Exercise 6

Foundations of Agent Programming

Prof. Michael Thielscher, Sebastian Voigt

International Master Program in Computational Logic — summer term 2009

11.06.2009

Exercise 6.1

Download the file 4_wumpus_excerpt.pl from the course web page, it contains some of the rules of the *Wumpus Domain* presented in the lecture. Additionally load FLUX (4_flux.pl) and its constraint handling rules (4_fluent.chr). Solve the following tasks using these files with Eclipse prolog.

a) Encode an initial state z_0 such that the agent is at an unknown position in the first row (considering a field of size $(4, 4)$) and faces either north or east. Furhtermore there is exactly one wumpus.

b) Give a query that formulates that the robot is facing west in z_0 (the query should fail).

c) Show that assuming the wumpus not to be at $(2, 1)$, the robot to be at $(1, 1)$ and sensing a stench (all in z_0) leads to the conclusion that the wumpus is at $(1, 2)$.

d) Show that after turning right in z_0 the assumption of facing north or west is wrong.

e) What can we conclude about the robot's position after successfully executing the action sequence $Turn\_Left$, Go, Go, Go starting in z_0? How can we show this with a query?

Exercise 6.2

The mathematical game *Towers of Hanoi* consists of three rods and a number of disks of different sizes. In the initial situation all disks are stacked in order of size on one rod, the smallest on top. The objective of the game is to move the entire stack to another rod by a sequence of moves. A move consists of placing the upper disk of a rod r on top of another rod r', which can only be done if r' is empty or its topmost disk is greater than the one to be placed.

We use the following fluents and events:

$Clear(d, r)$	$\hat{=}$	disk $d \in \mathbb{N}$ is in topmost position on rod $r \in \{Rod_1, Rod_2, Rod_3\}$
$Table(d, r)$	$\hat{=}$	disk d is in bottommost position on rod r
$On(d_1, d_2, r)$	$\hat{=}$	disk d_1 is directly on disk d_2 on rod r
$Move(d, r_1, r_2)$	$\hat{=}$	disk d is moved from rod r_1 to rod r_2

In the following we use ϕ to denote an arbitrary first order formula, f to denote a fluent or its negation, e to denote an event and t to denote a time. Solve the following tasks in the event calculus:

a) Give an initial state axiomatization such that the tower is on rod Rod_1 initially. The specifying formulas should be of the form $\phi \supset \mathit{Initially(f)}$ and use a free variable n which represents the number of disks the tower consists of.

b) Provide a precondition axiomatization using formulas of the form $Happens(e, t) \supset \phi$.

c) Specify state update axioms of the form $\phi \supset \mathit{Initiates(e,f,t)}$ and $\phi \supset \mathit{Terminates(e,f,t)}$.

d) Formulate a goal axiom such that the tower has to be placed at Rod_3 eventually.

Download the file `6.2_toh.pl` from the course web page. It contains an ASP encoding of the event calculus axiomatization from Sl. 5/8 as well as domain definitions for all neccessary variables. Test it using the Answer Set Solver `clingo` (cf. Lab2, link at the web page).

e) Encode the axiomatization from a) - d) in `6.2_toh.pl`. Running the file with `clingo` should then give a set of $Happens(e, t)$ predicates which represent the solution to the Towers of Hanoi problem.