Introduction to Artificial Intelligence (IAI) BSc and MSc Exam

IT University of Copenhagen

2 June 2022

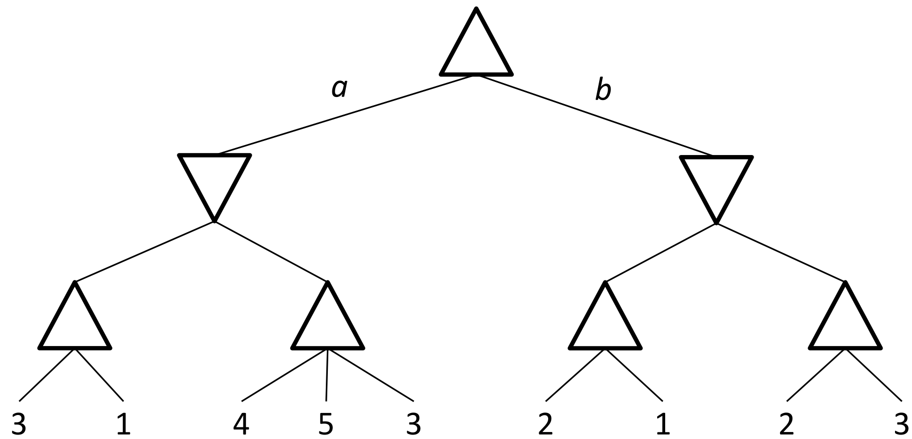
This exam consists of 7 numbered pages with 4 problems. Each problem is marked with the weight in percent it is given in the evaluation. **Notice that the first version of problem 4 only is to be solved by Bachelor (BSc) students, while the second version only is to be solved by Master (MSc) students**.

- 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: Adversarial Search (20%)

Consider the game tree of a two player zero-sum game shown below. As usual, the numbers at the bottom of the tree represent terminal states and are the utilization of MAX.



a) Write the minimax value of each MAX and MIN node in the tree. Mark the minimax decision of MAX in the initial state by highlighting the relevant outgoing edge.

b) Indicate the possible alpha and beta cuts in the game tree. Assume that child nodes are evaluated left to right.

c) Is it possible to prune a larger number of nodes in the tree if child nodes are evaluated using another ordering strategy than left to right (why/why not)?

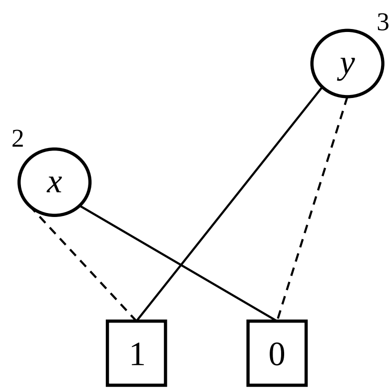
Problem 2: Logic and ROBDDs (25%)

a) Draw the ROBDD of $(x \Rightarrow y) \wedge \neg x$ using variable ordering $y \prec x$.

b) Write the simplest Boolean expression that $(x \Rightarrow y) \wedge \neg x$ is logically equivalent to. Explain your answer.

c) Does $(x \Rightarrow y) \wedge \neg x \models \neg y$ hold (why/why not)?

A unique table contains the multi-rooted ROBDD shown below. The variable ordering is $y \prec x$.

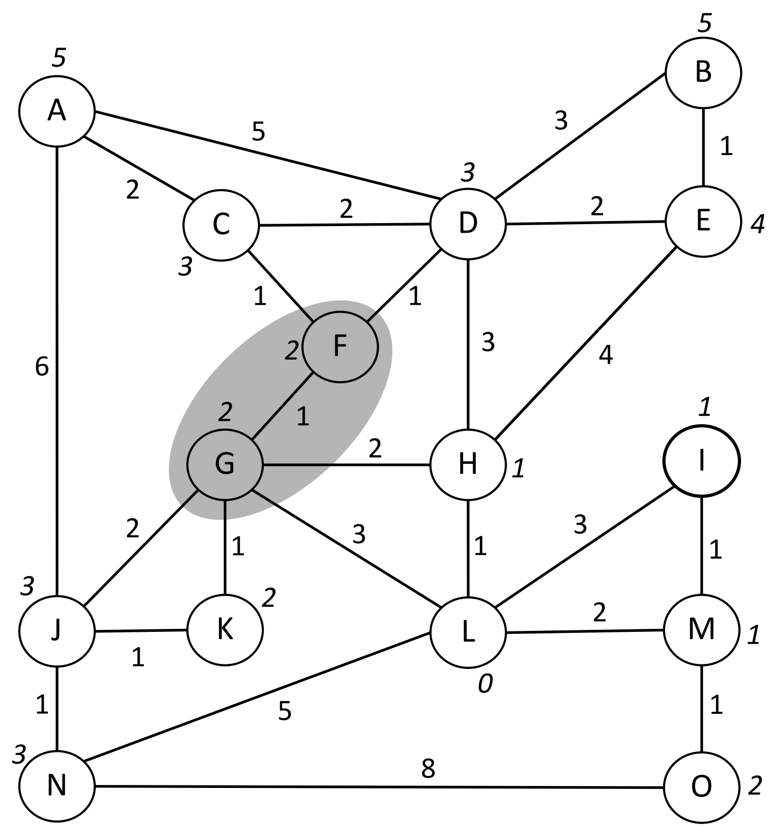


d) Use APPLY($\wedge$,2,3) to make the conjunction of the two ROBDDs in the multi-rooted ROBDD with identifiers 2 and 3. Draw the call tree of APP as in the APPLY example of Lecture 7. Indicate cache hits and draw the resulting multi-rooted ROBDD.

e) What is the relation between the ROBDD that you have computed using APPLY in the answer to question d) above and $(x \Rightarrow y) \wedge \neg x \models \neg y$?

Problem 3: Heuristic Search (40%)

Professor Smart has moved to a new city and plans to commute to work by car. He wants to use the A^* algorithm to find a cheapest route from his home to his work place. He starts by drawing the city map shown below. Each node in the map is an intersection and each line is a street. Note that the map is an abstraction. The real streets may not be straight lines. You can ignore the shaded ellipse covering node F and G for now.



Professor Smart lives at intersection C and his work place is at intersection L. Next to each street, he writes the cost of driving the street with his car. There are no one-way streets and the cost is the same in each direction.

As heuristic function, $h(n)$, Professor Smart chooses the cost of driving the straight line distance from intersection n to intersection L. He writes the value of this heuristic next to each intersection.

You can assume that the cost of driving Professor smart's car only depends on the distance it drives. Hence, a street that is two miles long will have twice the cost as one that is only one mile long.

a) Is $h(n)$ admissible (why/why not)?

b) Is $h(n)$ also consistent (why/why not)?

Professor Smart uses the Best-First-Search algorithm shown in Figure 3.7 of RN21 (also on page 34 of the slides for lecture 2) with $f(n) = g(n) + h(n)$ as required for A^*.

c) Write the content of the *frontier* queue and the *reached* lookup table at the end of each iteration of the algorithm. You can write your answer as a finished version of the table shown below. What is the solution returned by the algorithm?

Iter.	*frontier*	*reached*
0	$(f = 3, C, g = 0, h = 3)$	$(C, g = 0)$
1	...	...

Professor Smart is playing with weighted A^* with $f(n) = (1-w) \times g(n) + w \times h(n)$, where $0 \leq w \leq 1$.

d) Describe at least one advantage and one disadvantage of choosing w strictly larger than 0.5.

The city's mayor decides to introduce road pricing in the inner city that contains intersection G and F. It works as follows. When driving a street that enters the road pricing area (shaded ellipse in the city map), a fee of 2 is paid. The fee is only paid one time per day. Thus, the cost of driving the street from C to F increases to 3, when the road pricing fee must be paid.

e) Does the road pricing change the cheapest path from Professor Smart's home to his work (why/why not)?

According to Professor Smart, road pricing makes it impossible to use the A^* algorithm to find a cheapest route from his home to his work. The reason is that heuristic search problems require constant action costs, but when road pricing is introduced a street will have different cost depending on whether the road price fee already has been paid or not.

f) Is Professor Smart right or not? Carefully argue for your answer.

Problem 4: FOR BSC STUDENTS ONLY (15%)

Write whether you agree or disagree with each of the five statements below. Briefly explain your answers.

a) A local search algorithm assigns a single variable in each iteration.

b) WalkSat is an incomplete algorithm.

c) If Bactracking-Search is extended to find all solutions then the ordering of domain values by Order-Domain-Values is irrelevant.

d) If AC-3(*csp*) returns *true* then *csp* has at least one solution.

e) A^* traverses a search tree.

Problem 4: FOR MSC STUDENTS ONLY (15%)

Write whether you agree or disagree with each of the five statements below. Briefly explain your answers.

a) A local search algorithm assigns a single variable in each iteration.

b) If BACTRACKING-SEARCH is extended to find all solutions then the ordering of domain values by ORDER-DOMAIN-VALUES is irrelevant.

c) If AC-3(csp) returns *true* then csp has at least one solution.

d) If an LP has an optimal dual solution then it also has an optimal primal solution.

e) The average run-time of SIMPLEX is linear in the number of decision variables.