Assignment 3 (Partial Solutions)

Camilo Thorne

Room 00.012
Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart
+49 (0) 711 685-81369
camilo.thorne@ims.uni-stuttgart.de

Semantic Web, SS 2017

Institut für Maschinelle Sprachverarbeitung

Outline

1. OWL Modeling
2. OWL and SPARQL Equivalence
3. OWL and Description Logics

Institut für Maschinelle Sprachverarbeitung

Outline

1. OWL Modeling
2. OWL and SPARQL Equivalence
3. OWL and Description Logics

Institut für Maschinelle Sprachverarbeitung

OWL Modeling - Solution I

1. The Khumbu is a Nepalese region:
   `ex:Khumbu rdf:type ex:NepaleseRegion.`
2. A Nepalese region is a region:
   `ex:NepaleseRegion rdfs:subClassOf ex:Region.`
3. When A has climbed mountain M, M has been climbed by A:
   `ex:hasClimbed owl:inverseOf ex:hasBeenClimbedBy.`
4. Tenzing Norgay is a different person than Edmund Hillary:
   `ex:EdmundHillary owl:differentFrom ex:TenzingNorgay.`

OWL Modeling - Solution II

5. One thing among many that makes something an interesting mountains is that it has been climbed by Tenzing Norgay

```
ex:InterestingMountain rdfs:subClassOf [
   rdf:type owl:Restriction ;
   owl:onProperty ex:hasBeenClimbedBy ;
   owl:hasValue ex:TenzingNorgay ;
] .
```

6. Mountains and cities are located in only in regions, regions are the only thing they can be located in

```
ex:locatedInRegion rdfs:domain [
   owl:unionOf (ex:Mountain ex:City)
   ] .
ex:locatedInRegion rdfs:range ex:Region .
```

OWL Modeling - Solution III

7. A Nepalese Mountain is defined as something that is a mountain and it is located in one of the regions of Nepal, nothing else is a Nepalese mountain:

```
ex:NepaleseMountain owl:equivalentClass [
   rdf:type owl:Class ;
   owl:intersectionOf ( ex:Mountain
       [ rdf:type owl:Restriction ;
         owl:onProperty ex:locatedInRegion ;
         owl:someValuesFrom ex:NepaleseRegion
       ]
   )
] .
```

Outline

1. OWL Modeling
2. OWL and SPARQL Equivalence
3. OWL and Description Logics

OWL DL Equivalence

Show that the following equivalences hold:

1. $\exists_{\geq 1} r.A \equiv \exists r.A$
2. $\neg(\neg A \sqcap \neg B) \equiv A \sqcup B$
3. $A \sqcap A \equiv A$
4. $A \sqcap \bot \equiv \neg(A \sqcup \neg A)$

Institut für Maschinelle Sprachverarbeitung

OWL DL Equivalence - Solution

Two OWL DL / description logic concept descriptions A and B are equivalent, in symbols $A \equiv B$ whenever, irrespective of any ontology $\mathcal{O}$, for every interpretation $\mathcal{I}$, $A^{\mathcal{I}} = B^{\mathcal{I}}$ (i.e., set identity holds). Let $\mathcal{I}$ be an arbitrary interpretation, then:

1. $(\exists_{\geq 1} r.A)^{\mathcal{I}}$ implies that if $d \in (\exists_{\geq 1} r.A)^{\mathcal{I}}$, there exists an $r^{\mathcal{I}}$-successor $d' \in A$, hence $(\exists_{\geq 1} r.A)^{\mathcal{I}} \subseteq (\exists r.A)^{\mathcal{I}}$; the converse (i.e., $(\exists r.A)^{\mathcal{I}} \subseteq (\exists_{\geq 1} r.A)^{\mathcal{I}}$) follows by the same argument, by observing that if an $r^{\mathcal{I}}$-successor exists, then there is at least one of them; therefore $(\exists_{\geq 1} r.A)^{\mathcal{I}} = (\exists r.A)^{\mathcal{I}}$
2. $(\neg(\neg A \sqcap \neg B))^{\mathcal{I}} = \Delta \setminus ((\Delta \setminus A^{\mathcal{I}}) \cap (\Delta \setminus B^{\mathcal{I}})) = A^{\mathcal{I}} \cup B^{\mathcal{I}} = (A \sqcup B)^{\mathcal{I}}$ (by De Morgan's)
3. $(A \sqcap A)^{\mathcal{I}} = A^{\mathcal{I}} \cap A^{\mathcal{I}} = A^{\mathcal{I}}$ ($\cap$ is idempotent among sets)
4. $(A \sqcap \bot)^{\mathcal{I}} = A^{\mathcal{I}} \cap \emptyset = \emptyset = \Delta \setminus (A \sqcup \neg A)^{\mathcal{I}} = \Delta \setminus \Delta$

Institut für Maschinelle Sprachverarbeitung

SPARQL Equivalence I

Q1:
```
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?film ?director ?y WHERE {
 ?film rdf:type dbo:Film .
 ?film dbo:director ?director .
 ?director dbo:birthYear ?y .
} ORDER BY ?director
LIMIT 100
```

Q2:
```
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT DISTINCT ?film ?director ?y WHERE {
 ?film rdf:type dbo:Film .
 ?film dbo:director ?director .
 ?director dbo:birthYear ?y .
} ORDER BY DESC(?director)
LIMIT 100
```

SPARQL Equivalence II

Q3:
```
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?film ?director ?y WHERE {
 ?film rdf:type dbo:Film .
 ?film dbo:director ?director .
 ?director dbo:birthYear ?y .
 FILTER(?y = "1959"^^xsd:gYear)
} ORDER BY ?director
LIMIT 100
```

Q4:
```
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?film ?director ?y WHERE {
 ?film rdf:type dbo:Film .
 ?film dbo:director ?director .
 ?director dbo:birthYear
    "1959"^^<http://www.w3.org/2001/XMLSchema#gYear>.
 ?director dbo:birthYear ?y .
} ORDER BY ?director
LIMIT 100
```

SPARQL Equivalence - Solution

- Q1 $\not\equiv_{\mathcal{O}_{\text{DBpedia}}}$ Q2. This is because Q1 sorts answers in alphabetic (ascending) order, while Q2 sorts in (alphabetic) descending order. Interestingly the computation is very similar: all triples of directors, films and years are first returned (similar part), are then sorted and then cut to the first 100 answers. **Computation time is identical**.

- Q3 $\equiv_{\mathcal{O}_{\text{DBpedia}}}$ Q4. This is because the answers are identical: films from directors born in 1959. **The computation in each case is very different**. Q1 first computes **all** director, birth year and film triples, and then filters out cases where the birth year is different from 1959. Q2 on the other hand **searches only** for the triples where birth year is 1959 (restricts by join rather than by filter) rather than searching for all and then filtering. The computation time of Q2 is thus in principle much lower than that of Q1.

Institut für Maschinelle Sprachverarbeitung

Outline

1. OWL Modeling

2. OWL and SPARQL Equivalence

3. OWL and Description Logics

Description Logic Ontologies I

You are given the following Description Logic ontology fragment:

WildPig(rooter) pulledBy(hogfatherssleigh, rooter) belongsTo(hogfatherssleigh, hogfather)

We want to derive new knowledge about "rooter".

- Write the following statements in Description Logic syntax. Use the given class and property names:
  1. The Hogfather and Death are anthropomorphic personifications (APs). (AnthropomorphicPersonification, Death, Hogfather)
  2. Among other things, a wild pig sleigh is pulled only by wild pigs. (WildPigSleigh, pulledBy, WildPig)

Institut für Maschinelle Sprachverarbeitung

Description Logic Ontologies II

3. Discworld godlike beings can either be Discworld gods or APs and nothing else.

   (AnthropomorphicPersonification, DWGod, DWGodlikeBeing)

4. No-one can be an AP and a human at the same time.

   (AnthropomorphicPersonification, Human)

- What else do we know for certain now?

Description Logic Ontologies - Solution

1. AnthropomorphicPersonification(hogfather), AnthropomorphicPersonification(death)
2. $\text{WildPigSleigh} \sqsubseteq \forall \text{pulledBy.WildPig}$
3. $\text{DWGodlikeBeing} \equiv \text{DWGod} \sqcup \text{AnthropomorphicPersonification}$
4. $\text{AnthropomorphicPersonification} \sqcap \text{Human} \equiv \bot$

We can now derive (by entailment) the following new knowledge from the extended ontology:

- "hogfather" and death are antropomorphic representation and hence both godlike beings and gods – (1) and (3)
- "death" and "hogfather" are not human – (4) and (1)
- "hogfathersseligh" is a wild pig sleigh – (2)[1]

---

[1] (2) asserts a kind of domain restriction for **pulledBy**, viz., it is equivalent to $\exists \text{pulledBy}^{-}.\text{WildPig} \sqsubseteq \text{WildPigSleigh}$.

Institut für Maschinelle Sprachverarbeitung

Description Logic Reasoning

You are given the following Description Logic ontology $\mathcal{O}_a$:

- Penguin $\sqsubseteq$ Bird
- Bird $\sqsubseteq$ Flying
- Penguin $\sqcap$ Flying $\sqsubseteq \bot$
- Bird $\sqcap$ Giraffe $\sqsubseteq \bot$
- Pengin(skipper)
- Giraffe(melman)

Reason **model-theoretically** to answer the following:

1. Show that the ontology $\mathcal{O}_a$ is **(in)consistent**.
2. Show that $\mathcal{O}_a$ **does not entail** that:
   - neither melman flies, nor
   - giraffes fly.

Institut für Maschinelle Sprachverarbeitung

Description Logic Reasoning - Solution I

The ontology $\mathcal{O}_a$ is **inconsistent**. Indeed, $\mathcal{O}_a$ **entails** that **(1)** Skipper is a bird (penguins are birds) and **(2)** that skipper flies (all birds fly). However, Penguin $\sqcap$ Flying $\sqsubseteq \bot$ is **equivalent** to Penguin $\sqsubseteq \neg$Flying which, together with **(1)** entails **(3)** that Skipper does not fly. But: **(3) is in contradiction with (2)!** Therefore $\mathcal{O}_a \models \bot$, which implies that (course slides 9, slide 24) $\mathcal{O}_a$ is inconsistent, which means (same slide) that it has no model.

As a result, it can "entail" **anything**, as $Mod(\mathcal{O}_a) = \emptyset$ and the empty set of models is contained in any set of models ("ex falso sequitur quodlibet")[2]. However, as it has no models is doesn't describe ny knowledge either and is useless.

[2] This is consequence of the following fact:
for every ontology $\mathcal{O}$ and assertion α, $\quad \mathcal{O}_a \models \alpha \iff Mod(\mathcal{O}) \subseteq Mod(\alpha)$

Description Logic Reasoning - Solution II

Suppose we modify $\mathcal{O}_a$ as follows:

- Bird $\sqsubseteq$ Flying
- Bird $\sqcap$ Giraffe $\sqsubseteq \bot$
- Bird(skipper)
- Giraffe(melman)

Define $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$ as follows:

$\Delta = \{\text{melman}, \text{skipper}\}$
$\text{Bird}^{\mathcal{I}} = \text{Flying}^{\mathcal{I}} = \{\text{skipper}\}$
$\text{Giraffe}^{\mathcal{I}} = \{\text{melman}\}$

- $\mathcal{I} \models \mathcal{O}_a$ as in $\mathcal{I}$ birds and giraffes denote **disjoint sets**.
- $\mathcal{O}_a \not\models$ Flying(melman), as $\mathcal{O}_a \not\models$ Griraffe $\sqsubseteq$ Flying, as $\mathcal{I} \models \mathcal{O}_a$ but $\mathcal{I} \models \mathcal{O}_a \not\models$ Flying(melman).
- $\mathcal{O}_a \not\models$ Griraffe $\sqsubseteq$ Flying, as $\mathcal{I} \models \mathcal{O}_a$ but $\mathcal{I} \models \mathcal{O}_a \not\models$ Giraffe $\sqsubseteq$ Flying.

Institut für Maschinelle Sprachverarbeitung

Domain and Range

Description Logic semantics maps, broadly speaking, Description Logic concepts, properties and statements to sets, set-valued relations, and set-theoretical statements, resp.

1. Convert the expressions to set theory using Description Logic semantics and explain why $\exists r.\top \sqsubseteq A$ expresses that the domain of r is A?

2. Convert the expressions to set theory using Description Logic semantics and explain why $\top \sqsubseteq \forall r.A$ expresses that the range of r is A?

Institut für Maschinelle Sprachverarbeitung

Domain and Range – Solution

Let $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$ be an arbitrary set theoretical interpretation.

- `r rdfs:domain A.` **is equivalent to** $\exists r.\top \sqsubseteq A$:

  $\mathit{dom}(r^{\mathcal{I}}) \subseteq A^{\mathcal{I}}$ iff
  $\{d \in \Delta \mid \text{ exists } d' \in \Delta \text{ such that } (d, d') \in r^{\mathcal{I}}\} \subseteq A^{\mathcal{I}}$ iff
  $\{d \in \Delta \mid \text{ exists } d' \in \top^{\mathcal{I}} \text{ such that } (d, d') \in r^{\mathcal{I}}\} \subseteq A^{\mathcal{I}}$ iff
  $(\exists r.\top)^{\mathcal{I}} \subseteq A^{\mathcal{I}}$
  (**note:** by set theory, it holds that:
  $\mathit{dom}(r^{\mathcal{I}}) = \{d \in \Delta \mid \text{ exists } d' \in \Delta \text{ such that } (d, d') \in r^{\mathcal{I}}\}$)

- `r rdfs:range A.` **is equivalent to** $\top \sqsubseteq \forall r.A$:

  We observe that $\top \sqsubseteq \forall r.A$ is equivalent to $\exists r^{-}.\top \sqsubseteq A$, and then repeat the previous argument.