Artificial Intelligence (521279S), Spring 2022
Exercise 3 : Constraint satisfaction and adversarial search
Deadline for reports: Wed 9.2.2022 23:59 (+1h)

This handout contains four problems related to CSP and adversarial search methods. Problems 1-3 are just exercises to support learning and answers to them are provided in **answers3.pdf**. For Problem 4, the answer is not given and you can return an answer to it as a report. So, **include in the report only the answer to Problem 4**. The answer gives max. 1 point, which is taken into account in grading.

Introduction to the topics was provided in Lectures 6 and 8 (course book chapters 5 and 6).

Problem 1. Figure 1 shows a T-crossing, which has traffic lights. The lights A and B are for the vehicles coming from South, C and D for West, and E for North. Consider control of the lights A-E as a constraint satisfaction problem. It is enough to take a look at possible combinations of red and green lights.

(a) Provide definition of the CSP problem. Are there any specific features in the problem?

(b) Draw a constraint graph for the problem.

(c) Using the assignment order (A, B, C, D, E) draw a search tree for the problem. How many allowed traffic light combinations there are?

(d) Propose a better assignment ordering for the variables. In this task, consider the application of the Minimum Remaining Values and Degree Heuristic principles.

(e) Consider the structure of the constraint graph derived in (b) and utilize its nearly tree-structured characteristic for solving the problem.

(f) One target in the control design is to allow as many green lights as possible. Also each light must be green at least in one combination. What are those combinations?

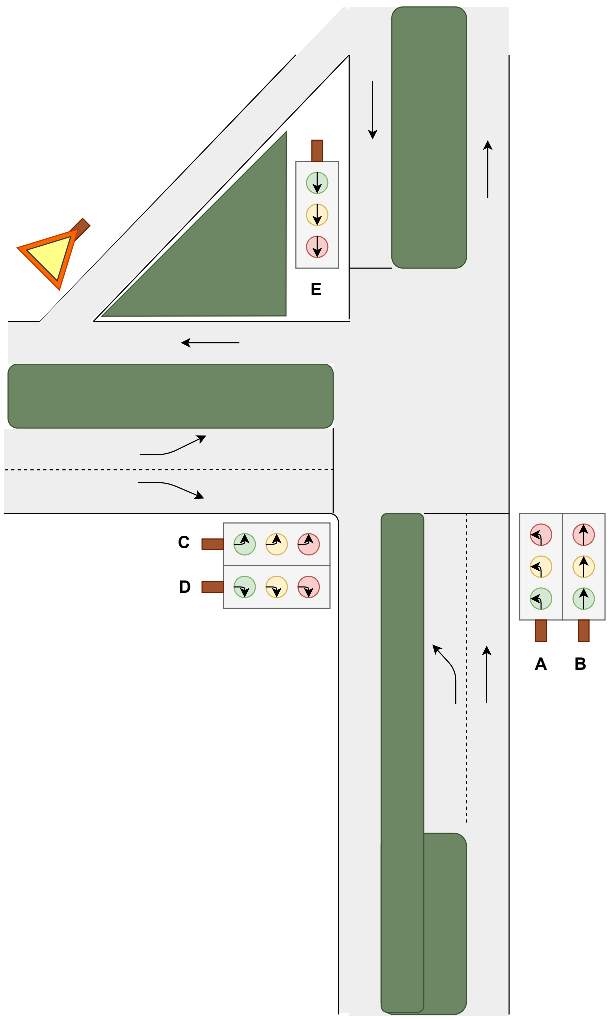

Figure 1: T-junction with traffic lights (Problem 1).

Problem 2. Figure 2 shows a minimax game tree for a situation where the max agent has the next turn. The numbers within the terminal leaf nodes represent the goodness of different end situations.

(a) If the max agent chooses the node C, what will be the goodness of the outcome of the game if both players play optimally?

(b) Show the order in which the tree will be analyzed when alpha-beta pruning is used.

(c) At which leaf node will the game end if both players play correctly from start to finish?

(d) What is the branch chosen at A if MIN nodes B and C are substituted by EXP nodes?

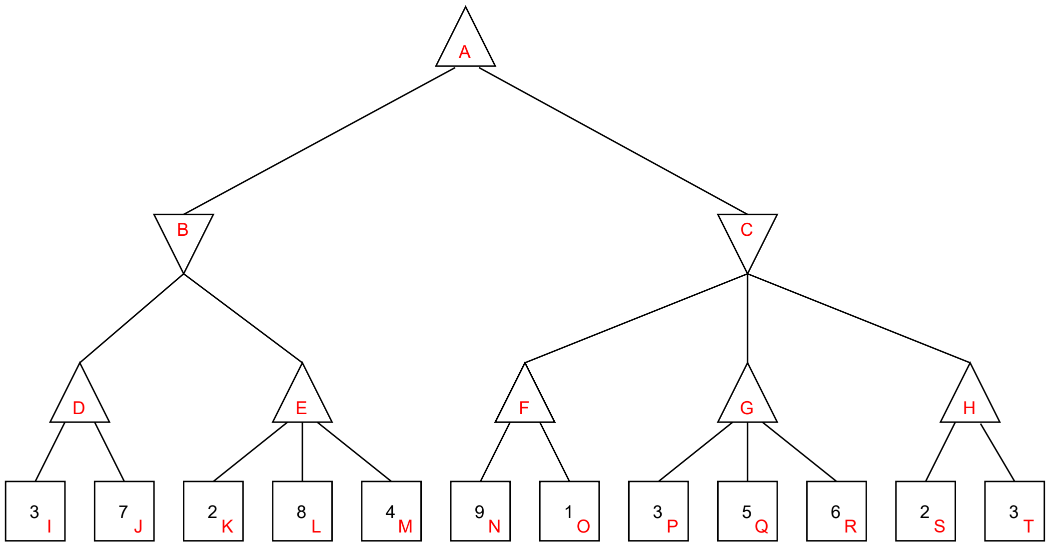

Figure 2: Minimax search tree (Problem 2).

Problem 3. The Matlab program shown in Figure 3 implements a backtracking search for solving the cryptarithmetic problem TWO+TWO=FOUR discussed in lectures. It takes some constraints on variable assignments into account. If you run the program in Matlab[1] , it prints a solution T=7, W=3, O=4, F=1, O=4, U=6, R=8. During the search, it generates 39638 leaf nodes, that is, it checks 39638 complete assignments to variables until it finds the solution. Modify the program to reduce this count.

```
function crypta
  C1s = [0,1];
  C2s = [0,1];
  Fs = [1]; % Because F cannot be zero and it is actually carry C3, it must be equal to 1
  Rs = [0,2,4,6,8];
  Os = [1,2,3,4,5,6,7,8,9];
  Us = [0,1,2,3,4,5,6,7,8,9];
  Ws = [0,1,2,3,4,5,6,7,8,9];
  Ts = [0,1,2,3,4,5,6,7,8,9];
  count = 0;
  for F = Fs
    for C1 = C1s
      for C2 = C2s
        for R = Rs
          for O = Os
            for U = Us
              for W = Ws
                for T = Ts
                  count = count+1;
                  if goal_test(T,W,O,F,U,R,C1,C2)
                    fprintf('T=%d,W=%d,O=%d,F=%d,O=%d,U=%d,R=%d\n',T,W,O,F,O,U,R);
                    fprintf('%d leaf nodes visited.\n',count);
                    return;
                  end
                end
              end
            end
          end
        end
      end
    end
  end
end

function ok = goal_test(T,W,O,F,U,R,C1,C2)
  ok = ((2 * O) == (R + 10 * C1)) ... % arithmetic constraints
    && ((C1 + 2 * W) == (U + 10 * C2)) ...
    && ((C2 + 2 * T) == (O + 10 * F)) ...
    && (T ~= W) && (T ~= O) && (T ~= F) && (T ~= U) && (T ~= R) ... % all different
    && (W ~= O) && (W ~= F) && (W ~= U) && (W ~= R) ...
    && (O ~= F) && (O ~= U) && (O ~= R) ...
    && (F ~= U) && (F ~= R) ...
    && (U ~= R);
end
```

Figure 3: Cryptarithmetic problem solver (available as **crypta.m** in Moodle).

[1] If you want to use Matlab note that university has a campus license for personnel and students. See https:// se.mathworks.com/academia/tah-portal/university-of-oulu-873976.html).

***Problem 4 [1p]**. Consider the following set of equalities and inequalities between five variables A, B, C, D, and E.

$$A + B = C \tag{1}$$

$$B + C + E < 8 \tag{2}$$

$$D - A = 2 \tag{3}$$

$$B + D > 4 \tag{4}$$

Each variable can take a value from the set $\{1,2,3,4,5,6,7,8\}$. Apply techniques of CSP to solving this problem.

(a) Draw a constraint graph for this problem.

(b) Perform constraint propagation in order to infer the sets of possible values for each variable. Try to find maximal reduction to the sizes of these sets. The table below could be used to represent the result (mark with x impossible values).

	1	2	3	4	5	6	7	8
A								
B								
C								
D								
E								

(c) Using the result of (b) as a starting point, perform backtracking search to find a solution. Apply at least the minimum remaining values principle to organize the search. In the report, draw a search tree which shows all the nodes generated during the search.

(d) Consider the constraint graph you determined in (a). Propose a minimal cutset for making the problem tree-structured. Determine a solution to the problem using cutset conditioning.