Foundations of Artificial Intelligence

Prof. Dr. W. Burgard, Prof. Dr. B. Nebel, Prof. Dr. M. Riedmiller
J. Aldinger, B. Frank
University of Freiburg
Department of Computer Science
Summer Term 2012

Exercise Sheet 3

Due: Wednesday, June 6, 2012

Exercise 3.1 (Path planning)

Consider the problem of finding the shortest path between two points on a plane that has convex polygonal obstacles (see Fig. 1). This is an idealization of the problem a robot has to solve to navigate its way around in a crowded environment.

(a) Suppose the state space consists of all positions (x, y) in the plane. How many states exist? How many paths are there to the goal?

(b) Explain briefly why the shortest path from one polygon vertex to any other in the scene must consist of (a) straight-line segments joining (b) vertices of the polygons. Define a good state space now. How large is this state space?

(c) How would you implement a successor function that takes a vertex as input and returns the set of vertices that can be reached in a straight line from the given vertex? Give a short explanation of the steps needed in either text or pseudo code.

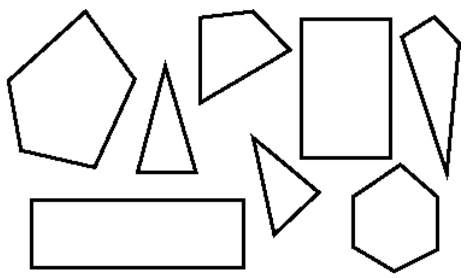


Figure 1: Robot navigation among polygons

Exercise 3.2 (CSP formalization)

Formalize the constraints of the crossword puzzle shown in Figure 2. The variables are given as combination of position and orientation (h: horizontal; v: vertical): $V = \{1h, 1v, 3v, 14h \ldots\}$.
The domains are the set of all possible solution words $dom(v) = \{$EIER, HOLZ, IE, IM, IT, NZ, ON, RAM, RE, ROLLE, ROT, ZAR, ZUHOERER$\}$ for all variables $v \in V$. Specify all unary[1] and binary constraints.

[1] Unary constraints limit the available values of a variable and could be formalized alternatively by a smaller domain.

Hint: A binary constraint is defined as binary relation $C_{uv} \subseteq dom(u) \times dom(v)$ that assignes to each pair of variables $u, v \in V$ its corresponding set of legal assignments.

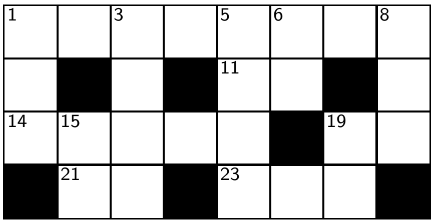
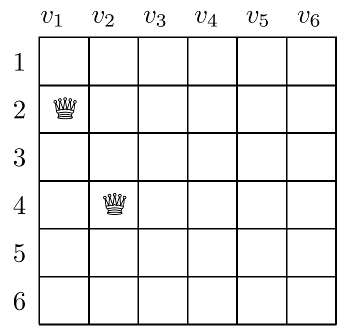


Figure 2: Crossword puzzle

Exercise 3.3 (Forward Checking / Arc consistency)

Consider the 6-queens problem, where 6 pieces have to be placed on a size 6×6 board in such a way that no two queens are on the same horizontal, vertical or diagonal line. Let the domains be $dom(v_i) = 1, \ldots, 6$ for all variables $v_i \in V$. Consider now state $\alpha = \{v_1 \mapsto 2, v_2 \mapsto 4\}$.



(a) Enforce arc consistency in α. Specify in particular the domains of the variables before and after applying arc consistency. You may assume that the domain of variables with allocated values only consists of that value, while the values of unassigned variables still range over the complete domain.

(b) Apply forward-checking in α. Compare with the result of (a).

Exercise 3.4 (Minimax algorithm)

(a) Perform the Minimax algorithm in the tree in Figure 3 using $\alpha\beta$-pruning. Traverse the tree from left to right. Annotate the nodes with their alpha and beta values.

(b) Can the nodes be ordered in such a way that $\alpha\beta$-pruning can cut off more branches? If so, give the order. Otherwise, argue why not.

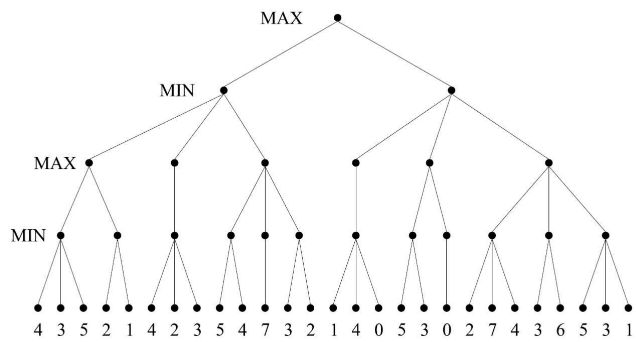
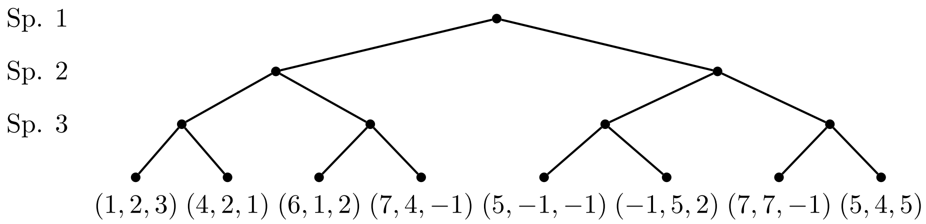


Figure 3: Minimax tree

(c) Consider the problem of search in a three-player game (you may assume that no alliances are allowed) without the zero-sum condition. The players are called 1, 2, and 3. Unlike in the case of two-player zero-sum games, the evaluation function now returns a triple (x_1, x_2, x_3) such that x_i is the value the node has for player i.

Complete the game tree given below by annotating all interior nodes and the root node with the backed-up value triples.



Note: The next exercise session is on **Wednesday, June 13th**. Group 1 will meet in Room 101-00-036, Group 2 will stay in Room 51-00-031 and Group 3 will meet in Room 52-00-016.

The exercise sheets may and should be worked on in groups of three (3) students. Please write all your names and the number of your exercise group on your solution.