Lab Class S:III

By November 26th 2014, solutions for the following exercises have to be submitted: 1, 2, 3, 4, 7.

Exercise 1 : Matchsticks

The matchsticks game. 7 matchsticks are placed next to each other on a table. Two players take turns removing 1, 2 or 3 matchsticks. The player who removes the last matchstick loses.

(a) Which problem representation is better suited here—state-space graph (OR graph), or problem-reduction graph (AND-OR graph)?

(b) What is a suitable encoding?

(c) With respect to your encoding, what characterizes a solved leaf node?

(d) In the resulting graph, when can a node be seen as solved?

(e) How could you show that the player who takes the first turn always wins, or always loses? Note: you are not required to produce the proof for the matchstick game, but to describe how this could be proven.

(f) Write a program that takes as input (1) the number of matchsticks on the table and (2) a list with the numbers of matchsticks that a player can choose to remove in each turn. That is, for the matchsticks game as stated above, your inputs would be `7` and `[1, 2, 3]`. Your program should output a winning strategy (i.e., the sequence of moves and counter-moves) for the player who takes the first turn, if such a strategy exists. For example, the output of your program could look like the following:

```
14 matchsticks; players can take 1, 2 or 3
move 1:  take 1 (13 left)
   move 2:  if opponent takes 3 -> take 1 (9 left)
      move 3:  if opponent takes 3 -> take 1 (5 left)
         move 4:  if opponent takes 3 -> take 1 (You win.)
         move 4:  if opponent takes 2 -> take 2 (You win.)
         move 4:  if opponent takes 1 -> take 3 (You win.)
      move 3:  if opponent takes 2 -> take 2 (5 left)
         move 4:  if opponent takes 3 -> take 1 (You win.)
         (...)
   move 2:  if opponent takes 2 -> take 2 (9 left)
   (...)
```

Exercise 2 : Serializability

Is the 8-Queens problem serializable? Explain your answer.

Exercise 3 : Depth-first search in a graph

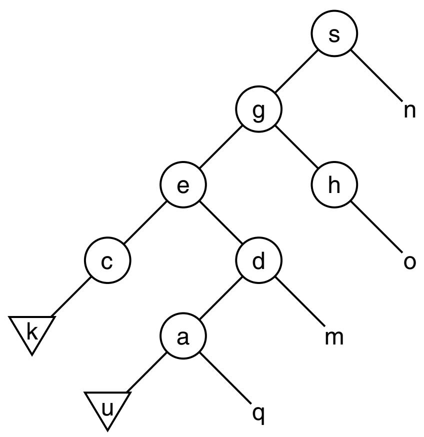
(a) How can we avoid expanding a node several times in depth-first search?

(b) How can cycles be detected by a depth-first search? Will your procedure detect all cycles in the graph?

(c) After a long night you awake in a maze. How can you find the exit? Describe your assumptions about the maze structure and any tools you require.

Exercise 4 : Search space

Given the following situation:



The letters without circle or triangle represent nodes in OPEN, letters in circles represent nodes in CLOSED and letters in triangles represent dead end nodes (which are also in CLOSED). Which nodes are removed from CLOSED by the method `cleanup_closed`?

Exercise 5 : Depth-first search in AND-OR trees

You are searching in a complete binary tree T of depth d with root r. The root is an OR node; AND nodes and OR nodes alternate along all paths in T.

(a) Describe the structure of all solution graphs.

(b) How many nodes does each solution have?

(c) How can you determine whether T contains a solution?

Exercise 6 : Multiple choice

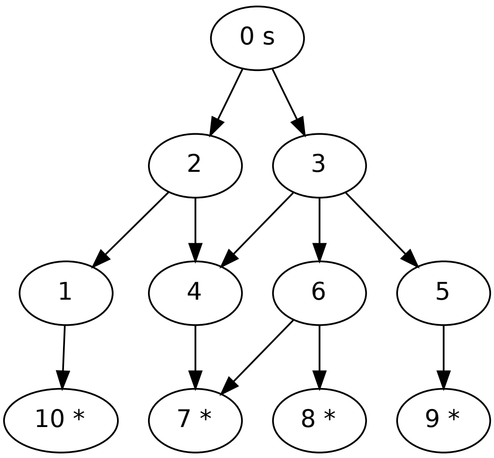
Which characteristic(s) of a systematic control strategy is/are violated by Hill-Climbing?

- ☐ Efficiency
- ☐ Completeness
- ☐ Optimality

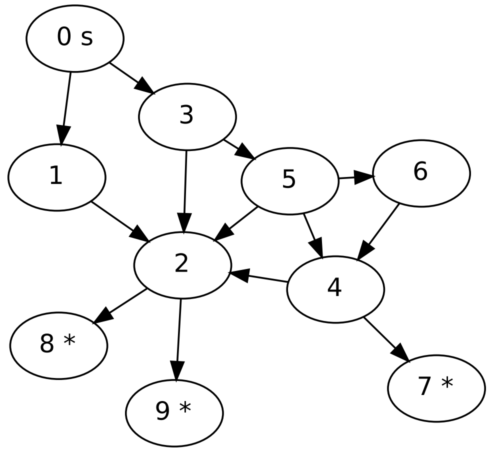
Exercise 7 : Hill-climbing

Apply the hill-climbing algorithm HC given in the lecture notes to the following graphs. In each case, write down (1) the return value of HC, and (2) all values that the variable $n_{\textsf{opt}}$ takes on throughout the execution of the algorithm. For the purposes of this exercise, the input to algorithm HC is the following:

- The node marked with s is the start node.
- The function *successors*(n) returns all nodes n' for which an edge (n, n') exists in the graph.
- The predicate $\star(n)$ is *True* if and only if node n is marked with an asterisk.
- The merit function $f(n)$ simply returns the node number of node n.



(a)



(b)