Computer lab assignment 1

Reasoning Agents

Prof. Michael Thielscher, Sebasian Voigt

International Masters Programme in Computational Logic — summer term 2009

08.05.2009

Assignment 1.1

Download the implementation of the Blocks World from Exercises 1.2 and 2.1 (`2_blocks.pl`) together with the GOLOG interpreter (`2_gologinterpreter.pl`) from the course web page. Solve the following tasks:

a) Use the predicate $Exec/1$ to run the procedure $FlattenTowers$ (both defined in `2_blocks.pl`) that unstacks arbitrary blocks until all blocks are on the table. Encode the initial state shown in Figure 1 and run the procedure again.

b) Implement the procedure $flattenTower(b)$ (cf. Exercise 1.2 a)) which places all the blocks in the tower with top block b onto the table.

c) Implement the procedure $reverseTower(b)$ (cf. Exercise 1.2 c)) which reverses the tower with top block b.

d) **optional** Implement the procedure $stackTower(b_1, b_2)$ (cf. Exercise 1.2 d) and e)) which puts the tower with top block b_1 on top of the tower with top block b_2 such that the resulting tower has top block b_1.

Hint: In this version of the GOLOG interpreter, the expression $B1 = B2$ cannot be used in conditions. However one can define an additional clause for $Holds/2$ to get the intended functionality.

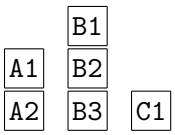

Figure 1: A Blocks World instance

Assignment 1.2

The n *-Queens-Problem* is concerned with the question of how to place n queens on an $n \times n$ chess board such that no two queens can attack each other. The goal of this exercise is to write a nondeterministic GOLOG program to solve this problem. We use the following fluents and actions:

$Size(n)$	$\hat{=}$	the chess board has size $n \times n$
$Occupied(row, col)$	$\hat{=}$	position $(row, col) \in \mathbb{N} \times \mathbb{N}$ is occupied by a queen
$Placed(i)$	$\hat{=}$	i queens are already placed
$Init(n)$	$\hat{=}$	initialize the problem with size n
$Place(row, col)$	$\hat{=}$	place a queen at position (row, col)

- Create a new file which loads the GOLOG interpreter.
- Define precondition axioms for the two actions $Init/1$ and $Place/2$.
- Define initial state axioms and successor state axioms for the three fluents $Size/1$, $Occupied/2$ and $Placed/1$.
- Write a nondeterministic GOLOG program that places queens at positions that are not attacked as long as n queens are placed.

Hints for the precondition axiom of $Place/2$*:*

- You may use the predicate `between(M,N,X)` from the library `util` (to be loaded with `:- lib(util).`) to get values for `X` that range between `M` and `N`.
- Define an additional predicate `attacks(Row,Col,Row1,Col1)` that is true iff a queen at position `(Row,Col)` attacks position `(Row1,Col1)`.