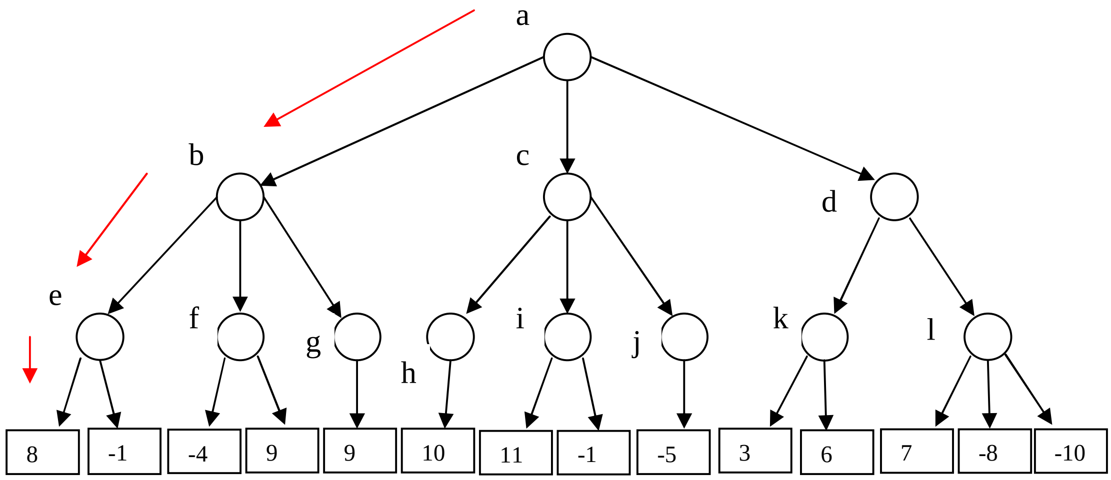
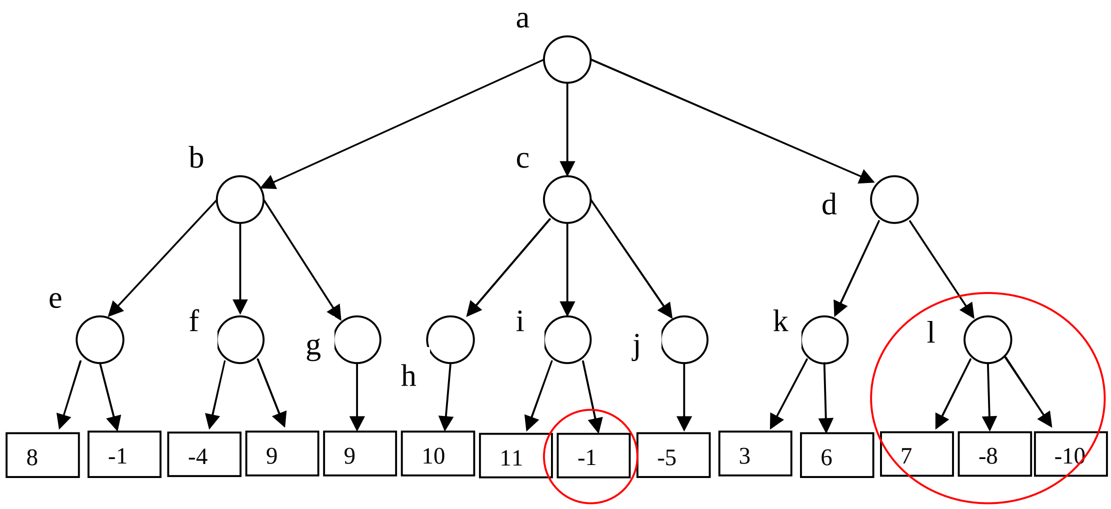
# Cooperating Intelligent Systems – Written Exam October 2009 – Suggested solution

## 1 Game playing

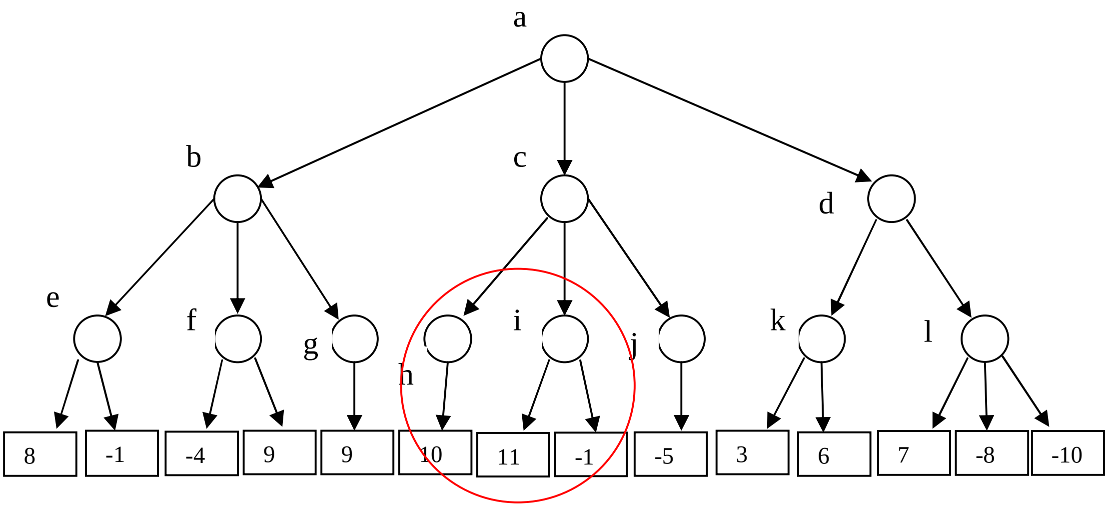
**(a)**




**(b)**



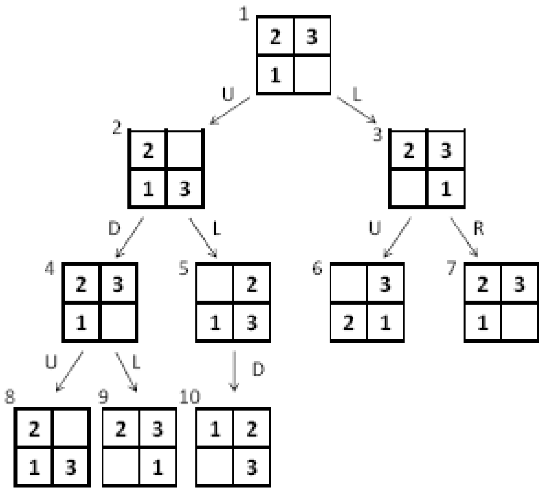
**(c)**



# 2 Informed search

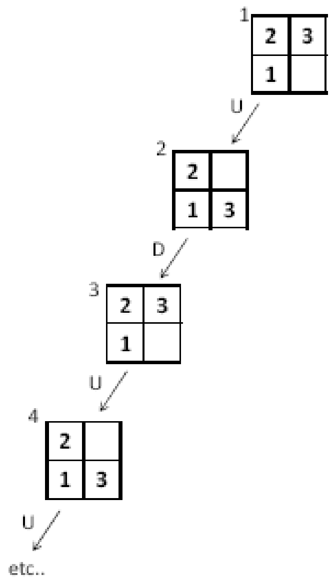
a)

Breadth first search will start from the root node, then expands all the successors of the root node, and then all their successors and so on. Breadth first search stops when first solution is found.



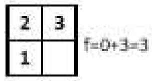

b)

Depth-first search always expands the deepest node in the search tree. Notice that there was no mechanism to remember states that have been visited earlier. Depth first will not find a solution as it will start oscillating between movements U and D.

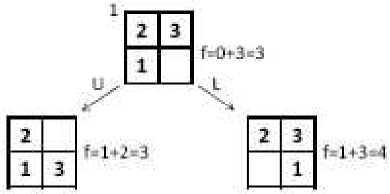


c)

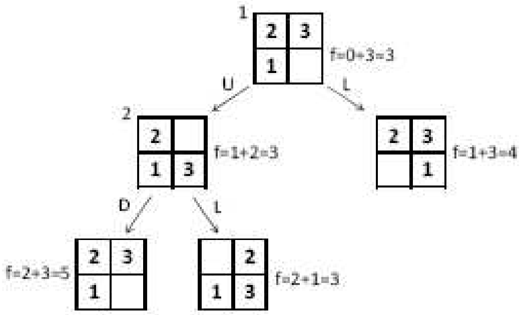
A* is a search method that opens the node with smallest cost function value. The search begins from the start state.



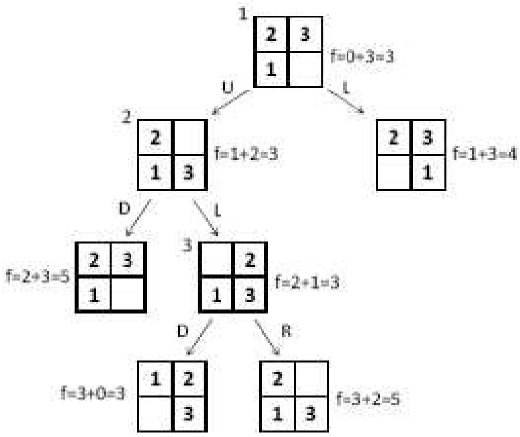

1. When the start state is opened we see two states




2. The leftmost state has a lower cost so that one is choosen and opened. We now see states with costs 5,3 and 4




3. The state with lowest cost is opened and we see two more nodes including the goal node. We notice that the goal node has the lowest cost so we choose that and can finish the search. If some other node with a lower cost function value was still visible, A* search would choose that instead of the higher cost goal node. This is because A* tries to find the path with lowest cost.



# 3 Logic

Let $r, b, v$ denote "Mr Rossi (respectively, Mr. Bianchi, Mr. Verdi) has voted for Berlusconi". The declarations of the three friends can be formalized as follows: Mr. Rossi declares that $b \to v$, Mr. Bianchi that $\neg v \to \neg r$ and Mr. Verdi that $b \wedge \neg v$. To easily answer the questions it suffices to build the truth table of $(b \to v) \wedge (\neg v \to \neg r) \wedge (b \wedge \neg r)$.

| $r$ | $b$ | $v$ | $b \to v$ | $\neg v \to \neg r$ | $b \wedge \neg r$ | $(b \to v) \wedge (\neg v \to \neg r) \wedge (b \wedge \neg r)$ |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 |

Answers:

(a) The declarations are compatible since $(b \to v) \wedge (\neg v \to \neg r) \wedge (b \wedge \neg r)$ is satisfiable.

(b) Mr. Verdi is lying since if $r, b, v$ are simultaneously true (first row) then Mr. Verdi's declaration $b \wedge \neg r$ is false.

(c) Mr. Verdi is lying again if $r, b, v$ are simultaneously false (last row).

(d) If they are all telling the truth (5th row), then Mr. Bianchi and Mr. Verdi voted for Berlusconi and Mr. Rossi against him.

(e) If they are all lying (2nd row), then Mr. Rossi and Bianchi voted for and Mr. Verdi against.

(f) The situation corresponds to the 4th row. Hence Mr. Rossi voted for, while Mr. Bianchi and Mr. Verdi against.

(g) The situation corresponds again to the 4th row. Hence Mr. Rossi voted for, while Mr. Bianchi and Mr. Verdi against.

# 4 Bayesian networks

Nodes F and H have no parents.
Node W has two parents: H and F.
Node B has one parent: H
Node T has one parent: B

$$P(W) = \Sigma_{F,H} P(W|F,H)P(F)P(H)$$

$$\begin{aligned} P(W|T) &= \frac{P(W,T)}{P(T)} \\ &= \frac{\Sigma_{B,H,F} P(W|H,F)P(H)P(F)P(T|B)P(B|H)}{\Sigma_{B,H} P(T|B)P(B|H)P(H)} \end{aligned}$$

# 5 Machine learning

(a)

Desired output $f(n) = \begin{cases} +1 & \text{if } \mathbf{x}(n) \text{ belongs to class } A \\ -1 & \text{if } \mathbf{x}(n) \text{ belongs to class } B \end{cases}$

Repeat until no errors are made anymore

1. Pick a random example [$\mathbf{x}(n)$,$f(n)$]
2. If the classification is correct,
   i.e. if $y(\mathbf{x}(n)) = f(n)$ , then do nothing
3. If the classification is wrong, then do the following update to the parameters
   ($\eta$, the learning rate, is a small positive number)

$$w_i = w_i + \eta f(n) x_i(n)$$

(b)

- **Simplest:** Construct a decision tree with one leaf for every example = memory based learning. Not very good generalization.
- **Advanced:** Split on each variable so that the purity of each split increases (i.e. either only yes or only no). Purity measured,e.g, with <u>entropy</u>