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 5

Due: November 28th, 2012

Exercise 5.1 (Reasoning in Default Logic, 2)

Translate into default first order logic and check whether the given conclusion follows credulously and/or skeptically.

(a) *Bob is a criminal. Typically, criminals are not innocent. People who don't get accused are usually not criminal. But each person is considered innocent as long as the contrary is not proven.* Conclusion: *Bob gets accused.*

(b) *Typically, computer scientists like computers. People who like computers and are female are typically interested in psychology. A typical computer scientist is female, as for example Anne.* Conclusion: *Anne is interested in psychology.*

Remark: In (a), (b) you may assume that there is only one individual (Bob or Anne, respectively).

Exercise 5.2 (Extensions, 2+2)

Let S be a set of propositional logic formulae. Consider the set of defaults $D = \{\frac{\top : \phi}{\phi} | \phi \in S\}$. A *maximal consistent* subset of S is a subset C of S such that C is satisfiable, but each proper superset $C \subsetneq C' \subseteq S$ is not. Show the following statements:

(a) If C is a maximal consistent subset of S, then the default theory $\langle D, \emptyset \rangle$ has an extension E such that $C = E \cap S$.

(b) If E is an extension of $\langle D, \emptyset \rangle$, then $E \cap S$ is a maximal consistent subset of S.

Exercise 5.3 (Tableau solver for modal logic K, 6)

Write a tableau solver for modal logic K. You can use any programming language you like (given that it is usable under Ubuntu 12). You can re-use the code you made for parsing input formulae and the one used for propositional formula tableau proof. Source code **must be submitted** on time to: `westpham@informatik.uni-freiburg.de`.

The solver should take as input a formula from modal logic with the format based on the one given on the Exercise Sheet 2[1]. We extend this language with the following connectives:

- [ ] representing the box ($\Box$) operator.
- <> representing the diamond ($\Diamond$) operator.

[1] `http://www.informatik.uni-freiburg.de/~ki/teaching/ws1213/krr/`

The solver should take as an option within the command line:

–**check_sat** and in that case, provide as a result either SAT followed by a model or UNSAT.

–**check_valid** and in that case, provide as a result either FALSIFIABLE or VALID.

For example, the formula $\Box(\neg a \vee b) \wedge \Box\neg a \wedge \Diamond b$ would be represented by:

`a not b or [] a neg [] and b <> and`

The tableau solver must then apply the following rules:

$\frac{w \models \Box\varphi,\, wRv}{v \models \varphi}$	If $\Box\varphi$ is true in w and v can be reached from w, then the formula φ must be true in v.
$\frac{w \not\models \Box\varphi}{wRv,\, v \not\models \varphi}$	If $\Box\varphi$ is false in w, then the formula φ must be false in some world v. Thus a new branch v must be created where φ is false.
$\frac{w \models \Diamond\varphi}{wRv,\, v \models \varphi}$	If $\Diamond\varphi$ is true in w, then the formula φ must be true in some world v. Thus a new branch v must be created where φ is true.
$\frac{w \not\models \Diamond\varphi,\, wRv}{v \not\models \varphi}$	If $\Diamond\varphi$ is false in w and v can be reached from w, then the formula φ must be false in v.

It is thus necessary to integrate possible worlds into your implementation.
The main idea of the solver does not change with respect to the lecture or to the propositional tableau. For satisfiability solving, apply tableau rules as long as it can create new formulas. If a contradiction within any of the possible world can be derived, then backtrack to the last disjunction. If no new formula can be deduced and there is no contradiction, then build a model. If all branches have been explored without a solution found we can answer that the formula is unsatisfiable.