Computer lab assignment 2

Reasoning Agents

Prof. Michael Thielscher, Sebasian Voigt

International Masters Programme in Computational Logic — summer term 2009

22.05.2009

Assignment 2.1

The game *Lightsout* (you can find a link to a flash implementation on the web page) considers an $n \times n$ grid such that every field in it has exactly one lamp which can be either in state "on" or "off". Each lamp can be switched which results in a change of its own state as well as a change of the states of all lamps that are horizontally or vertically next to it (cf. Figure 1). The goal of the game is to switch off all the lamps. This exercise leads to a solver for instances of this game using Anwer Set Programming. We use the following action and fluent:

$Switch(x, y)$	$\hat{=}$	the lamp at position $(x, y) \in \mathbb{N} \times \mathbb{N}$ is switched
$Cell(x, y)$	$\hat{=}$	the lamp at position (x, y) is on

Download the Answer Set Solver `clingo` via the link from the webpage and solve the following tasks:

a) `clingo` allows the global assignment of a domain to a variable name X via the construct `#domain dom(X).`, assuming an additional predicate $Dom/1$ defining the domain. Provide domain definitions for:

   – T - ranging over time steps which are encoded via natural numbers

   – M - ranging over possible move actions

   – X, Y, X_1, Y_1 - ranging over integers between 1 and 5.

b) Define a predicate $Adjacent(x, y)$ which is true iff $|x - y| = 1$. Test your predicate using `clingo`.

c) Define a predicate $Legal(p, m, t)$ specifying the legal moves m of player p at time t and a predicate $Does(p, m, t)$ such that player p does exactly one move m that is legal at every non-terminal time t.

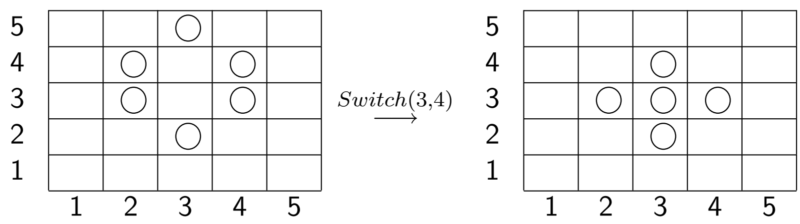

Figure 1: A sample transition step of the Lightsout game, switching on the lamp at position $(3, 4)$.

d) Define a predicate $Affected(x, y, t)$ which is true iff position (x, y) results in a changed state after switching a lamp at some position (x_1, y_1) at time t.

e) Define $Holds(f, t)$ to encode the initial configuration shown in the left hand side of Figure 1 and the state update for every time t.

f) Define a predicate $Terminal(t)$ which indicates whether a state is terminal at time t. Adjust the definitions of $Legal(p, m, t)$ and $Does(p, m, t)$ accordingly.

g) Define a predicate $Goal$ which is true iff at some time we have the goal of all lights being switched off. Restrict the possible models such that only those reaching the goal are considered.

h) Run your answer set program and test if it determines a sequence of just two switch actions (for example $Switch(3, 3)$, $Switch(3, 4)$) to win the game with the encoded initial state.

   Run your program on the harder instance specified in Figure 2. *Hint:* The shortest action sequence needs 11 steps.

i) For every action sequence that solves the game there is another solving sequence which switches each lamp at most once. Think about an efficient encoding of this property.

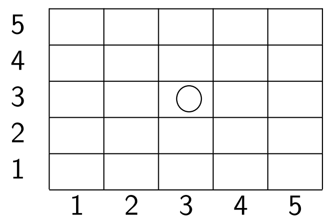

Figure 2: An instance of the Ligthsout game.