Principles of Knowledge Representation and Reasoning

B. Nebel, S. Wölfl, J. Hué
M. Westphal
Winter Semester 2012/2013

University of Freiburg
Department of Computer Science

Exercise Sheet 13

Due: February 6th, 2013

Exercise 13.1 (Decidabiltiy of Description Logics, 1+1+1)

In the lecture it was shown that satisfiability of concept descriptions is decidable, when one restricts to the following constructs:

- Concepts: $C \sqcap D$, $C \sqcup D$, $\neg C$, $\forall r.C$, $\exists r.C$, $r \sqsubseteq s$
- Roles: $r \sqcap s$, $r \sqcup s$, $\neg r$, r^{-1}

The idea of the proof is based on the fact that the FOL fragment L_2 is decidable. Provide for the following concept descriptions translations into L_2 such that the concepts are satisfiable if and only if their L_2-translations (FOL formulae without free variables) are satisfiable.

(a) $\forall r^{-1}.(C \sqcap \neg \exists s.D)$

(b) $\forall r \sqcap s.\big(\forall t.(\neg C \sqcup \exists r.D)\big)$

(c) $\exists r.\big((C \sqcup (\neg s \sqsubseteq t)) \sqcap \forall s.(t \sqsubseteq u)\big)$

Exercise 13.2 (The Ontology Editor Protégé, 2+2)

Download and install the ontology editor Protégé from the web page:

`http://protege.stanford.edu/`.

(a) Load the mad cows ontology from the URI:

`http://www.cs.man.ac.uk/~horrocks/OWL/Ontologies/mad_cows.owl`

into the editor. Provide the DL expressivity used in this ontology. Further, provide a list of unsatisfiable concepts in this ontology: use one of the reasoners Hermit, FaCT++, RacerPro or Pellet which come with Protégé or can be installed via the plugin interface.[1]

(b) Explain the inconsistencies in a formally precise way.

Exercise 13.3 (Modeling in Protégé, 2+1+2)

(a) Generate a new ontology with Protégé and specify the family TBox from the lecture (chapter 11, slide 25).

(b) Extend the family TBox by the concept `Patricide` by using a new role name `isMurdererOf`.

[1] For a tutorial on how to use Protégé consult
`http://owl.cs.manchester.ac.uk/tutorials/protegeowltutorial/resources/ProtegeOWLTutorialP4_v1_2.pdf`.

(c) Check (and explain) whether for the following Oedipus-ABox $\mathcal{A}_{oe}$ it holds: Jocasta has a child that is a patricide, who in turn has a child that is no patricide.

`hasChild(JOCASTA,OEDIPUS)`	`hasChild(JOCASTA,POLYNICES)`
`hasChild(OEDIPUS,POLYNICES)`	`hasChild(POLYNICES,THERSANDROS)`
`Patricide(OEDIPUS)`	`not Patricide(THERSANDROS)`

Please email your ontologies as OWL-files to `westpham@informatik.uni-freiburg.de`.