Übung Künstliche Intelligenz SS 2007
Wirtschaftsinformatik und Maschinelles Lernen (ISMLL)
Prof. Dr. Dr. Lars Schmidt-Thieme , Leandro Marinho

Übungsblatt 2

Abgabe: Mittwoch, 02.05.07 bis 13 Uhr

Exercise 1 Problem Formulation (20 Points)

a) [5 pts.] Define in your own words the terms: state, state space, search tree, search node, goal, action, successor function and branching factor.

b) [10 pts.] Give the initial state, goal state, successor function, and cost function for each of the following. Choose a formulation that is precise enough to be implemented.

- You want to attend the following courses: Machine Learning (WS), Bayesian Networks (WS) and Semantic Web (SS). Artificial Intelligence (SS) is an obligatory requirement for Machine Learning and Bayesian Networks.
- You have to color a planar map using only four colors, in such a way that no adjacent regions have the same color.
- A 3-foot-tall boy is in a room where some bananas are suspended from the 8-foot ceiling. He would like to get the bananas. The room contains two stackable, movable, 3-foot-high crates.

Exercise 2 Uninformed Search Methods (20 Points)

Suppose you have the following search space:

State	Next state	Cost
A	B	4
A	C	1
B	D	3
B	E	8
C	C	0
C	D	2
C	F	6
D	C	2
D	E	4
E	G	2
F	G	8

a) [5 pts.] Draw the state space of this problem as a directed graph in which the states are represented as nodes and actions/operators as directed edges, labelled by their cost.

b) [10 pts.] Assume that the initial state is **A** and the goal state is **G**. Show how each of the following search strategies would create a search tree to find a path from the initial state to the goal state:

1. Breadth-first search

2. Depth-first search
3. Uniform cost search
4. Iterative deepening search
5. Bidirectional search

At each step of the search algorithm, show

- which node is being expanded, and
- the content of the **nodes** list

Programming Task: (Optional) Abgabe: Mittwoch, 09.05.07

[20 pts.] Solve the 8-Queen problem (Uninformed Search slides, pages 20-21) using an **uninformed search** algorithm of your choice.

Each run should print the following:

a) The final board configuration of the solution, e.g., written as an ASCII table.
b) The board configurations tested during a given search.

Hints: To represent search trees queues are usually used. Recursion can simplify your algorithm. Beware of infinite looping (i.e., repeated states).