**Foundations of Artificial Intelligence**

Prof. Dr. W. Burgard, Prof. Dr. B. Nebel, Prof. Dr. M. Riedmiller
J. Aldinger, J. Boedecker, P. Ruchti
University of Freiburg
Department of Computer Science

Summer Term 2014

# Exercise Sheet 3

**Due: Friday, June 6, 2014**

**Exercise 3.1** (Sudoku)

(a) Show that it is possible to represent *Sudoku* puzzles as graph coloring problems in which some nodes are already initiated with a color. Describe a procedure that transforms a given Sudoku into an equivalent graph coloring problem (give graph nodes and edges, colors and initial colors).

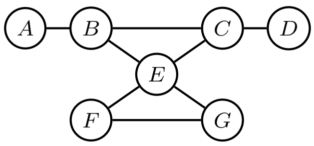
For the game's description, see `http://en.wikipedia.org/wiki/Sudoku`.

(b) Describe how a given *Killer Sudoku* can be formalized as a Constraint Satisfaction Problem.

For the game's description, see `http://www.killersudokuonline.com` or `http://en.wikipedia.org/wiki/Killer_sudoku`.

**Exercise 3.2** (Tree Decomposition)

You want to 3-color the following graph (say with the colors $r$, $g$, $b$).




Show a minimal tree decomposition of the graph and give the sets of all solutions for each of the subproblems. Merge the solutions of the subproblems into an overall solution in the way presented in the lecture. Write down such an overall solution.

**Exercise 3.3** (Forward Checking / Arc consistency)

Consider the 6-queens problem, where 6 pieces have to be placed on a size $6 \times 6$ 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 $\alpha = \{v_1 \mapsto 2, v_2 \mapsto 4\}$.

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $v_5$ | $v_6$ |
|---|---|---|---|---|---|---|
| 1 | | | | | | |
| 2 | ♕ | | | | | |
| 3 | | | | | | |
| 4 | | ♕ | | | | |
| 5 | | | | | | |
| 6 | | | | | | |

(a) Enforce arc consistency in $\alpha$. 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 $\alpha$. Compare with the result of (a).

**Exercise 3.4** (Minimax algorithm)

(a) Perform the Minimax algorithm in the tree in Figure 1 using $\alpha\beta$-pruning. Traverse the tree from left to right. Annotate the nodes with their alpha and beta values.

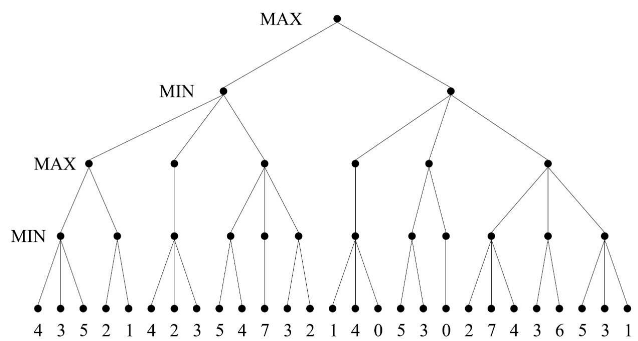
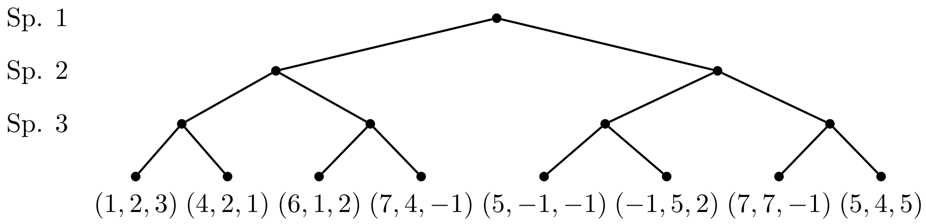



Figure 1: Minimax tree

(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.

(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.




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.