* (Datalog + Beschreibungslogik):

(1) $\mathtt{Vegetarier}(x) \wedge \mathtt{Fischprodukt}(y) \to \mathtt{magNicht}(x, y)$

(2) $\mathtt{hatBestellt}(x, y) \wedge \mathtt{magNicht}(x, y) \to \mathtt{Unglücklich}(x)$

(3) $\mathtt{hatBestellt}(x, y) \to \mathtt{Gericht}(y)$

(4) $\mathtt{magNicht}(x, z) \wedge \mathtt{Gericht}(y) \wedge \mathtt{enthält}(y, z) \to \mathtt{magNicht}(x, y)$

(5) $\to \mathtt{Vegetarier}(\mathtt{markus})$

(6) $\mathtt{Glücklich}(x) \wedge \mathtt{Unglücklich}(x) \to$

(7) $\exists \mathtt{hatBestellt}.\mathtt{ThaiCurry}(\mathtt{markus})$

(8) $\mathtt{ThaiCurry} \sqsubseteq \exists \mathtt{enthält}.\mathtt{Fischprodukt}$

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

* Wir meinen mit „SWRL“ auch in dieser Vorlesung die freie Kombination von Datalog & Beschreibungslogik. Das entspricht nicht ganz ursprünglichen Definition von SWRL für OWL (1) DL.

Outline

1. Einleitung und Wiederholung
2. Logisches Schließen in SWRL
3. Description Logic Rules
4. DL-safe Rules
5. Zusammenfassung

Wie schwer ist SWRL?

1. Logisches Schließen in OWL DL ist NEXPTIME-vollständig.
2. Logisches Schließen in OWL 2 DL ist N2EXPTIME-vollständig.
3. Logisches Schließen in Datalog ist EXPTIME-vollständig.

$\leadsto$ Wie schwer ist logisches Schließen in SWRL?

Wie schwer ist SWRL?

1. Logisches Schließen in OWL DL ist NExpTime-vollständig.
2. Logisches Schließen in OWL 2 DL ist N2ExpTime-vollständig.
3. Logisches Schließen in Datalog ist ExpTime-vollständig.

$\rightsquigarrow$ Wie schwer ist logisches Schließen in SWRL?

Logisches Schließen in SWRL ist unentscheidbar
(für OWL und damit auch für OWL 2).

Unentscheidbarkeit von SWRL

SWRL ist unentscheidbar

Es gibt keinen Algortihmus, mit dem man *alle* logischen Schlüsse aus *allen* SWRL-Wissensbasen ziehen kann, selbst wenn man beliebig (endlich) viel Rechenzeit und Speicher zur Verfügung hat.

Praktisch möglich dagegen sind:

1. Algorithmen, die alle Schlüsse aus einem Teil der SWRL-Wissensbasen ziehen
2. Algorithmen, die aus allen SWRL-Wissensbasen einen Teil der Schlüsse ziehen

$\rightsquigarrow$ Beides ist trivial möglich, wenn der entsprechende „Teil“ nur sehr klein ist.

Entscheidbare Fragmente von SWRL

Für welche Arten von SWRL-Wissensbasen kann man vollständige Inferenz-Algorithmen finden?

- Für die Menge aller SWRL-Wissensbasen, die nur aus Ontologien in OWL (2) bestehen.
- Für die Menge aller SWRL-Wissensbasen, die nur aus Datalog-Regeln bestehen.
- Für jede feste endliche Menge an SWRL-Wissensbasen.

$\rightsquigarrow$ Gibt es noch interessantere entscheidbare Fragmente?

Entscheidbare Fragmente von SWRL

Für welche Arten von SWRL-Wissensbasen kann man vollständige Inferenz-Algorithmen finden?

- Für die Menge aller SWRL-Wissensbasen, die nur aus Ontologien in OWL (2) bestehen.
- Für die Menge aller SWRL-Wissensbasen, die nur aus Datalog-Regeln bestehen.
- Für jede feste endliche Menge an SWRL-Wissensbasen.

$\rightsquigarrow$ Gibt es noch interessantere entscheidbare Fragmente?

- Description Logic Rules
- DL-safe Rules

Outline

Description Logic Rules

Beobachtung

Manche SWRL-Regeln lassen sich bereits in OWL 2 (also der Beschreibungslogik $\mathcal{SROIQ}$) ausdrücken.

- Identifizierung dieser Description Logic Rules liefert ein entscheidbares Fragment von SWRL
- Ziel: „Versteckte“ Ausdrucksstärke von OWL 2 nutzen
- Implementierung direkt durch OWL-2-Tools

Rückblick: $\mathcal{SROIQ}$

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	$\exists S.\mathrm{Self}$
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.	$R_1^{(-)} \circ \ldots \circ R_n^{(-)} \sqsubseteq R$
Transitivität	$\mathrm{Tra}(R)$
Symmetrie	$\mathrm{Sym}(R)$
Reflexivität	$\mathrm{Ref}(R)$
Irreflexivität	$\mathrm{Irr}(S)$
Disjunktheit	$\mathrm{Dis}(S, T)$

Abox (Fakten)

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

Einfache Regeln mit $\mathcal{SROIQ}$

Alle $\mathcal{SROIQ}$-Axiome können als SWRL-Regeln geschrieben werden:

- $C \sqsubseteq D$ entspricht $C(x) \to D(x)$
- $R \sqsubseteq S$ entspricht $R(x, y) \to S(x, y)$

Einfache Regeln mit $\mathcal{SROIQ}$

Alle $\mathcal{SROIQ}$-Axiome können als SWRL-Regeln geschrieben werden:

- $C \sqsubseteq D$ entspricht $C(x) \to D(x)$
- $R \sqsubseteq S$ entspricht $R(x,y) \to S(x,y)$

Einige Klassen können innerhalb von Regeln „zerlegt" werden:

- $\mathtt{Glücklich} \sqcap \mathtt{Unglücklich} \sqsubseteq \bot$ entspricht $\mathtt{Glücklich}(x) \land \mathtt{Unglücklich}(x) \to$
- $\exists\mathtt{wohnort}.\exists\mathtt{liegtIn}.\mathtt{EULand} \sqsubseteq \mathtt{EUBürger}$ entspricht $\mathtt{wohnort}(x,y) \land \mathtt{liegtIn}(y,z) \land \mathtt{EULand}(z) \to \mathtt{EUBürger}(x)$

Einfache Regeln mit $\mathcal{SROIQ}$

Alle $\mathcal{SROIQ}$-Axiome können als SWRL-Regeln geschrieben werden:

- $C \sqsubseteq D$ entspricht $C(x) \to D(x)$
- $R \sqsubseteq S$ entspricht $R(x,y) \to S(x,y)$

Einige Klassen können innerhalb von Regeln „zerlegt“ werden:

- $\mathtt{Glücklich} \sqcap \mathtt{Unglücklich} \sqsubseteq \bot$ entspricht $\mathtt{Glücklich}(x) \land \mathtt{Unglücklich}(x) \to$
- $\exists\mathtt{wohnort}.\exists\mathtt{liegtIn}.\mathtt{EULand} \sqsubseteq \mathtt{EUBürger}$ entspricht $\mathtt{wohnort}(x,y) \land \mathtt{liegtIn}(y,z) \land \mathtt{EULand}(z) \to \mathtt{EUBürger}(x)$

$\mathcal{SROIQ}$-Rollenaxiome liefern weitere Regeln:

- $\mathtt{hatMutter} \circ \mathtt{hatBruder} \sqsubseteq \mathtt{hatOheim}$ entspricht $\mathtt{hatMutter}(x,y) \land \mathtt{hatBruder}(y,z) \to \mathtt{hatOheim}(x,z)$

Noch mehr Regeln (I)

Was ist mit
$\texttt{magNicht}(x, z) \wedge \texttt{Gericht}(y) \wedge \texttt{enthält}(y, z) \rightarrow \texttt{magNicht}(x, y)$?

Noch mehr Regeln (I)

Was ist mit
$\texttt{magNicht}(x,z) \land \texttt{Gericht}(y) \land \texttt{enthält}(y,z) \rightarrow \texttt{magNicht}(x,y)$?

- Regelkopf mit zwei Variablen $\rightsquigarrow$ nicht durch Subklassen-Axiom darstellbar
- Regelrumpf enthält Klassenausdrücke $\rightsquigarrow$ nicht durch Subproperty-Axiom darstellbar

Noch mehr Regeln (I)

Was ist mit
$\texttt{magNicht}(x,z) \wedge \texttt{Gericht}(y) \wedge \texttt{enthält}(y,z) \to \texttt{magNicht}(x,y)$?

- Regelkopf mit zwei Variablen $\rightsquigarrow$ nicht durch Subklassen-Axiom darstellbar
- Regelrumpf enthält Klassenausdrücke $\rightsquigarrow$ nicht durch Subproperty-Axiom darstellbar

Trotzdem ist diese Regel in OWL 2 darstellbar!

Noch mehr Regeln (II)

Einfacheres Beispiel: $\texttt{Mann}(x) \wedge \texttt{hatKind}(x, y) \rightarrow \texttt{vaterVon}(x, y)$

Idee

Ersetze $\texttt{Mann}(x)$ durch ein Rollen-Atom, so dass die Regel als allgemeine Rolleninklusion mit $\circ$ darstellbar wird.

Noch mehr Regeln (II)

Einfacheres Beispiel: $\texttt{Mann}(x) \wedge \texttt{hatKind}(x, y) \to \texttt{vaterVon}(x, y)$

Idee

Ersetze $\texttt{Mann}(x)$ durch ein Rollen-Atom, so dass die Regel als allgemeine Rolleninklusion mit $\circ$ darstellbar wird.

Trick: mit $\exists R.\mathrm{Self}$ kann man Klassen in Rollen umwandeln:

- Hilfsrolle $R_{\texttt{Mann}}$
- Hilfsaxiom $\texttt{Mann} \equiv \exists R_{\texttt{Mann}}.\mathrm{Self}$
- Intuition: „Männer sind genau die Dinge, die ein $R_{\texttt{Mann}}$-Beziehung zu sich selbst haben.“

Mit diesem Hilfsaxiom kann die Regel geschrieben werden als:
$R_{\texttt{Mann}} \circ \texttt{hatKind} \sqsubseteq \texttt{vaterVon}$

Noch mehr Regeln (III)

Beispiel:

$$\texttt{magNicht}(x,z) \land \texttt{Gericht}(y) \land \texttt{enthält}(y,z) \rightarrow \texttt{magNicht}(x,y)$$

Noch mehr Regeln (III)

Beispiel:

$$\texttt{magNicht}(x,z) \wedge \texttt{Gericht}(y) \wedge \texttt{enthält}(y,z) \rightarrow \texttt{magNicht}(x,y)$$

wird zu

$$\texttt{Gericht} \equiv \exists R_{\texttt{Gericht}}.\mathsf{Self}$$

$$\texttt{magNicht} \circ \texttt{enthält}^{-} \circ R_{\texttt{Gericht}} \sqsubseteq \texttt{magNicht}$$

Noch mehr Regeln (IV)

Nicht so einfach:

$\texttt{Vegetarier}(x) \wedge \texttt{Fischprodukt}(y) \rightarrow \texttt{magNicht}(x, y)$

Noch mehr Regeln (IV)

Nicht so einfach:
$\mathtt{Vegetarier}(x) \wedge \mathtt{Fischprodukt}(y) \rightarrow \mathtt{magNicht}(x, y)$

Idee

Verbinde unzusammenhängende Teile im Regelrumpf durch die universelle Rolle U.

- Hilfsrollen $R_{\mathtt{Vegetarier}}$ und $R_{\mathtt{Fischprodukt}}$
- Hilfsaxiome $\mathtt{Vegetarier} \equiv \exists R_{\mathtt{Vegetarier}}.\mathrm{Self}$ und $\mathtt{Fischprodukt} \equiv \exists R_{\mathtt{Fischprodukt}}.\mathrm{Self}$

Mit diesen Hilfsaxiomen kann die Regel geschrieben werden als:
$R_{\mathtt{Vegetarier}} \circ U \circ R_{\mathtt{Fischprodukt}} \sqsubseteq \mathtt{magNicht}$

Die Grenzen von Description Logic Rules

Nicht alle SWRL-Regeln können so dargestellt werden!

Die Grenzen von Description Logic Rules

Nicht alle SWRL-Regeln können so dargestellt werden!

Beispiel:
$\mathtt{hatBestellt}(x, y) \wedge \mathtt{magNicht}(x, y) \rightarrow \mathtt{Unglücklich}(x)$
ist nicht in $\mathcal{SROIQ}$ darstellbar.

Mögliche Umwandlungen im Regelrumpf im Überblick

- Rollen umkehren, z.B. $\mathtt{enthält}(y, z) \mapsto \mathtt{enthält}^{-}(z, y)$
- Seitenarme „aufrollen“, z.B.
  $\mathtt{liegtIn}(y, z) \wedge \mathtt{EULand}(z) \mapsto \exists\mathtt{liegtIn}.\mathtt{EULand}(y)$
- Konzepte durch Rollen ersetzen, z.B. $\mathtt{Mann}(x) \mapsto R_{\mathtt{Mann}}(x, x)$
- Ketten in Rolleninklusionen umwandeln ($\wedge$ durch $\circ$ ersetzen)

Description Logic Rules: Definition

Vorbereitung: Regel normalisieren

- Für jedes *Vorkommen* (!) einer Konstante a der Regel: Füge im Rumpf $\{a\}(x)$ mit einer neuen Variable x ein und ersetze das Vorkommen von a durch x.
- Ersetze jedes Atom $R(x,x)$ durch $\exists R.\mathrm{Self}(x)$.

Description Logic Rules: Definition

Vorbereitung: Regel normalisieren

- Für jedes *Vorkommen* (!) einer Konstante a der Regel:
  Füge im Rumpf $\{a\}(x)$ mit einer neuen Variable x ein und ersetze das Vorkommen von a durch x.
- Ersetze jedes Atom $R(x,x)$ durch $\exists R.\mathrm{Self}(x)$.

Abhängigkeitsgraph einer Regel: *Ungerichteter* Graph mit

- Knoten = Variablen der Regel
- Kanten = Rollenatome der Regel (ohne Richtung!)

Description Logic Rules: Definition

Vorbereitung: Regel normalisieren

- Für jedes *Vorkommen* (!) einer Konstante a der Regel: Füge im Rumpf $\{a\}(x)$ mit einer neuen Variable x ein und ersetze das Vorkommen von a durch x.
- Ersetze jedes Atom $R(x, x)$ durch $\exists R.\mathrm{Self}(x)$.

Abhängigkeitsgraph einer Regel: *Ungerichteter* Graph mit

- Knoten = Variablen der Regel
- Kanten = Rollenatome der Regel (ohne Richtung!)

Eine SWRL-Regel ist eine Description Logic Rule wenn gilt:

1. alle Regelatome verwenden $\mathcal{SROIQ}$-Konzepte und -Rollen,
2. der Abhängigkeitsgraph der normalisierten Regel hat keine Zyklen.

Beispiel

DL Rules in der früheren SWRL-Wissensbasis:

(1) $\mathtt{Vegetarier}(x) \wedge \mathtt{Fischprodukt}(y) \rightarrow \mathtt{magNicht}(x, y)$

(3) $\mathtt{hatBestellt}(x, y) \rightarrow \mathtt{Gericht}(y)$

(4) $\mathtt{magNicht}(x, z) \wedge \mathtt{Gericht}(y) \wedge \mathtt{enthält}(y, z) \rightarrow \mathtt{magNicht}(x, y)$

(5) $\rightarrow \mathtt{Vegetarier}(\mathtt{markus})$

(6) $\mathtt{Glücklich}(x) \wedge \mathtt{Unglücklich}(x) \rightarrow$

Regel (2) $\mathtt{hatBestellt}(x, y) \wedge \mathtt{magNicht}(x, y) \rightarrow \mathtt{Unglücklich}(x)$ ist keine DL Rule

Anmerkung: Description Logic Rules müssen nach Umwandlung in $\mathcal{SROIQ}$ natürlich auch die Bedingungen and einfache Rollen und reguläre RBoxen erfüllen!

Umwandlung von DL Rules nach $\mathcal{SROIQ}$ (I)

Eingabe: Eine Description Logic Rule

Umwandlung von DL Rules nach $\mathcal{SROIQ}$ (I)

Eingabe: Eine Description Logic Rule

1. Normalisiere die Regel.

Umwandlung von DL Rules nach $\mathcal{SROIQ}$ (I)

Eingabe: Eine Description Logic Rule

1. Normalisiere die Regel.
2. Für jedes Paar von Variablen x und y:
   Sind x und y im Abhängigkeitsgraph nicht verbunden, dann füge im Rumpf $U(x, y)$ ein.

Umwandlung von DL Rules nach $\mathcal{SROIQ}$ (I)

Eingabe: Eine Description Logic Rule

1. Normalisiere die Regel.
2. Für jedes Paar von Variablen x und y:
   Sind x und y im Abhängigkeitsgraph nicht verbunden, dann füge im Rumpf $U(x, y)$ ein.
3. Der Regelkopf hat nun die Form $D(z)$ oder $S(z, z')$.
   Für jedes Atom $R(x, y)$ im Rumpf:
   Falls im Abhängigkeitsgraph der Pfad von z nach y kürzer ist als der von z nach x, so ersetze $R(x, y)$ mit $R^{-}(y, x)$.

Umwandlung von DL Rules nach $\mathcal{SROIQ}$ (I)

Eingabe: Eine Description Logic Rule

1. Normalisiere die Regel.
2. Für jedes Paar von Variablen x und y:
   Sind x und y im Abhängigkeitsgraph nicht verbunden, dann füge im Rumpf $U(x, y)$ ein.
3. Der Regelkopf hat nun die Form $D(z)$ oder $S(z, z')$.
   Für jedes Atom $R(x, y)$ im Rumpf:
   Falls im Abhängigkeitsgraph der Pfad von z nach y kürzer ist als der von z nach x, so ersetze $R(x, y)$ mit $R^-(y, x)$.
4. Falls im Rumpf ein Atom $R(x, y)$ vorkommt, so dass y in keinem anderen zweistelligen Atom der Regel auftritt:
   - Wenn der Rumpf n einstellige Atome $C_1(y), \ldots, C_n(y)$ enthält, dann definiere $E := C_1 \sqcap \ldots \sqcap C_n$ und entferne $C_1(y), \ldots, C_n(y)$ aus dem Rumpf. Andernfalls definiere $E := \top$.
   - Ersetze $R(x, y)$ durch $\exists R.E(x)$.

   Wiederhole Schritt 4 solange es solche $R(x, y)$ gibt.

Umwandlung von DL Rules nach $\mathcal{SROIQ}$ (II)

Die Regel kann jetzt in $\mathcal{SROIQ}$ ausgedrückt werden:

- Falls der Regelkopf einstellig ist, dann hat die Regel die Form $C_1(x) \wedge \ldots \wedge C_n(x) \to D(x)$.
  Ersetze sie durch $C_1 \sqcap \ldots \sqcap C_n \sqsubseteq D$.
- Falls der Regelkopf zweistellig ist, dann
  - Für jedes einstellige Atom $C(z)$ im Rumpf:
    Erzeuge ein neues Axiom $C \equiv \exists R_C.\mathrm{Self}$ (die Rolle R_C ist neu) und ersetze $C(z)$ durch $R_C(z, z)$.
  - Die Regel hat nun die Form
    $R_1(x, x_2) \wedge \ldots \wedge R_n(x_n, y) \to S(x, y)$.
    Ersetze sie durch $R_1 \circ \ldots \circ R_n \sqsubseteq S$.

Diese Umformung von Regeln einer SWRL-Wissensbasis verändert ihre Erfüllbarkeit nicht.

(Natürlich dürfen Hilfssymbole wie R_C noch nirgends vorkommen.)

Outline

DL-safe Rules

Beobachtung

Datalog ist entscheidbar, weil Regeln nur auf endlich viele Arten angewendet werden müssen: Variablen stehen nur für Konstanten.

- Variablen in SWRL könnten für unendlich viele „geschlussfolgerte“ Individuen stehen.
- Ziel: Regeln durch Datalog-Prädikate „absichern“ um Variablenbelegung einzuschränken
- DL-safe Rules als weiteres entscheidbares Fragment von SWRL

DL-safe Rules: Definition

Diesmal enthalten Regeln auch Nicht-DL-Atome:

- Ein Datalog-Atom ist ein Atom, dessen Prädikatsymbol in keinem $\mathcal{SROIQ}$-Axiom der Wissensbasis vorkommt.

DL-safe Rules: Definition

Diesmal enthalten Regeln auch Nicht-DL-Atome:

- Ein Datalog-Atom ist ein Atom, dessen Prädikatsymbol in keinem $\mathcal{SROIQ}$-Axiom der Wissensbasis vorkommt.

Eine SWRL-Regel ist DL-safe wenn gilt:

- Jede Variable der Regel kommt auch in einem Datalog-Atom im Rumpf vor.

$\rightsquigarrow$ Variablenbelegungen für Datalog-Atome können letztlich nur Konstanten entsprechen!

Erzwingen von DL-safety

Beispiel:

$\texttt{hatBestellt}(x, y) \wedge \texttt{magNicht}(x, y) \rightarrow \texttt{Unglücklich}(x)$
$\rightsquigarrow$ nicht DL-safe, wenn „hatBestellt" und „magNicht" in DL-Axiomen vorkommen.

Erzwingen von DL-safety

Beispiel:

$\texttt{hatBestellt}(x, y) \wedge \texttt{magNicht}(x, y) \to \texttt{Unglücklich}(x)$
$\leadsto$ nicht DL-safe, wenn „hatBestellt“ und „magNicht“ in DL-Axiomen vorkommen.

Erzwingen von DL-safeness durch Einschränken der Regeln auf bekannte Individuen:

$\texttt{hatBestellt}(x, y) \wedge \texttt{magNicht}(x, y) \wedge O(x) \wedge O(y) \to \texttt{Unglücklich}(x)$

wobei ein Fakt $O(a)$ für alle OWL-Individuen a angelegt wird.

$\leadsto$ Regel nur noch auf bekannte OWL-Individuen anwendbar

DL-safe Rules in der Praxis

- OWL 2 DL mit DL-safe Rules ist entscheidbar
- Naive Implementierung: jede Regel ist ausdrückbar durch viele Regeln, indem man alle Variablen durch Konstanten ersetzt (auf jede denkbare Weise!)
- Keine größere *theoretische* Komplexität der Berechnung

Implementierungen (01.02.2009):

- KAON2: sehr effiziente Umsetzung von DL-safe Rules, Unterstützung für disjunktive Regeln ($\rightarrow$ Webseite)
- Pellet: „preliminary implementation“ für DL-safe rules ($\rightarrow$ Webseite)

$\rightsquigarrow$ Umsetzung mit klassischen Tableau-Methoden kompliziert
$\rightsquigarrow$ „Vorberechnung“ von OWL-Ergebnissen für die Verwendung in eigenständigen Datalog-Systemen ist nicht ausreichend (unvollständig aber ev. effizienter, optional in Pellet verfügbar)

Ein kombiniertes Beispiel (I)

$\mathcal{SROIQ}$ + Description Logic Rules + DL-safe Rules weiterhin entscheidbar:

(1) $\texttt{Vegetarier}(x) \wedge \texttt{Fischprodukt}(y) \rightarrow \texttt{magNicht}(x,y)$

(2) $\texttt{hatBestellt}(x,y) \wedge \texttt{magNicht}(x,y) \wedge O(x) \wedge O(y) \rightarrow \texttt{Unglücklich}(x)$

(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) $\texttt{ThaiCurry} \sqsubseteq \exists\texttt{enthält}.\texttt{Fischprodukt}$

(9) $O(\texttt{markus})$

Ein kombiniertes Beispiel (I)

$\mathcal{SROIQ}$ + Description Logic Rules + DL-safe Rules weiterhin entscheidbar:

(1) $\texttt{Vegetarier}(x) \land \texttt{Fischprodukt}(y) \to \texttt{magNicht}(x, y)$

(2) $\texttt{hatBestellt}(x, y) \land \texttt{magNicht}(x, y) \land O(x) \land O(y) \to \texttt{Unglücklich}(x)$

(3) $\texttt{hatBestellt}(x, y) \to \texttt{Gericht}(y)$

(4) $\texttt{magNicht}(x, z) \land \texttt{Gericht}(y) \land \texttt{enthält}(y, z) \to \texttt{magNicht}(x, y)$

(5) $\to \texttt{Vegetarier}(\texttt{markus})$

(6) $\texttt{Glücklich}(x) \land \texttt{Unglücklich}(x) \to$

(7) $\exists\texttt{hatBestellt}.\texttt{ThaiCurry}(\texttt{markus})$

(8) $\texttt{ThaiCurry} \sqsubseteq \exists\texttt{enthält}.\texttt{Fischprodukt}$

(9) $O(\texttt{markus})$

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

Ein kombiniertes Beispiel (II)

$\mathcal{SROIQ}$ + Description Logic Rules + DL-safe Rules weiterhin entscheidbar:

(1) $\texttt{Vegetarier}(x) \wedge \texttt{Fischprodukt}(y) \to \texttt{magNicht}(x, y)$

(2) $\texttt{hatBestellt}(x, y) \wedge \texttt{magNicht}(x, y) \wedge O(x) \wedge O(y) \to \texttt{Unglücklich}(x)$

(3) $\texttt{hatBestellt}(x, y) \to \texttt{Gericht}(y)$

(4) $\texttt{magNicht}(x, z) \wedge \texttt{Gericht}(y) \wedge \texttt{enthält}(y, z) \to \texttt{magNicht}(x, y)$

(5) $\to \texttt{Vegetarier}(\texttt{markus})$

(6) $\texttt{Glücklich}(x) \wedge \texttt{Unglücklich}(x) \to$

(7) $\texttt{hatBestellt}(\texttt{markus}, \texttt{redThaiCurry})$
$\texttt{ThaiCurry}(\texttt{redThaiCurry})$

(8) $\texttt{ThaiCurry} \sqsubseteq \exists\texttt{enthält}.\texttt{Fischprodukt}$

(9) $O(\texttt{markus}) \qquad O(\texttt{redThaiCurry})$

Ein kombiniertes Beispiel (II)

$\mathcal{SROIQ}$ + Description Logic Rules + DL-safe Rules weiterhin entscheidbar:

(1) $\mathtt{Vegetarier}(x) \wedge \mathtt{Fischprodukt}(y) \to \mathtt{magNicht}(x,y)$

(2) $\mathtt{hatBestellt}(x,y) \wedge \mathtt{magNicht}(x,y) \wedge O(x) \wedge O(y) \to \mathtt{Unglücklich}(x)$

(3) $\mathtt{hatBestellt}(x,y) \to \mathtt{Gericht}(y)$

(4) $\mathtt{magNicht}(x,z) \wedge \mathtt{Gericht}(y) \wedge \mathtt{enthält}(y,z) \to \mathtt{magNicht}(x,y)$

(5) $\to \mathtt{Vegetarier}(\mathtt{markus})$

(6) $\mathtt{Glücklich}(x) \wedge \mathtt{Unglücklich}(x) \to$

(7) $\mathtt{hatBestellt}(\mathtt{markus}, \mathtt{redThaiCurry})$
$\mathtt{ThaiCurry}(\mathtt{redThaiCurry})$

(8) $\mathtt{ThaiCurry} \sqsubseteq \exists \mathtt{enthält}.\mathtt{Fischprodukt}$

(9) $O(\mathtt{markus})$ $O(\mathtt{redThaiCurry})$

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

Outline

Zusammenfassung

SWRL ist untentscheidbar.

Description Logic Rules

- in OWL 2 ausdrückbares SWRL-Fragment
- indirekte Unterstützung durch alle OWL-2-Tools
- Definition und Algorithmus basieren auf Abhängigkeitsgraph

DL-safe Rules

- SWRL-Fragment, in dem Variablen nur Konstanten als Werte annehmen
- Unterstützung durch einige OWL-Reasoner
- frei mit Description Logic Rules kombinierbar

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:
Vorlesung 14