Principles of Knowledge Representation and Reasoning

B. Nebel, S. Wölfl, F. Lindner — University of Freiburg
Winter Semester 2015/2016 — Department of Computer Science

Exercise Sheet 5

Due: November 25th, 2015

Exercise 5.1 (Domain Modeling using Modal Logics, 5)

In this exercise you are asked to become familiar with the theorem prover SPASS, which you can download here: http://www.spass-prover.org . (Check the SPASS website for a tutorial and the `/doc` folder in downloaded archive file for a handbook and an overview about the SPASS syntax. You also find a minimal example on our website as attachment to this exercise, viz., `example.dfg`.) In particular we will make use of the modal logics capabilities of SPASS to represent and reason about two simple domains. For each of these domains you will find a file on the web prepared to be used by you to fill in axioms that model the domain as well as conjectures to be proved. Please send your implementations with comments(!) to `lindner@informatik.uni-freiburg.de`.

(a) Adam und Eve have beliefs about a concrete apple, which we denote by *apple*. Use two modal belief operators to represent the following statements and write them as axioms into the file `adameve.dfg`:

  i) If Adam believes that Eve believes that *apple* is sweet if it is red, then Adam believes that as well.

  ii) Both Adam and Eve believe that, if *apple* is red, the respective other believes that *apple* is sweet.

  Given (a) and (b), does it follow that Adam believes that the apple is sweet? Check if this conjecture follows from your axioms by first writing this conjecture into `adameve.dfg` and then running `./SPASS adameve.dfg -DocProof`. Among others, you should see the message *SPASS beiseite: Proof found.* as well as a proof.

(b) The robot Robo-C works as a servant in a household. The flat consists of four rooms. Being in the bedroom, by moving east one reaches the living room. From the living room one can go back to the bed room by moving west, one can reach the balkony by moving north, and one reaches the kitchen by moving east. From the balkony one reaches the living room by moving south. From the kitchen one can go to the living room by moving west.

  Model the possibilities to get from one room to the other by using as many box operators as there are actions. Formalize the reachability by filling in axioms into the file `robotservant.dfg`. Prove that if the robots starts in the bedroom: If it first goes east twice, then goes west, and finally goes north, then the robot ends up on the balkony. Based on what you know from the lecture, can you put the proof provided by SPASS in your own words?

Exercise 5.2 (Disjunctive Concepts in Inheritance Networks, 2)

In this exercise we consider strict inheritance networks that allow for expressing disjunctive concepts of the form C_1 **or** $\ldots$ **or** C_k. A formula of the form

$$\phi = C_1 \textbf{ or } \ldots \textbf{ or } C_k \textbf{ isa } C'_1 \textbf{ or } \ldots \textbf{ or } C'_l$$

is called a *disjunctive inheritance formula* and its intended interpretation is given by the first-order translation $\pi(\phi) = \forall x\big((C_1(x) \vee \cdots \vee C_k(x)) \to (C'_1(x) \vee \cdots \vee C'_l(x))\big)$, where the C_i and C'_i are atomic concept terms.
Show that the inheritance problem for disjunctive inheritance networks (given a set of disjunctive inheritance formulae Θ and a disjunctive inheritance formula ϕ, does $\pi(\phi)$ follow from $\{\pi(\vartheta) \mid \vartheta \in \Theta\}$?) can be solved in polynomial time.

Hint: Instead of the translation into first-order logic provide a translation of disjunctive inheritance formulae into propositional logic. You may use the deduction theorem for propositional logic and the fact that the validity of a given set of propositional anti-Horn-formulae can be decided in polynomial time.

Exercise 5.3 (Inheritance networks with negation, 5)

For this practical assignment, you are asked to implement reasoning in simple inheritance networks that feature negation but no conjunction. Your task is to implement the graph-based approach to decide the inheritance problem for inheritance networks with negation as presented in the lecture.

Input: Your program should be invoked the following way:

```
./inheritance-simple <knowledge base> <query>
```

Here, `<knowledge base>` is the name of a file that contains the knowledge base, and `<query>` is a string[1] that contains an **isa**-statement which is to be evaluated on the knowledge base. Each line in the knowledge base must also be an **isa**-statement. These statements are written as: `concept1 isa concept2`. Both `concept1` and `concept2` are concept terms in either one of the following forms:

- *atomic concepts* are terms consisting of letters, numbers, and hyphens. Not allowed are the keywords `isa` and `not`.
- *negated concepts* are written as terms `not concept`, where `concept` is an atomic concept.

Output: Depending on whether the query follows from the knowledge base the output should be either `Yes.` or `No.` .

Please send your source code to `lindner@informatik.uni-freiburg.de` until **December 2nd**.

[1] Usually, when you are working on a shell you have to put the string in quotation marks to protect the spacing.