Exercise 1

Foundations of Agent Programming

Prof. Michael Thielscher, Sebastian Voigt

International Master Program in Computational Logic — summer term 2009

23.04.2009

Exercise 1.1

Consider the GOLOG program for the Mailbot on Slides II/7-10. Improve the strategy of the agent so as to always pick up the package with the nearest delivery distance from the current location.

Exercise 1.2

The *Blocks World* domain (cf. Figure 1) is concerned with the movement of blocks in a two dimensional environment. Every block is located either on the table or on another block and can have at most one block on its top. Only one block can be moved at a time and only if there is no block on top of it. The table provides sufficient space to hold all blocks.

We use the following fluents and actions:

$Clear(b)$	$\hat{=}$	block b is a topmost block
$On(b_1, b_2)$	$\hat{=}$	block b_1 is directly on Block b_2
$Table(b)$	$\hat{=}$	block b is on the table
$Stack(b_1, b_2)$	$\hat{=}$	Put block b_1 from the table to the top of block b_2
$Unstack(b_1, b_2)$	$\hat{=}$	Put block b_1 from the top of block b_2 to the table

Implement the following procedures in GOLOG:

a) $flattenTower(b)$: Place all the blocks in the tower whose top block is b onto the table.

b) $makeOneTower$: Create a single tower consisting of all the blocks.

c) $reverseTower(b)$: Create a tower consisting of the blocks of the tower whose top block is b , but with the blocks in reverse order.

d) $stackTower_{rev}(b_1, b_2)$: Stack all the blocks in the tower whose top block is b_1 onto the tower whose top block is b_2 . The resulting tower has as its top block the bottom block of the original tower with top block b_1 .

e) $stackTower(b_1, b_2)$: Same as the previous procedure except that the resulting tower has b_1 as its top block.

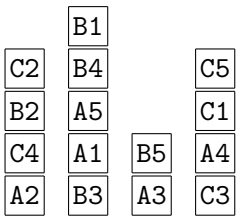

Figure 1: A Blocks World instance

Exercise 1.3

In the board game *Connect Four* (cf. Figure 2) Black and Red take turn in dropping discs of their own color into a seven-column, six-row vertically-suspended grid. The player wins who is the first to have four connecting discs horizontally, vertically, or diagonally.

With the underlying sorts $\text{COLOR} = \{B, R\}$, $\text{X} = \{1, 2, \ldots, 7\}$ and $\text{Y} = \{1, 2, \ldots, 6\}$, we can define the following relational fluents:

$CtrB$:	$\mapsto$ FLUENT	$CtrB \mathrel{\hat{=}}$ Black has next move
$Filled$:	$\text{COLOR} \times \text{X} \times \text{Y} \mapsto \text{FLUENT}$	$Filled(c, x, y) \mathrel{\hat{=}}$ field (x, y) is filled with c-colored disc

The single action is denoted by the function symbol

$Drop$: $\text{COLOR} \times \text{X} \mapsto \text{ACTION}$ $\quad Drop(c, x) \mathrel{\hat{=}}$ a c-colored disc is dropped into column x

Solve the following tasks in the General Action Calculus:

a) Formulate an Initial State Axiom, assuming that Black has the first move.

b) Specify a precondition axiom for the primitive action.

c) Define a state formula which characterizes a predicate $Win(c, s)$ such that $Win(c, s)$ is true if and only if player c has won in situation s.

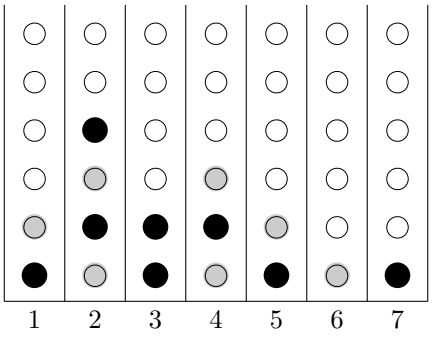

Figure 2: An example situation in the two-player game Connect Four: Red should drop a disc in column 3 to prevent Black from completing a diagonal $2 - 5$. Incidentally, this happens to be a forced win for Red, because Black cannot hinder it to get four discs in a row with the next move.