

# Semantic Web Grundlagen

Lösung zur Übung 8: Übungsklausur

Birte Glimm

WS 2011/2012

Übung 8.1:	6 Punkte
Übung 8.2:	3 Punkte
Übung 8.3:	4 Punkte
Übung 8.4:	6 Punkte
Übung 8.5:	12 Punkte
Übung 8.6:	7 Punkte
Übung 8.7:	8 Punkte
Übung 8.8:	9 Punkte
gesamt:	55 Punkte

## Bewertungsschema

Note	Prozent	Punkte
1	95.5	52.5
1.3	90.9	50
1.7	84.8	46.5
2	80.3	44
2.3	75.8	41.5
2.7	69.7	38
3	65.2	36
3.3	60.6	33
3.7	54.5	30
4	50	27.5



Abbildung 1: Grafische Darstellung des RDF Graphen für Aufgabe 8.1

**Aufgabe 8.1.** Beschreiben Sie in natürlicher Sprache die in dem RDF Graphen aus Abbildung 1 gegebenen Informationen. Welche Art von Elementen enthält der Graph? Wie sind die beiden rechteckigen Knoten zu verstehen?

**Lösung (8.1).**

6 Punkte

Der Graph enthält vier Elemente. Der runde beschriftete Knoten repräsentiert einen International Resource Identifier (IRI). Dieser ist über die Property `<http://ex.org/termin>` mit einem leeren Knoten verbunden, dargestellt durch einen unbeschrifteten Knoten. Der leere Knoten hat eine existentielle Semantik und besagt, dass ein entsprechendes Element in jedem Model des Graphen existiert.

Der leere Knoten ist mit zwei getypten Literalen verbunden, die durch einen rechteckigen Kasten symbolisiert werden. Die Verbindung ist einmal über die Property `<http://ex.org/start>` und einmal über `<http://ex.org/dauer>`. Getypte Literale haben die Form `"xxx"^^uri`, wobei `uri` die URI eines Datentyps ist, der angibt, wie die sogenannte lexikalische Form des Literals, `xxx`, zu interpretieren ist. Für das `dateTime` Literal gibt der Teil vor dem `T` das Datum im Format `YYYY-MM-DD` an. Danach folgt die Zeitangabe in der Form `hh:MM:ss`. Dann kommt der Zeitunterschied zur Universal Standard Time (UTC), hier plus zwei Stunden. Der lexikalische Wert des `duration` Datentyps startet immer mit dem Zeichen `P` gefolgt von der Anzahl Jahre, gefolgt von dem Zeichen `Y`, dann die Anzahl der Monate plus dem Zeichen `M`, dann die Anzahl der Tage gefolgt von dem Zeichen `D`, dann folgt das Zeichen `T` und die Zeitangabe von einer Stunde (gefolgt von dem Zeichen `H`) und 30 Minuten (gefolgt von dem Zeichen `M`).

**Aufgabe 8.2.** Markus hat folgendes in RDF modelliert:

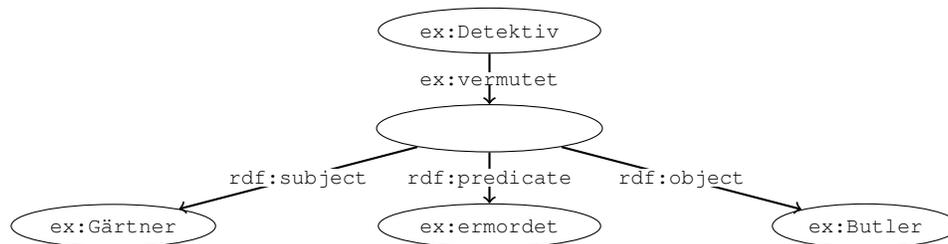
```
@prefix ex: <http://ex.org/> .  
ex:Detektiv ex:vermutet "Der Gärtner hat den Butler ermordet" .
```

Erklären Sie, was Reifikation in RDF ist. Wie können Sie diese sehr grobe Modellierung mittels Reifikation verbessern?

**Lösung (8.2).**

3 Punkte

Mittels Reifikation lassen sich Aussagen über Aussagen modellieren. Dadurch lässt sich auch die Information in dem ungetypten Literal direkt modellieren. Reifikation erlaubt dabei festzuhalten, dass das Fakt `ex:Gärtner ex:ermordet ex:Butler` lediglich eine Vermutung des Detektivs ist. Die Prädikate `rdf:subject`, `rdf:predicate` und `rdf:object` sind dabei Schlüsselworte aus dem RDF Vokabular.



**Aufgabe 8.3.** Bringen Sie den beschreibungslogischen Ausdruck  $\neg(\exists r.(\neg\neg(\neg B \sqcap (\neg C \sqcup C))))$  in Negationsnormalform und vereinfachen Sie diesen wo möglich.

**Lösung (8.3).**

4 Punkte

$$\begin{aligned} & \neg(\exists r.(\neg\neg(\neg B \sqcap (\neg C \sqcup C)))) \\ = & \neg(\exists r.(\neg B \sqcap (\neg C \sqcup C))) && \text{(Vereinfachung Doppelnegation)} \\ = & \neg(\exists r.(\neg B \sqcap \top)) && \text{(Vereinfachung Tautologie)} \\ = & \neg(\exists r.(\neg B)) && \text{(Vereinfachung Tautologie in Konjunktion)} \\ = & \forall r.(\neg(\neg B)) && \text{(Schieben der Negation)} \\ = & \forall r.B && \text{(Vereinfachung Doppelnegation)} \end{aligned}$$

**Aufgabe 8.4.** Peter hat folgendes in Beschreibungslogik modelliert

$$\forall \text{besucht.BachelorVorlesung} \sqsubseteq \text{BachelorStudent}$$

Erkennen Sie ein potentiell Problem mit der Modellierung? Was können Sie verbessern?

**Lösung (8.4).**

6 Punkte

Das Konzept auf der linken Seite der Subsumption wird auch von allen Elementen erfüllt, die keinen besucht-Nachfolger haben, der zur Extension der Klasse BachelorVorlesung gehört. Dies führt dazu, dass derartige Elemente als Instanzen der Klasse Bachelorstudent interpretiert werden. Z.B. Master-Studenten, die ihre Master Arbeit schreiben und daher keine Vorlesungen besuchen oder auch Professoren, die ja keine Vorlesungen besuchen, würden dann als Bachelorstudenten klassifiziert. Um dies zu verhindern, kann man fordern, dass mindestens ein besucht-Nachfolger existiert:

$$\exists \text{besucht.} \top \sqcap \forall \text{besucht.BachelorVorlesung} \sqsubseteq \text{BachelorStudent}$$

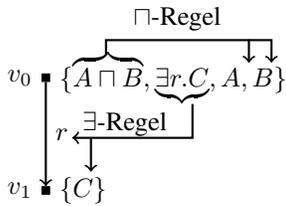
**Aufgabe 8.5.** Gegeben sei die folgende TBox:

$$\mathcal{T} = \{A \sqsubseteq \exists r.(B \sqcap C) \\ \exists r.C \sqsubseteq \neg A\}$$

Ist das Konzept  $A$  erfüllbar bzgl. der TBox? Wenden Sie das Tableau Kalkül an und zeigen Sie grafisch, wie Sie das Tableau entwickeln. Sie müssen nicht für jede Regelanwendung eine neue Grafik zeichnen. Solange die Übersichtlichkeit es erlaubt, können Sie an dem bestehenden Tableau verdeutlichen, welche Regeln Sie angewandt haben. Z.B. gegeben das Tableau

$$v_0 \blacksquare \{A \sqcap B, \exists r.C\}$$

können Sie verdeutlichen, dass Sie die  $\sqcap$ -Regel und die  $\exists$ -Regel angewandt haben:



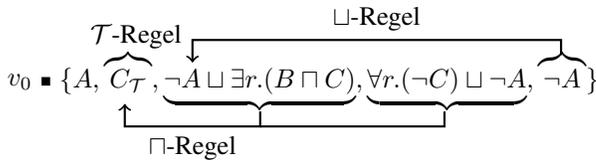
**Lösung (8.5).**

12 Punkte

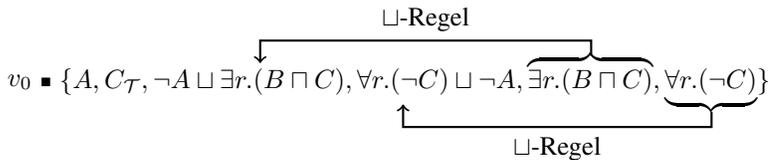
Umwandlung der TBox in das Konzept  $C_{\mathcal{T}}$ :

$$C_{\mathcal{T}} = (\neg A \sqcup \exists r.(B \sqcap C)) \sqcap (\neg(\exists r.C) \sqcup \neg A) \\ = (\neg A \sqcup \exists r.(B \sqcap C)) \sqcap (\forall r.(\neg C) \sqcup \neg A) \quad (\text{NNF})$$

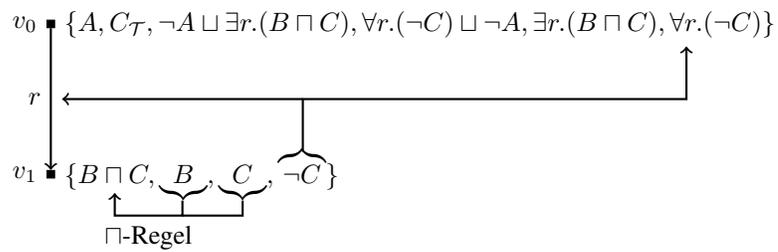
Es soll Erfüllbarkeit getestet werden, also ob mindestens ein Modell existiert, in dem das Konzept  $A$  nicht leer ist. Das Tableau wird daher mit einem Knoten, der  $A$  im Label hat initialisiert. Dann werden die Tableau Regeln angewandt:



Die Wahl von  $\neg A$  führt zu einem Widerspruch, daher Backtracking mit dem anderen Teil der Disjunktion:



Nun können wir die  $\exists$ -Regel auf  $\exists r.(B \sqcap C)$  anwenden, um einen neuen Knoten und eine neue Kante hinzuzufügen:



Wieder gibt es einen Widerspruch. Für die Disjunktion  $\forall r.(\neg C) \sqcup \neg A$  würde auch der andere Teil der Disjunktion zu einem Widerspruch führen, wie bereits bei der Disjunktion  $\neg A \sqcup \exists r.(B \sqcap C)$  getestet. Da es keine weiteren Wahlmöglichkeiten gibt, ist das Konzept  $A$  unerfüllbar bzgl. der TBox.

**Aufgabe 8.6.** Gegeben seien die beschreibungslogischen Axiome  $\exists r.(A \sqcup B) \sqsubseteq C$  und  $C \sqsubseteq \forall s.(B \sqcap \neg D)$ . Wenden Sie die optimierte strukturelle Transformation an, die auch die Polarität beachtet, um die Axiome in einfachere Axiome zu transformieren. Vereinfachen Sie wo möglich.

**Lösung (8.6).**

7 Punkte

$$\exists r. \underbrace{(A \sqcup B)}_{C_1} \sqsubseteq C \qquad C \sqsubseteq \forall s. \underbrace{(B \sqcap \neg D)}_{C_2}^{C_3}$$

$A \sqcup B$  kommt negativ in dem ersten Axiom vor,  $B \sqcap \neg D$  und  $\neg D$  kommen positiv in dem zweiten Axiom vor:

$$\begin{aligned} A \sqcup B &\sqsubseteq C_1 \\ C_2 &\sqsubseteq B \sqcap C_3 \\ C_3 &\sqsubseteq \neg D \end{aligned}$$

Die Disjunktion auf der linken Seite und die Konjunktion auf der rechten Seite kann vereinfacht werden. Das negierte Konzept auf der rechten Seite kommt konjunktiv verknüpft und nicht mehr negiert auf die linke Seite:

$$\begin{array}{ll} A \sqsubseteq C_1 & B \sqsubseteq C_1 \\ C_2 \sqsubseteq B & C_2 \sqsubseteq C_3 \\ C_3 \sqcap D \sqsubseteq \perp & \end{array}$$

Die transformierten Axiome sind somit:

$$\begin{array}{ll} \exists r. C_1 \sqsubseteq C & C \sqsubseteq \forall s. C_2 \\ A \sqsubseteq C_1 & B \sqsubseteq C_1 \\ C_2 \sqsubseteq B & C_2 \sqsubseteq C_3 \\ C_3 \sqcap D \sqsubseteq \perp & \end{array}$$

**Aufgabe 8.7.** Gegeben seien die folgenden RDF Tripel (die Nummerierung dient nur dazu, dass Sie auf die einzelnen Tripel Bezug nehmen können):

```
@prefix ex:    <http://ex.org/> .
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

ex:SemWebBuch rdf:type          ex:Lehrbuch .                (1)
ex:SemWebBuch ex:hatTitel      "Semantic Web Grundlagen" .   (2)
ex:hatTitel   rdfs:domain      ex:Veröffentlichung .         (3)
ex:hatTitel   rdfs:domain      ex:Buch .                     (4)
ex:Lehrbuch  rdfs:subClassOf  ex:Buch .                     (5)
```

Welche Antworten hat die Anfrage

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX ex:  <http://ex.org/>
SELECT ?typ WHERE { ex:SemWebBuch rdf:type ?typ }
```

unter dem RDFS Entailment Regime, also unter RDFS Semantik. Begründen Sie ihre Ergebnisse. Schlussfolgerungsregeln die für das RDFS Schlussfolgern relevant sind, finden Sie auf der letzten Seite.

**Lösung (8.7).**

8 Punkte

```
?typ ⇨ ex:Lehrbuch wegen (1)
?typ ⇨ ex:Buch wegen (1), (5) und der Regel rdfs9
?typ ⇨ ex:Veröffentlichung wegen (2), (3) und der Regel rdfs2
?typ ⇨ rdfs:Resource wegen (1) und der Regel rdfs4a
```

**Aufgabe 8.8.** Gegeben sei die folgende SPARQL Abfrage:

```
PREFIX ex:      <http://ex.org/>
SELECT ?name ?mbox
WHERE
  { FILTER regex(?name, "Ma*")
    { ?x ex:surname ?name } UNION
    { ?x ex:lastname ?name } OPTIONAL
    { ?x ex:mbox ?mbox }
  }
```

Beschreiben Sie in natürlicher Sprache, was die Abfrage selektiert. Wandeln Sie das Abfragemuster der Abfrage in einen SPARQL Algebra Ausdruck um. Sie können dabei das leere Muster  $Z$  sofort vereinfachen. Sie können den ersten Schritt, in dem abgekürzte IRI in voll qualifizierte umgewandelt werden, auslassen und auch in der Algebra mit abgekürzten IRIs arbeiten. Welche Abfrageformen ausser SELECT stellt SPARQL zur Verfügung und wann verwenden Sie diese?

**Lösung (8.8).**

9 Punkte

```
Filter(regex(?name, "Ma*"),
  LeftJoin(
    Union(
      Join(
        Z,
        Bgp(?x ex:surname ?name)
      ),
      Bgp(?x ex:lastname ?name)
    ),
    Bgp(?x ex:mbox ?mbox),
    true
  )
)
```

Nach der Vereinfachung:

```
Filter(regex(?name, "Ma*"),
  LeftJoin(
    Union(
      Bgp(?x ex:surname ?name),
      Bgp(?x ex:lastname ?name)
    ),
    Bgp(?x ex:mbox ?mbox),
    true
  )
)
```

Die Abfrage selektiert Namen ( $?name$ ) und Mailadressen ( $?mbox$ ), so dass die Bindungen für  $?name$

entweder mit der Property `ex:surname` oder mit der Property `ex:lastname` von einem Knoten (`?x`) in dem abgefragten Graphen aus zu erreichen sind. Die Bindung für `?mbox` ist optional und werden nur als Antwort geliefert, wenn `?x` einen `ex:mbox`-Nachfolger hat. Die Bindungen für `?name` werden auch noch nach einem regulären Ausdruck gefiltert, müssen als mit `Ma` beginnen.

Ausser der `SELECT` Form, die Ergebnisse (tabellarisch) liefert, gibt es noch `ASK` für Boolesche Anfragen, `CONSTRUCT` zur Konstruktion von RDF Graphen aus den Lösungen durch Instanziierung eines gegebenen Musters und `DESCRIBE`, welches in nicht-spezifizierter Form die Lösungen durch RDF Tripel beschreibt.