Intelligent Systems Programming (ISP)
BSc and MSc Exam

IT University of Copenhagen

24 May 2019

This exam consists of 9 numbered pages with 3 problems. Each problem is marked with the weight in percent it is given in the evaluation. **Notice that the first version of problem 3 only is to be solved by Bachelor (BSc) students, while the second version only is to be solved by Master (MSc) students**.

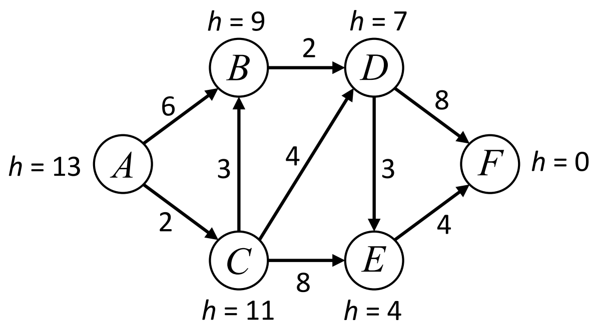
Use notations and methods that have been discussed in the course.

Make appropriate assumptions when a problem has been incompletely specified.

Good luck!

Problem 1: Heuristic Search (25%)

The state-space graph shown below represents a heuristic search problem. The vertices A, B, C, D, E, and F denote states and the edges denote actions. The step cost of each action is written next to its edge. For example, the step cost of taking the action from C to B is 3. The initial state is A and there is only one goal state F. The value of a heuristic function (h) for this search problem is written next to each state.



a) Argue that h is a valid and admissible heuristic function.

b) Argue that h also is a consistent heuristic function (hint: there is a simple reason for this).

c) Draw the search tree grown by A* graph search to solve the search problem. Write the final g and f-values associated with each node in the tree when the algorithm terminates.

Problem 2: Propositional Logic and BDDs (40%)

Prof. Smart is planning a weekend trip with his wife. There are four sights that they consider seeing, namely X, Y, Z and W. The rest of the time, they will just enjoy the atmosphere of the city. Smart's wife makes the following statements about their trip.

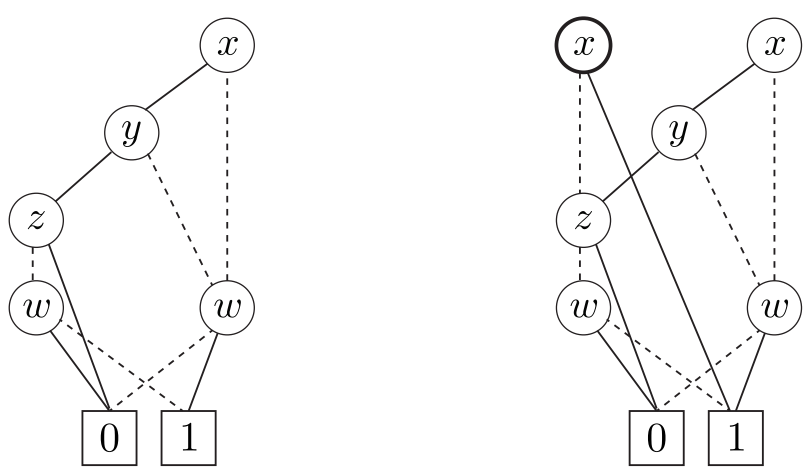
1. "I'm fine whether we see X or not, but if we choose to see both X and Y, we don't have time for neither Z nor W."

2. "However, if we leave out either X or Y (or both), we do have time for W and then we should definitely go see it."

a) Write logical expressions for the statements (1) and (2) above.

- Use the variables x, y, z and w to write the two statements. The variable x should denote whether Prof. Smart and his wife will see X, i.e. x is true iff they go see X. Similarly for the other variables.
- Translate the expressions to CNF.

Prof. Smart makes a BDD corresponding to the conjunction of the two requirements of Mrs. Smart using the the variable ordering $x \prec y \prec z \prec w$, see Figure 1a. After thinking for a while, he then makes another BDD representing his own requirement - an implication. This results in the multi-rooted BDD in Figure 1b. That is, Prof. Smart's requirement is represented by the BDD with the root that has a bolder outline in Figure 1b.



(a) Conjunction of (1) and (2). (b) Multi-rooted BDD.

Figure 1: BDDs

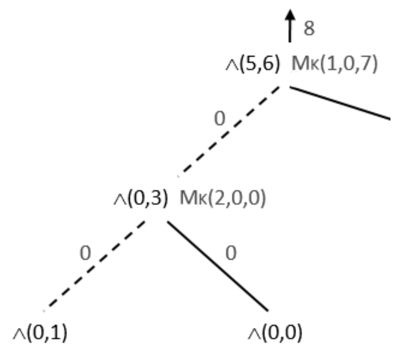


Figure 2: Call tree of Apply of unrelated BDDs.

b) What does Prof. Smart's requirement correspond to?

- Write a logical expression that is equivalent to his requirement. You can use the if-then-else operator if you want to, but it is not required.
- Write the expression on a form, where the main connective is an implication.
- Further, write a sentence in English natural language that this requirement could correspond to.

c) Write the unique table T corresponding to the multi-rooted BDD in Figure 1b. You should assume that the variables are represented by their number in the ordering. Thus the first variable (x) has variable index 1 while the second variable (y) has variable index 2 and so on.

- Make sure that the nodes are numbered as they would when made through a number of calls to Mk.
- Copy the BDD from Figure 1b next to T on the same page and write the IDs next to the nodes in the figure.

Now Prof. Smart wants to see what happens when both his own and his wife's requirements are taken into account. That is, he wants to make the conjunction of the two BDDs with roots in the multi-rooted BDD.

d) Use APPLY to make the conjunction ($\wedge$) of the BDDs corresponding to the two nodes testing on x in Figure 1b.

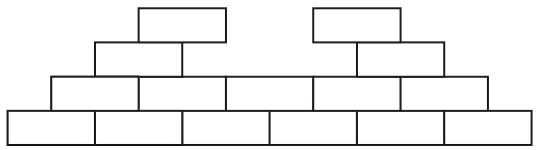
- Write the call tree. When doing this, you have to use the following guidelines. For each call to APP, you have to write the two arguments (two node identifiers), and you have to write which call to MK is made. On the arcs, write what the method call returns. That is, you follow the notation in the slides giving an example of APPLY, part of which can be seen in Figure 2.
- You also need to have a table showing G, and you have to underline the nodes of the call tree in which there is a cache hit. The call tree, G, and T has to be on the same page. If necessary, turn your paper to "landscape" mode.
- Draw the resulting BDD.

Looking at his result, Prof. Smart exclaims: "Oh, then it is not possible for us to see both Z and W!"

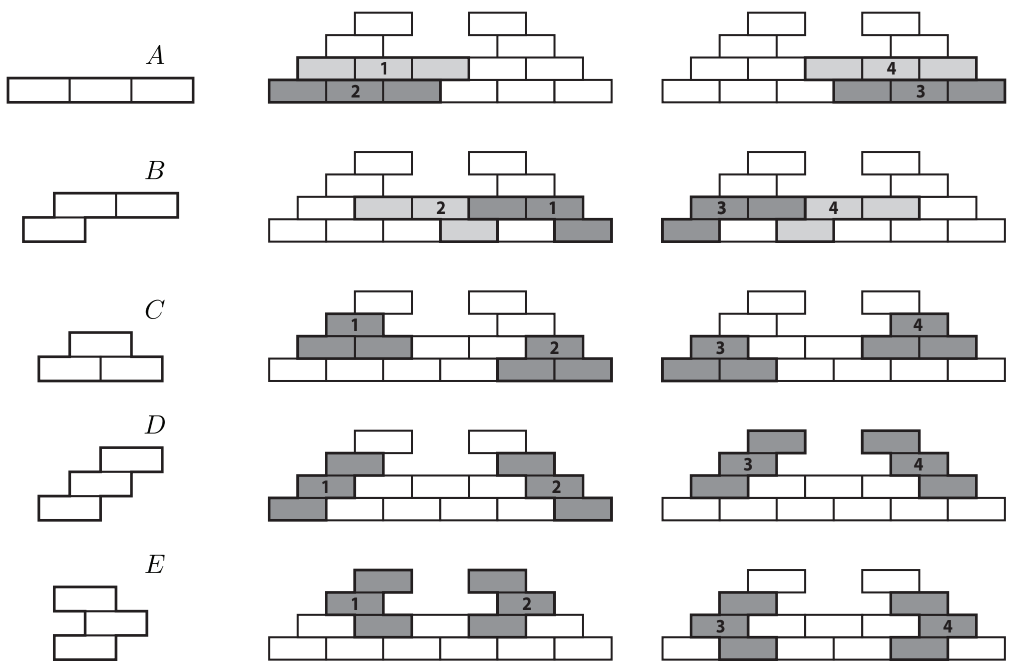
e) Is Prof. Smart right? Give an explanation why/why not.

Problem 3: FOR BSC STUDENTS ONLY (35%)

In the *Brick by Brick* puzzle, the goal is to use each of five different blocks to build the wall shown below. Each block must be used exactly one time.



In a CSP formulation of the problem, we use five variables A, B, C, D, and E to represent the position of each block. Each variable has domain $\{1, 2, 3, 4\}$ corresponding to four different positions of the block in the wall. It is allowed to flip the blocks horizontally and vertically. The only constraint is that the blocks cannot overlap each other. The blocks and the possible positions that we model are defined in the figure shown below.



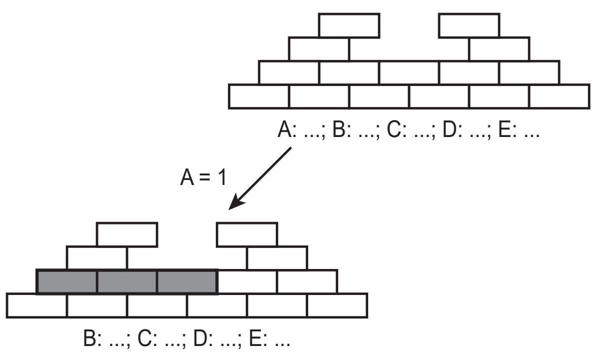
a) Is the CSP formulation complete in the sense that all configurations of the blocks can be represented (why/why not)?

b) Draw the constraint graph of the CSP.

c) Assume that $A \in \{1, 2, 3, 4\}$ and $B \in \{1, 2\}$. What are the $A \to B$ arc-consistent domains of A and B?

d) Recall that the initial domain of all variables in the CSP is $\{1, 2, 3, 4\}$. Why is an arc consistency algorithm like AC-3 unable to remove any domain values from this CSP?

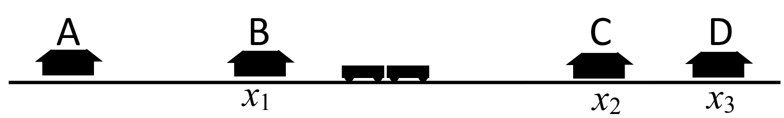
We will use Forward-Checking-Search to try to solve the puzzle. We use the MRV heuristic to select an unassigned variable, where ties are broken by choosing the variable that comes first in the alphabet. Domain values are tried in ascending order. Each call to Recursive-Forward-Checking is drawn as a puzzle state given by assigned variables, with the domains of unassigned variables written under it. The initial call and a child call corresponding to the assignment $A = 1$ are illustrated below (notice that the domains have not been written in the example, but you will have to in your answer).

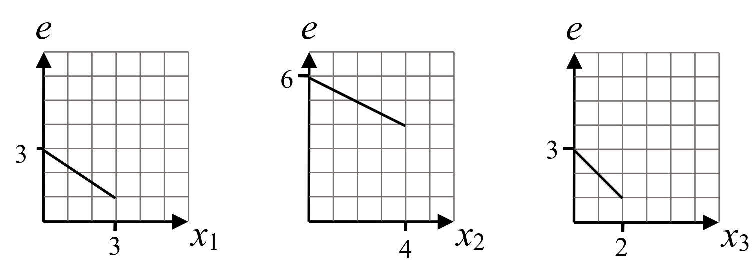


e) Draw the complete call tree of Forward-Checking-Search using the approach described above. Indicate a *failure* result with an F in the tree. We recommend that you use squared paper in "landscape" orientation.

Problem 3: FOR MSC STUDENTS ONLY (35%)

A metro train drives between four stations A, B, C, and D. The management wants to minimize the total amount of energy that the train uses to drive between the stations. It is currently 12 units, but can be reduced by delaying the arrival time at station B, C, and D. Let x_1, x_2, and x_3 denote the arrival delay in minutes at each of these three stations. The graphs below shows the energy e needed to drive from station A to B, B to C, and C to D as a function of x_1, x_2, and x_3, respectively. Thus, if the management decides the delays $x_1 = 0$, $x_2 = 2$, and $x_3 = 1$, the total needed energy is reduced from 12 to $3 + 5 + 2 = 10$ units.





The delays have to fulfil the following rules:

- The sum of all delays is at most 4 minutes.
- The maximum delay at station B, C, and D is 3, 4, and 2 minutes, respectively.

a) Write a linear program (LP) in standard form that can be used to solve the management's optimization problem. You can assume that the objective is to maximize the amount of saved energy. The only decision variables should be x_1, x_2, and x_3.

b) Write the slack form of the LP. Is the slack form a feasible initial dictionary for the SIMPLEX algorithm (why/why not)?

c) Solve the problem using the SIMPLEX algorithm. Write all dictionaries produced by the algorithm.

d) Based on your answer to c), what is the minimum energy needed by the train and what are the corresponding delays at station B, C, and D?

e) Based on your answer to c), what is the minimum energy needed by the train, if the total allowed delay is increased from 4 to $4\frac{1}{2}$ minutes, but the maximum delay in station D is reduced from 2 to $1\frac{1}{2}$ minutes?