Computer lab assignment 3

Reasoning Agents

Prof. Michael Thielscher, Sebasian Voigt

International Masters Programme in Computational Logic — summer term 2009

12.06.2009

Assignment 3.1

In the *Knight's Tour Problem* a knight is placed on an empty chessboard with variable size. The goal is to find a sequence of knight moves such that every position of the board is visited exactly once (cf. Figure 1).

For an implementation we use the fluents

$At(x, y)$	to denote that the knight is currently at position (x, y)
$NotVis(x, y)$	to denote that position (x, y) has not been visited yet

and the action

$Move(x, y)$	which moves the knight to position (x, y).

Download the file `4_knights.pl` from the course web page. It contains an ALP interpreter in FLUX.

a) Define a predicate $Poss/2$ (the precondition axiom) and a predicate $State\_Update/3$ (the state update axiom) for the action $Move$.

b) Define a predicate $Init(z)$ which assigns z to an initial state according to our specification. Use the two facts $Size(m, n)$ and $Start(x, y)$ which are already defined in `4_knights.pl` in order to enable a variable board size and starting position.

c) Write an ALP that implements a brute force strategy solving the problem. To execute your ALP define a rule `p :- <ALP>` and use the command `run(p)` at the command line.

d) A good heuristics for this problem is to choose the next step position such that it provides the fewest possible further knight's moves. Write another ALP which implements this strategy.

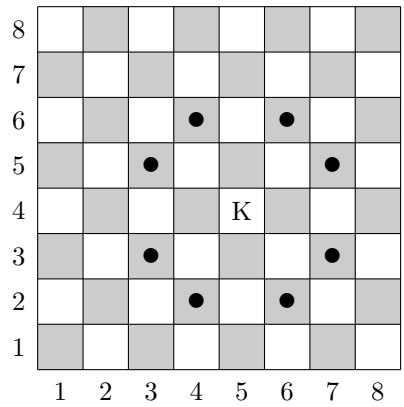

Figure 1: The Knight's Tour Problem: Considering an example chessboard of size 8×8, the eight possible moves of a knight on $(5, 4)$ are depicted. A knight at the border or in a corner has similar but fewer moves.