University of Freiburg
De Raedt / Kersting

Exercise Sheet No 2

April 26, 2002

Deadline: May 03, 2002, before the lecture

1 bonus point

To register for the exercises, please email `subo@informatik.uni-freiburg.de`. The subject should be `ai`, and the body should include *last name*, *first name*, and the *Matrikelnummer*.

Exercise 2.1 (3 points)

Consider the accessible, two-location vacuum world under Murphy's Law, i.e. if a room is clean, then the action *suck* may make the room dirty. Show, that for each initial state, there is a sequence of actions that is guaranteed to reach a goal state.

Exercise 2.2 (2 points)

Describe a search space in which *iterative deeping* search performs much worse than *depth-first* search.

Exercise 2.3

So far, we have assumed in the lecture that path costs are not negative. In this exercise, we would like to discuss negative path costs and cyclic paths.

1. Suppose that a negative lower bound $c < 0$ is placed on the cost of any given step, i.e. negative costs are allowed, but the cost of a step cannot be less than c. Does this allow *uniform-cost* search to avoid searching the whole tree? If so, why? (3 points)

2. Suppose that there is a set of operators that form a loop, so that executing the set in some order results in no net change to the state. If all of these operators have negative costs, what does this imply about the *optimal* behaviour for an agent in such an environment. (2 points)

Exercise 2.4

The $SEND + MORE = MONEY$ problem consists in finding distinct digits for the letters D, E, M, N, O, R, S, Y such that S and M are different from zero, i.e. no leading zeros, and the equation

$$SEND + MORE = MONEY$$

is satisfied.

1. Explain in a nutshell, why it would be good to formulate the problem as a *constraint satisfaction problem*? (3 points)

2. Formulate the problem as a *constraint satisfaction problem*, i.e. what are the variables, what constraints do we have, etc. (1 point)

3. Find a solution using the *Backtracking Constraint Solver* incorporating *node consistency* and *arc consistency*. Give the search tree. (Hint: consider the variables in the following order: O, M, Y, E, N, D, R, S.) (6 points)