Written examination
TIN174/DIT410, Artificial Intelligence

Thursday 8 June 2017, 14:00–18:00

Examiner: Peter Ljunglöf (tel: 772 1065)

Solution suggestions

This examination consists of six questions divided into twelve subquestions.
A correctly answered subquestion gives you one point, the total number of points is 12.

Grades: To get grade 3/G/pass you need at least 66% correct, i.e., 8 points.

This is only for students from previous years:

To get Chalmers grade 4 you need at least 10 points.
To get GU grade VG/distinction you need at least 11 points.
To get Chalmers grade 5 you need all 12 points.

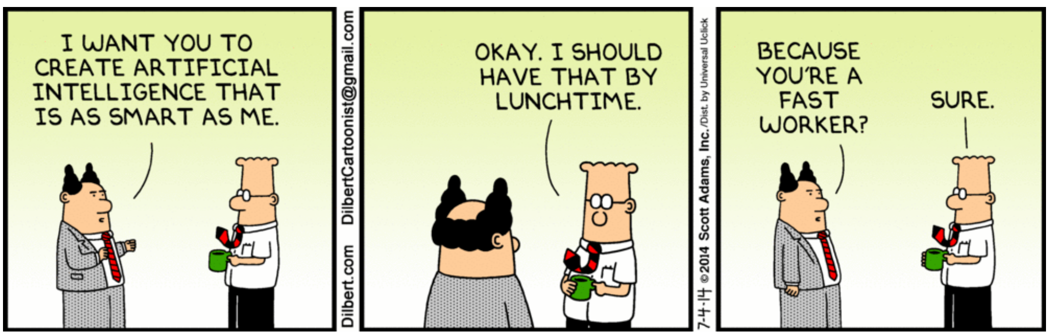
Tools: Paper and pencil.

No extra books, papers or calculators.

Notes: Answer every question directly on the question paper, and write your ID number at the top of every paper.

If you have any extra papers with associated calculations, you should hand in them too.

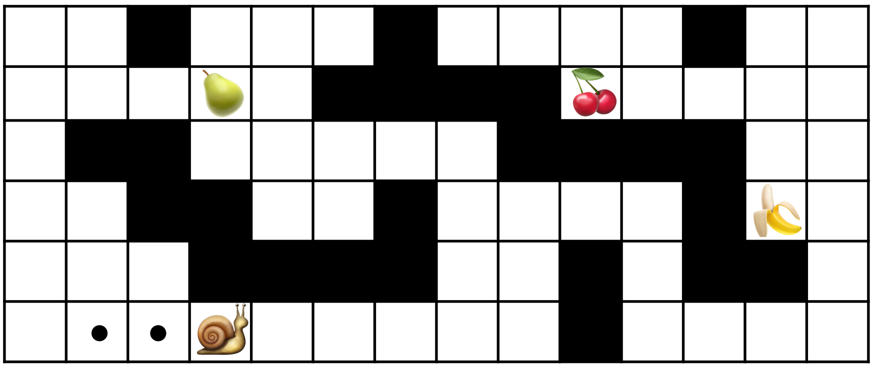
Write legibly, and explain your answers!



http://dilbert.com/strip/2014-07-04

1. A very hungry snail with a very sticky trail [2p]

A snail is hungry and wants to eat the three fruits as soon as possible. However, this particular snail leaves a very sticky trail – if it walks over its own trail it gets stuck. This stickyness only lasts for two turns, which means that the last two squares that the snail visited are sticky and should not be walked over. In the figure below these two squares are denoted by the two dots left of the snail (which means that the snail has been moving right the last two turns). At each turn the snail moves exactly one square up, down, left or right, and it cannot enter (or climb over) any walls (i.e., the black squares), but it has GPS and a map where the walls and the fruits are shown. When it reaches a fruit it eats it immediately (the snail is very hungry).



a) Give a suitable representation of the states in this search problem.
Don't forget to specify the domain of each state variable.
You can assume that the maze has size $w \times h$ squares.

correct?

A tuple $(x,y, x_1,y_1, x_2,y_2, f_1,f_2,f_3)$, where $1 \leq x_i \leq w$, $1 \leq y_i \leq h$, and f_i are booleans.

(x,y), (x_1,y_1) and (x_2,y_2) are the three latest positions of the snail, and f_i states whether fruit i has been eaten or not.

Alternative solution: $(x,y, a_1,a_2, f_1,f_2,f_3)$, where $a_i \in$ {n,s,w,e} are the two latest actions that the snail performed.

b) Give a nontrivial admissible heuristics.
(I.e., the heuristics should not be constant)

correct?

Of course there are several solutions, but here are two possibilities:

- the Manhattan distance from the snail to the closest fruit
- the Manhattan distance from the snail to the fruit the farthest away

2. Generic heuristic search [4p]

Assume we define an evaluation function f for a heuristic search problem as:

$$f(x) = u \cdot g(x) + w \cdot h(x) \qquad [u, w \geq 0]$$

where $g(x)$ is the cost of the best path found from the start state to state x,
and $h(x)$ is a consistent heuristic function that estimates the cost of a path from x to a goal state,

a) What search algorithm do you get when $u = 0$ and $w = 1$?

correct?

Greedy best-first search

b) What search algorithm do you get when $u = 1$ and $w = 0$?

correct?

Uniform-cost search

c) What search algorithm do you get when $u = w = {}^1/_2$?

correct?

A* search
(note that you get A* search as long as $u = w$, regardless of their value)

d) For which values of u and w will the search algorithm always find the optimal solution?

correct?

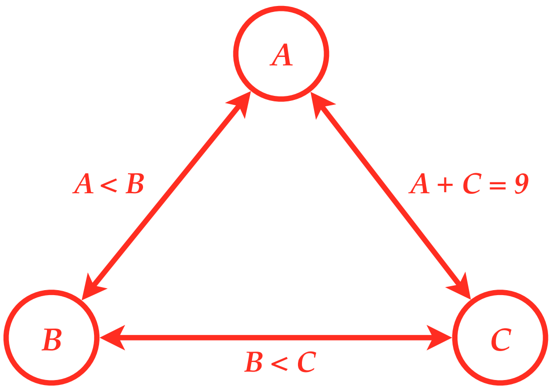
For all values such that $0 \leq w \leq u$
(also, u has to be strictly positive, but that's a minor detail)

3. Constraint satisfaction [2p]

Assume that you have three variables, A, B and C, all with domains {1, 2, 3, 4, 5, 6}.
The constraints on the values are that $A < B$, $B < C$ and $A + C = 9$.

a) Draw the constraint graph and write the constraints next to the edges.

correct?



3. Constraint satisfaction (continued)

(Recall that the constraints are $A < B$, $B < C$ and $A + C = 9$, and the domains {1,2,3,4,5,6}).

b) Perform arc-consistency on the problem, using the table below.
Use **A→B**, **B→C**, **A→C**, **B→A**, etc., to denote edges.

correct?

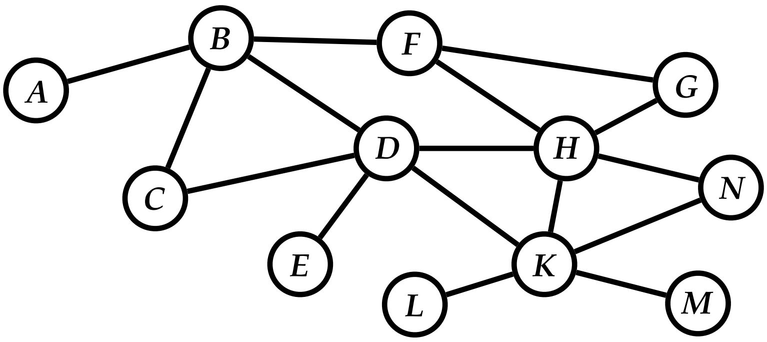
There are several orderings of the edges in the queue – this is just one possibility:

Edge removed from queue	*Constraint to be checked*	*Affected variable*	*Values removed from variable domain (if any)*	*Edge(s) added to queue (if any)*
A → B	A < B	A	6	(C → A)
B → A	A < B	B	1	(C → B)
B ~~A~~ → ~~B~~ C	B < C	B	6	A → B
C → B	B < C	C	1, 2	(A → C)
C → A	A + C = 9	C	3	B → C
A → C	A + C = 9	A	1, 2	B → A
A → B	A < B	A	5	C → A
B → C	B < C	B		
B → A	A < B	B	2, 3	C → B
C → A	A + C = 9	C	4	B → C
C → B	B < C	C		
B → C	B < C	B		

Variable	*Final domain*
A	3, 4
B	4, 5
C	5, 6

4. An almost tree-structured CSP [1p]

Assume the following complex constraint graph with 12 variables and 16 binary constraints.

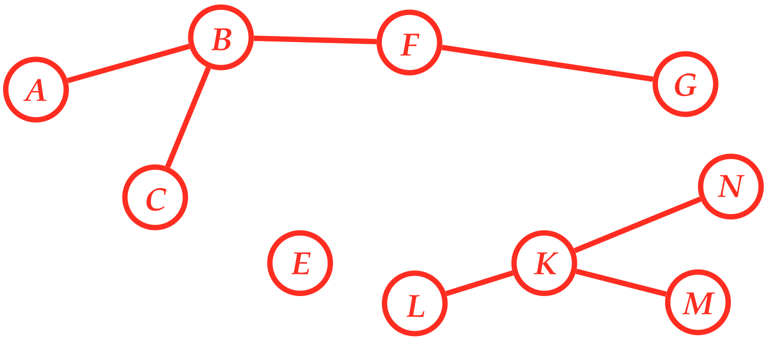


a) Find the smallest *cycle cutset** for the constraint graph, and draw the resulting constraint graph, after removing the cycle cutset from the original graph.

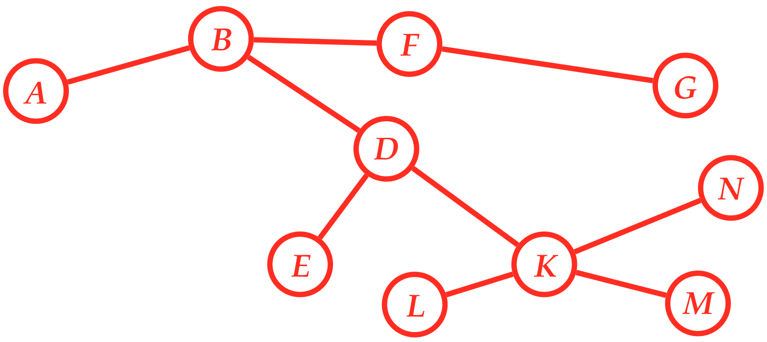
correct?

There are two possible smallest cycle cutsets:

- removing {*D*, *H*} results in the graph:



- removing {*C*, *H*} results in the graph:



* The *cycle cutset* is the smallest set of variables such that removing those from the constraint graph makes the resulting graph tree-structured.

5. A stochastic game tree [1p]

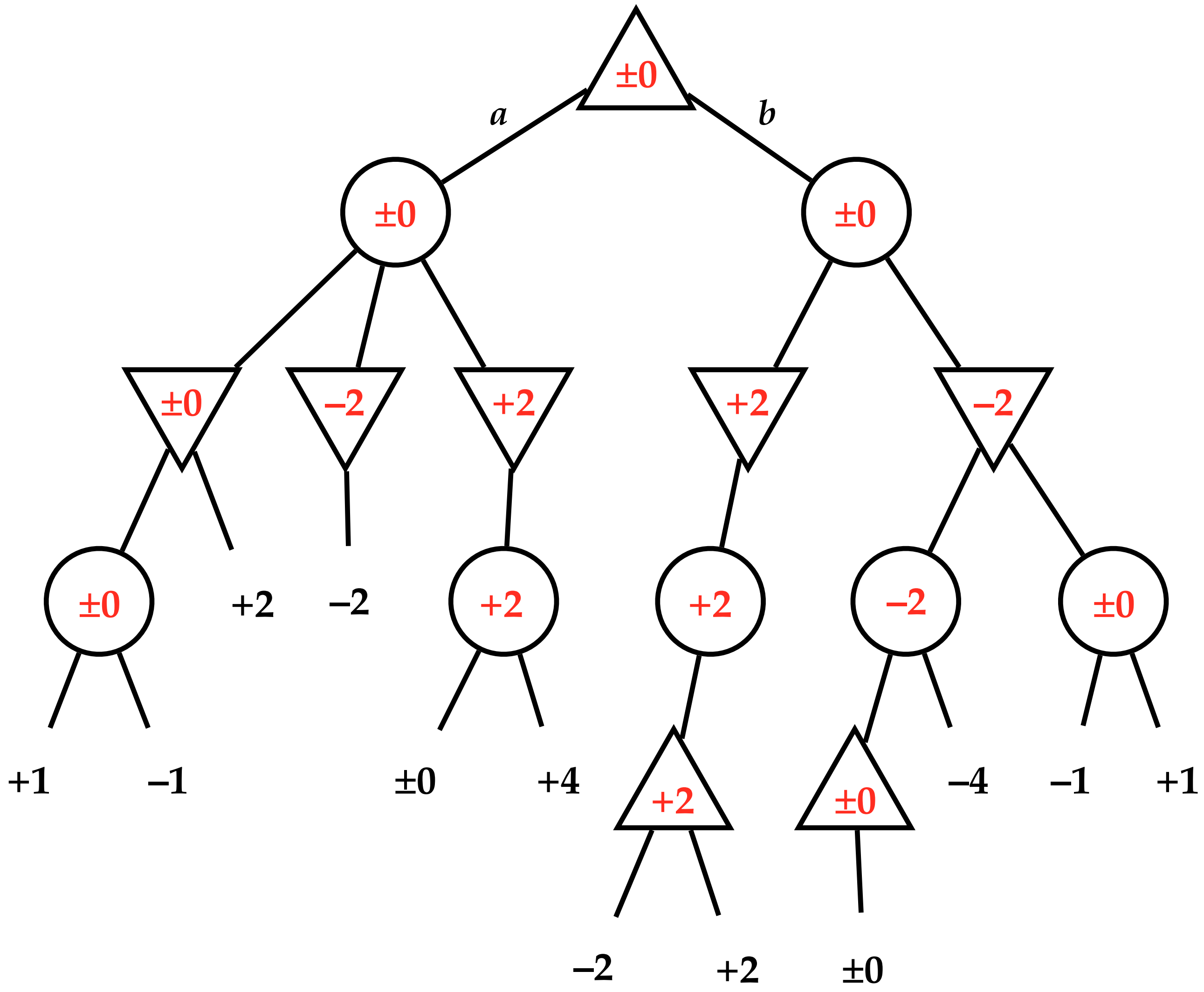
Assume the stochastic minimax game tree below, where ◯ are chance nodes, △ are maximising nodes, and ▽ are minimising nodes. Furthermore, assume that every chance node has a uniform probability for its actions to occur.

a) Perform the *expecti-minimax* algorithm on the game tree, and write the resulting values inside the empty nodes.

correct?

Which next move is the best move for the maximising player, according to the expecti-minimax algorithm?

☐ *a* ☐ *b* ✅ *they are equally good/bad*



6. Nim, or the subtraction game [2p]

Here is a very simplified variant of the game of Nim, called the subtraction game:

The game starts with N stones on a board. The two players, Anna and Bengt, alternate by removing *at most* k stones from the board ($0 < k < N$), i.e., if $k = 3$, then a player can remove either 1 or 2 or 3 stones at each turn. Anna makes the first move, and the player who clears the board wins, i.e., the one who makes the final move wins.

When you draw game tree for this game, use △ for Anna's and ▽ for Bengt's nodes.
Use the utility +1 for the leaf nodes where Anna wins, and –1 where Bengt wins.

a) Draw a game tree below for the case $N = 3$ and $k = 2$.
Write 1 or 2 at each edge, to show how many stones the player removes.

correct?

b) Perform the minimax algorithm on the game tree.
Write the minimax values inside Anna's and Bengt's nodes.
Which next move is the best first move for Anna, according to minimax?

correct?

☐ *1* ☐ *2* ✅ *they are equally good/bad*