# Course "**Fundamentals of Artificial Intelligence**"
# EXAM TEXT

Prof. Roberto Sebastiani
DISI, Università di Trento, Italy

2024.01.11

37580917

[COPY WITH SOLUTIONS]

# 1

In the following FOL formulas, let $P$, $Q$, $R$, and $>, \leq, <, \geq$ denote predicates,
$f$, $g$, $h$, $F_1$, $F_2$, $F_3$ and $+, -, \cdot, /$ denote functions,
$x$, $y$, $z$, $x_1$, $x_2$, $x_3$ denote variables,
$A$, $B$, $C$, $C_1$, $C_2$, $C_3$ and $0, 1, 2, 3, 4$ denote constants.

For each of the following facts, say if it is true or false.

($a$) The FOL formula $\forall x_1 \forall x_2 . P(x_1, x_2)$ is equivalent to $\forall x_1 \forall x_2 . P(x_2, x_1)$
[ Solution: true (quantified-variable names are irrelevant provided there are no name conflicts) ]

($b$) The FOL formula $(2 > 4)$ is unsatisfiable.
[ Solution: false (in FOL ">", "2", "4" have no fixed interpretation) ]

($c$) The FOL formula $\forall x_1 \exists x_1 . Q(x_1)$ is equivalent to $\exists x_1 . Q(x_1)$
[ Solution: true (in case of name conflicts, inner quantifiers dominate) ]

($d$) The FOL formula $(\forall x_1 . \neg P(x_1)) \leftrightarrow (\neg \forall x_1 . P(x_1))$ is valid
[ Solution: false ]

# 2

Consider (normal) modal logics. Let $\mathsf{Engine} = \mathsf{F355355v8}$, $\mathsf{IsFast}(\mathsf{Engine})$, $\mathsf{IsFast}(\mathsf{F355355v8})$, $\mathsf{IsIdle}(\mathsf{Engine})$, $\mathsf{IsOff}(\mathsf{Switch})$ be possible facts, let $\mathsf{Joe}, \mathsf{Eve}$ be agents and let $\mathbf{K}_{\mathsf{Joe}}, \mathbf{K}_{\mathsf{Eve}}$ denote the modal operators "Joe knows that..." and "Eve knows that..." respectively.
For each of the following facts, say if it is true or false.

($a$) If $\mathsf{Engine} = \mathsf{F355355v8} \wedge \mathbf{K}_{\mathsf{Joe}}\mathsf{IsFast}(\mathsf{Engine})$ holds, then $\mathbf{K}_{\mathsf{Joe}}\mathsf{IsFast}(\mathsf{F355355v8})$ holds.
[ Solution: false ]

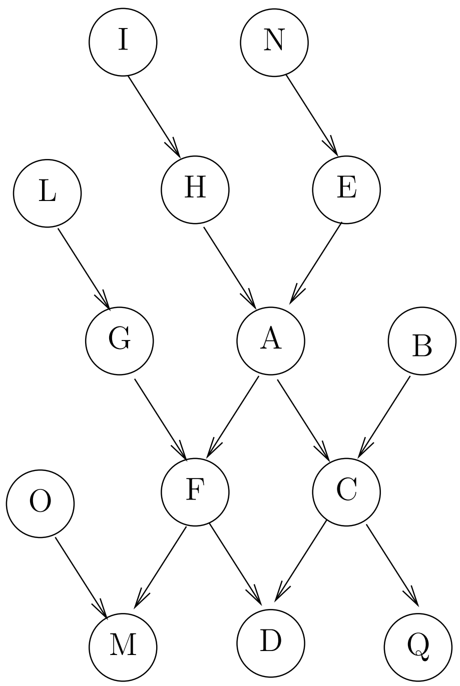
($b$) If $\mathbf{K}_{\mathsf{Eve}}\mathsf{Engine} = \mathsf{F355355v8} \wedge \mathbf{K}_{\mathsf{Eve}}\mathsf{IsFast}(\mathsf{Engine})$ holds, then $\mathbf{K}_{\mathsf{Eve}}\mathsf{IsFast}(\mathsf{F355355v8})$ holds.
[ Solution: true ]

($c$) If $\neg\mathbf{K}_{\mathsf{Joe}}\mathsf{IsIdle}(\mathsf{Engine})$ holds, then $\mathbf{K}_{\mathsf{Joe}}\neg\mathsf{IsIdle}(\mathsf{Engine})$ holds
[ Solution: false ]

($d$) If $\mathbf{K}_{\mathsf{Joe}}\mathsf{IsIdle}(\mathsf{Engine})$ and $\mathbf{K}_{\mathsf{Joe}}(\mathsf{IsIdle}(\mathsf{Engine}) \rightarrow \mathbf{K}_{\mathsf{Eve}}\mathsf{IsIdle}(\mathsf{Engine}))$ hold, then $\mathbf{K}_{\mathsf{Joe}}\mathbf{K}_{\mathsf{Eve}}\mathsf{IsIdle}(\mathsf{Engine}))$ holds
[ Solution: true ]

# 3

Consider the following DAG of a Bayesian network.




For each of the following facts, say if it is true or false.

($a$) For $\mathbf{P}(H|F)$, G is irrelevant.
[ Solution: false, because it belongs to $Ancestors(H, F)$. ]

($b$) For $\mathbf{P}(G|A)$, C is irrelevant.
[ Solution: true, because it does not belongs to $Ancestors(G, A)$. ]

($c$) $\mathbf{P}(C|ABHF) = \mathbf{P}(C|AB)$
[ Solution: true, due to local semantics (F is a nondescendant of C) ]

($d$) $\mathbf{P}(F|GACDOMHQ) = \mathbf{P}(F|GACDOM)$
[ Solution: true, due to Markov blanket rule ]

# 4

Let us consider the PEAS description of the task environment for an automated taxi. For each of the following facts, say if it is true or false.

($a$) Safety is a Sensor
[ Solution: false ]

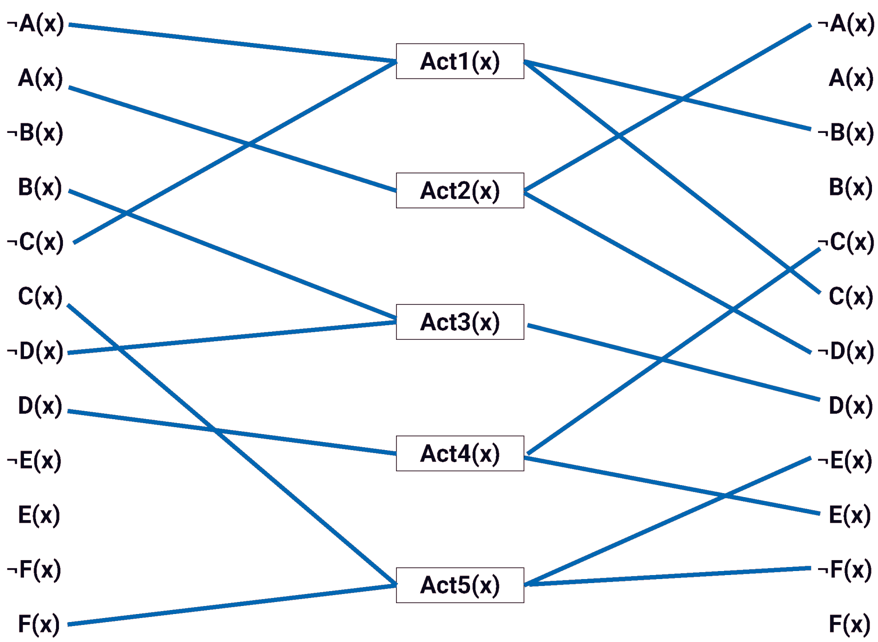
($b$) Police is a Environment
[ Solution: true ]

($c$) Speech is a Actuator
[ Solution: true ]

($d$) Cameras is a Actuator
[ Solution: false ]

# 5

Consider the graph shown below.




For each of the following facts, say if it is true or false.

($a$) There is at least one mutex between Act2 and Act5.
[ Solution: false ]

($b$) The mutex between Act2 and Act4 is an interference.
[ Solution: true ]

($c$) There are no mutex between A(x) and D(x).
[ Solution: true ]

($d$) The mutex between the persistence of A(x) and Act5 is an inconsistent effect.
[ Solution: false ]

# 6

($a$) Describe as Pseudo-Code the Uniform-Cost Search (UCS) stategy (graph version).

[ Solution:

```
function UNIFORM-COST-SEARCH(problem) returns a solution, or failure
  node ← a node with STATE = problem.INITIAL-STATE, PATH-COST = 0
  frontier ← a priority queue ordered by PATH-COST, with node as the only element
  explored ← an empty set
  loop do
      if EMPTY?(frontier) then return failure
      node ← POP(frontier)  /* chooses the lowest-cost node in frontier */
      if problem.GOAL-TEST(node.STATE) then return SOLUTION(node)
      add node.STATE to explored
      for each action in problem.ACTIONS(node.STATE) do
          child ← CHILD-NODE(problem, node, action)
          if child.STATE is not in explored or frontier then
              frontier ← INSERT(child, frontier)
          else if child.STATE is in frontier with higher PATH-COST then
              replace that frontier node with child
```

or any schema equivalent to the above one (from AIMA book). ]

($b$) When is the goal test applied to a node?

1 when the node is selected for expansion

2 when the node is first generated

(Say if 1. or 2.)

[ Solution: 1. ]

# 7

Consider the following propositional formula $\varphi$:

$$((\neg A_3 \wedge \neg A_2) \leftrightarrow ( \quad A_5 \wedge \neg A_4))$$

1. Using the $CNF_{label}$ conversion, produce the CNF formula $CNF_{label}(\varphi)$.

   [ Solution: We introduce fresh Boolean variables naming the subformulas of $\varphi$:

$$\overbrace{(\overbrace{(\neg A_3 \wedge \neg A_2)}^{B_1} \leftrightarrow \overbrace{( \quad A_5 \wedge \neg A_4)}^{B_2})}^{B}$$

   from which we obtain:

$$\begin{array}{ll} & ( \quad B) \\ \wedge & (\neg B \vee \neg B_1 \vee \quad B_2) \\ \wedge & (\neg B \vee \quad B_1 \vee \neg B_2) \\ \wedge & ( \quad B \vee \quad B_1 \vee \quad B_2) \\ \wedge & ( \quad B \vee \neg B_1 \vee \neg B_2) \\ \wedge & (\neg B_1 \vee \neg A_3) \wedge (\neg B_1 \vee \neg A_2) \\ \wedge & ( \quad B_1 \vee \quad A_3 \vee \quad A_2) \\ \wedge & (\neg B_2 \vee \quad A_5) \wedge (\neg B_2 \vee \neg A_4) \\ \wedge & ( \quad B_2 \vee \neg A_5 \vee \quad A_4) \end{array} .$$

   ]

2. For each of the following sentences, only one is true. Say which one.

   (i) $\varphi$ and $CNF_{label}(\varphi)$ are equivalent. [ Solution: False ]

   (ii) $\varphi$ and $CNF_{label}(\varphi)$ are not necessarily equivalent. $CNF_{label}(\varphi)$ has a model if and only $\varphi$ has a model. [ Solution: true ]

   (iii) There is no relation between the satisfiablity of $\varphi$ and that of $CNF_{label}(\varphi)$. [ Solution: False ]

# 8

For each of the following FOL formulas, compute its CNF-Ization.
Use symbols $C_1, C_2, C_3, ...$ for Skolem constants and symbols $F_1, F_2, F_3, ...$ for Skolem functions.

$(a)$ $\forall x.[\exists y.\forall z.P(x,y,z) \to \exists y.\forall z.Q(x,y,z)]$
[ Solution:
$\forall x.[\exists y.\forall z.P(x,y,z) \to \exists y.\forall z.Q(x,y,z)]$
$\forall x.[\neg\exists y.\forall z.P(x,y,z) \vee \exists y.\forall z.Q(x,y,z)]$
$\forall x.[\forall y.\exists z.\neg P(x,y,z) \vee \exists y.\forall z.Q(x,y,z)]$
$\forall x.[\forall y_1.\exists z_1.\neg P(x,y_1,z_1) \vee \exists y_2.\forall z_2.Q(x,y_2,z_2)]$ //not strictly necessary
$\neg P(x,y_1,F_1(x,y_1)) \vee Q(x,F_2(x),z_2)$
]

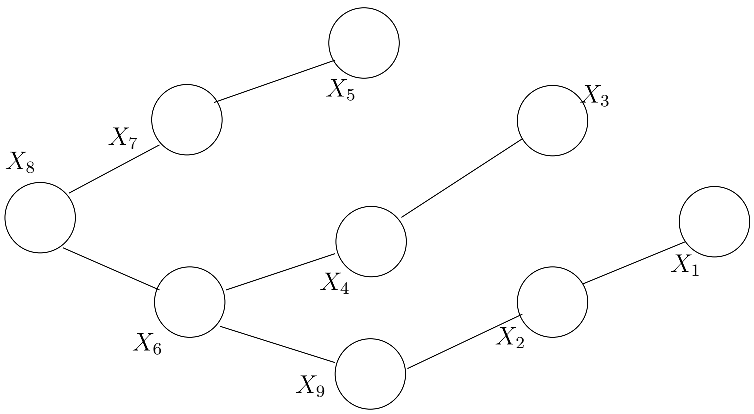
$(b)$ $\exists x.[\forall y.\exists z.P(x,y,z) \to \forall y.\exists z.Q(x,y,z)]$
[ Solution:
$\exists x.[\forall y.\exists z.P(x,y,z) \to \forall y.\exists z.Q(x,y,z)]$
$\exists x.[\neg\forall y.\exists z.P(x,y,z) \vee \forall y.\exists z.Q(x,y,z)]$
$\exists x.[\exists y.\forall z.\neg P(x,y,z) \vee \forall y.\exists z.Q(x,y,z)]$
$\exists x.[\exists y_1.\forall z_1.\neg P(x,y_1,z_1) \vee \forall y_2.\exists z_2.Q(x,y_2,z_2)]$ //not strictly necessary
$\neg P(C_1,C_2,z_1) \vee Q(C_1,y_2,F_1(y_2))$ ]

$(c)$ $\exists x.[\forall y.P(x,y) \leftrightarrow \exists z.Q(x,z)]$
[ Solution:
$\exists x.[\forall y.P(x,y) \leftrightarrow \exists z.Q(x,z)]$
$\exists x.[(\neg\forall y.P(x,y) \vee \exists z.Q(x,z)) \wedge (\forall y.P(x,y) \vee \neg\exists z.Q(x,z))]$
$\exists x.[(\exists y.\neg P(x,y) \vee \exists z.Q(x,z)) \wedge (\forall y.P(x,y) \vee \forall z.\neg Q(x,z))]$
$\exists x.[(\exists y_1.\neg P(x,y_1) \vee \exists z_1.Q(x,z_1)) \wedge (\forall y_2.P(x,y_2) \vee \forall z_2.\neg Q(x,z_2))]$ //not strictly necessary
$(\neg P(C_1,C_2) \vee Q(C_1,C_3)) \wedge (P(C_1,y_2) \vee \neg Q(C_1,z_2))$
]

$(d)$ $\forall x.[\exists y.P(x,y) \leftrightarrow \exists z.Q(x,z)]$
[ Solution:
$\forall x.[\exists y.P(x,y) \leftrightarrow \exists z.Q(x,z)]$
$\forall x.[(\neg\exists y.P(x,y) \vee \exists z.Q(x,z)) \wedge (\exists y.P(x,y) \vee \neg\exists z.Q(x,z))]$
$\forall x.[(\forall y.\neg P(x,y) \vee \exists z.Q(x,z)) \wedge (\exists y.P(x,y) \vee \forall z.\neg Q(x,z))]$
$\forall x.[(\forall y_1.\neg P(x,y_1) \vee \exists z_1.Q(x,z_1)) \wedge (\exists y_2.P(x,y_2) \vee \forall z_2.\neg Q(x,z_2))]$ //not strictly necessary
$(\neg P(x,y) \vee Q(x,F_1(x))) \wedge (P(x,F_2(x)) \vee \neg Q(x,z_2))$
]

# 9

Consider the following tree-structured map-coloring problem, with domain $D \stackrel{\text{def}}{=} \{\text{Red}, \text{Green}, \text{Blue}\}$, with the following initial domain restrictions:




| | | | |
|---|---|---|---|
| $X_1 = \{$ | Red , | , | $\}$ |
| $X_2 = \{$ | Red | , Green | , $\}$ |
| $X_3 = \{$ | | , Green | , $\}$ |
| $X_4 = \{$ | Red | , Green | , $\}$ |
| $X_5 = \{$ | , | | , Blue $\}$ |
| $X_6 = \{$ | Red | , Green | , Blue $\}$ |
| $X_7 = \{$ | | , Green | , Blue $\}$ |
| $X_8 = \{$ | Red | , Green | , Blue $\}$ |
| $X_9 = \{$ | | , Green | , Blue $\}$ |

**Using the Tree-Structured Algorithm**, and considering the following node ordering: $\{X_8, X_7, X_5, X_6, X_4, X_3, X_9, X_2, X_1\}$,

($a$) show the ordered progression of domain restrictions induced by the algorithm.

[ Solution:

| | | | |
|---|---|---|---|
| $X_1 = \{$ | Red | , | , $\}$ |
| $X_2 = \{$ | | , Green | , $\}$ |
| $X_9 = \{$ | | , | , Blue $\}$ |
| $X_3 = \{$ | | , Green | , $\}$ |
| $X_4 = \{$ | Red | , | , $\}$ |
| $X_6 = \{$ | | , Green | , $\}$ |
| $X_5 = \{$ | | , | , Blue $\}$ |
| $X_7 = \{$ | | , Green | , $\}$ |
| $X_8 = \{$ | Red | , | , Blue $\}$ |

]

($b$) As a consequence of the above process, state if the problem is solvable or not. If yes, show one solution. If not, explain why it is not.

[ Solution: It is solvable. One possible solution is: $\{X_1 = \text{Red}, X_2 = \text{Green}, X_9 = \text{Blue}, X_3 = \text{Green}, X_4 = \text{Red}, X_6 = \text{Green}, X_5 = \text{Blue}, X_7 = \text{Green}, X_8 = \text{Red}\}$ ]

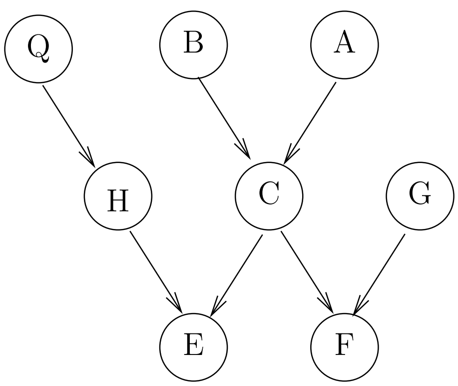
($c$) Let $n, d$ be ths number of nodes and the (maximum) domain size respectively. What is the worst-case complexity of this algortithm?

[ Solution: $O(n \cdot d^2)$ ]

Notation: to represent the current domain of a node $X_i$, substitute with a blank " " any value in $\{\text{Red}, \text{Green}, \text{Blue}\}$ which cannot be assigned. (Ex: in current graph: $X_5 : \{\ ,\ , \text{Blue}\}$)

# 10

Consider the following Bayesian network:




where, for every node $X$, $Y$, $Z$:

- $P(x) \stackrel{\text{def}}{=} 0.6$, if $X$ has no parent
- $P(x|Y) \stackrel{\text{def}}{=}$

| $Y$ | $P(x\|Y)$ |
|---|---|
| $T$ | 0.4 |
| $F$ | 0.3 |

if $X$ has only one parent $Y$

- $P(x|Y,Z) \stackrel{\text{def}}{=}$

| $Y$ | $Z$ | $P(x\|Y,Z)$ |
|---|---|---|
| $T$ | $T$ | 0.7 |
| $F$ | $T$ | 0.6 |
| $T$ | $F$ | 0.2 |
| $F$ | $F$ | 0.1 |

if $X$ has one left parent $Y$ and one right parent $Z$.

Compute the conditional distribution $\mathbf{P(A|C)}$.

[ Solution: Recall the Irrelevant-variable Theorem: For every query $X$ and evidence $E$, variable $Y$ is irrelevant unless $Y \in Ancestors(X,E)$. Hence, all variables except $C, A, B$ are irrelevant for $\mathbf{P(A|C)}$. Thus:

$$\mathbf{P(A|C)} = \alpha\mathbf{P(A,C)} = \alpha\sum_B \mathbf{P(A,C,B)} = \alpha\sum_B \mathbf{P(C|A,B)} * \mathbf{P(A)} * \mathbf{P(B)} = \alpha\mathbf{P(A)} * \sum_B \mathbf{P(C|A,B)} * \mathbf{P(B)}$$

$$\begin{aligned}
P(a|c) &= \alpha_1 \cdot P(a)\,[P(c|a,b)P(b) + P(c|a,\neg b)P(\neg b)] \\
&= \alpha_1 \cdot 0.6 * (0.7 * 0.6 + 0.6 * 0.4) = \alpha_1 \cdot 0.396 \\
P(\neg a|c) &= \alpha_1 \cdot P(\neg a)\,[P(c|\neg a,b)P(b) + P(c|\neg a,\neg b)P(\neg b)] \\
&= \alpha_1 \cdot 0.4 * (0.2 * 0.6 + 0.1 * 0.4) = \alpha_1 \cdot 0.064 \\
\alpha_1 &= 1/(0.6 * (0.7 * 0.6 + 0.6 * 0.4) + 0.4 * (0.2 * 0.6 + 0.1 * 0.4)) \simeq 2.173 \\
(P(a|c),)P(\neg a|c)) &\simeq (0.86, 0.139) \\
P(a|\neg c) &= \alpha_2 \cdot P(a)\,[P(\neg c|a,b)P(b) + P(\neg c|a,\neg b)P(\neg b)] \\
&= \alpha_2 \cdot 0.6 * (0.3 * 0.6 + 0.4 * 0.4) = \alpha_2 \cdot 0.204 \\
P(\neg a|\neg c) &= \alpha_2 \cdot P(\neg a)\,[P(\neg c|\neg a,b)P(b) + P(\neg c|\neg a,\neg b)P(\neg b)] \\
&= \alpha_2 \cdot 0.4 * (0.8 * 0.6 + 0.9 * 0.4) = \alpha_2 \cdot 0.336 \\
\alpha_2 &= 1/(0.6 * (0.3 * 0.6 + 0.4 * 0.4) + 0.4 * (0.8 * 0.6 + 0.9 * 0.4)) \simeq 1.851 \\
(P(a|\neg c),)P(\neg a|\neg c)) &\simeq (0.377, 0.621)
\end{aligned}$$

]

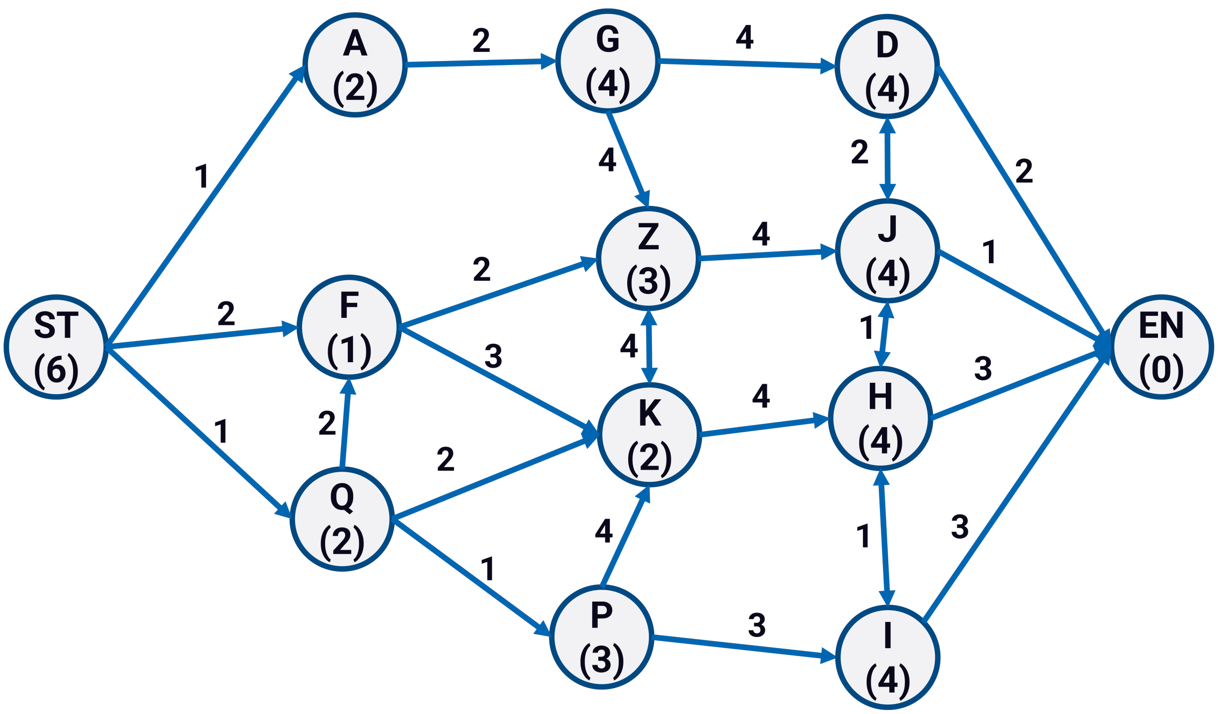
Notation: uppercase letters are used for propositional variables representing random events, whereas the corresponding lowercase letters represent truth assignments to such propositional variables.
Ex: $a \stackrel{\text{def}}{=} (A = \mathsf{true})$, $\neg a \stackrel{\text{def}}{=} (A = \mathsf{false})$.

# 11

The graph shown represents the states space of a hypothetical search problem where:

- States are denoted by letters.
- Arcs are labeled with the cost of traversing them.
- The estimated cost to a goal (i.e., the $h$ function) is reported inside nodes (so that lower scores are better).

Considering $ST$ as the initial state and $EN$ as the goal state, please apply the **A*** search algorithm and report each step of the resolution process. Then, explain if the heuristic adopted is admissible or not. In the case of possible equal choices please proceed with the selection in alphabetical order. The solution format has to be provided as shown during the laboratory session.

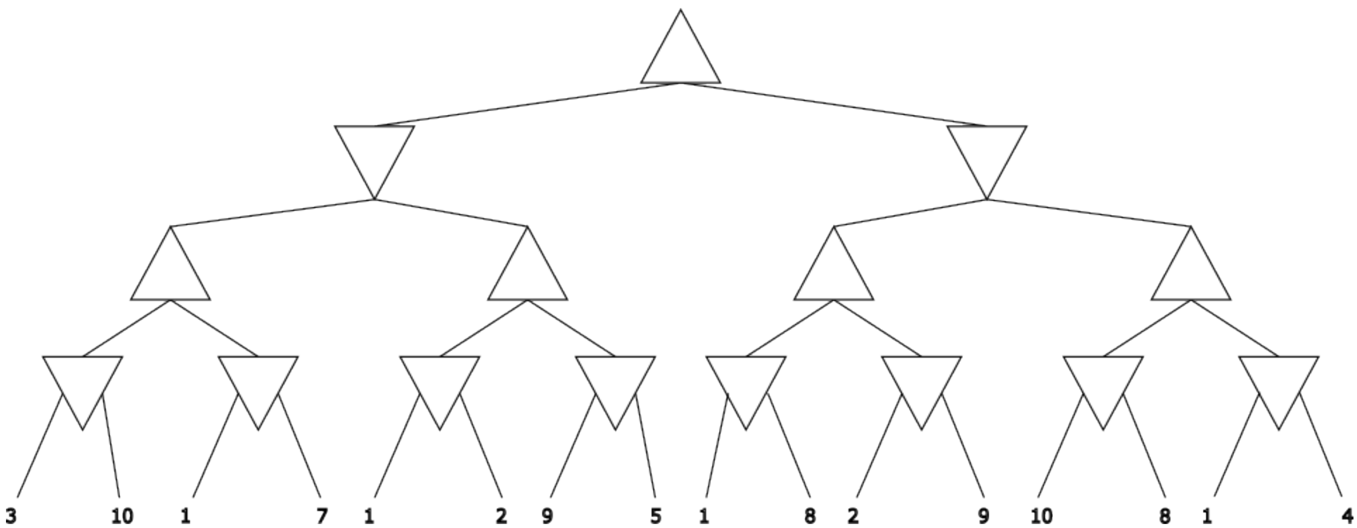



[ Solution:

]

# 12

Solve the following adversarial search problem using Alpha/Beta search:

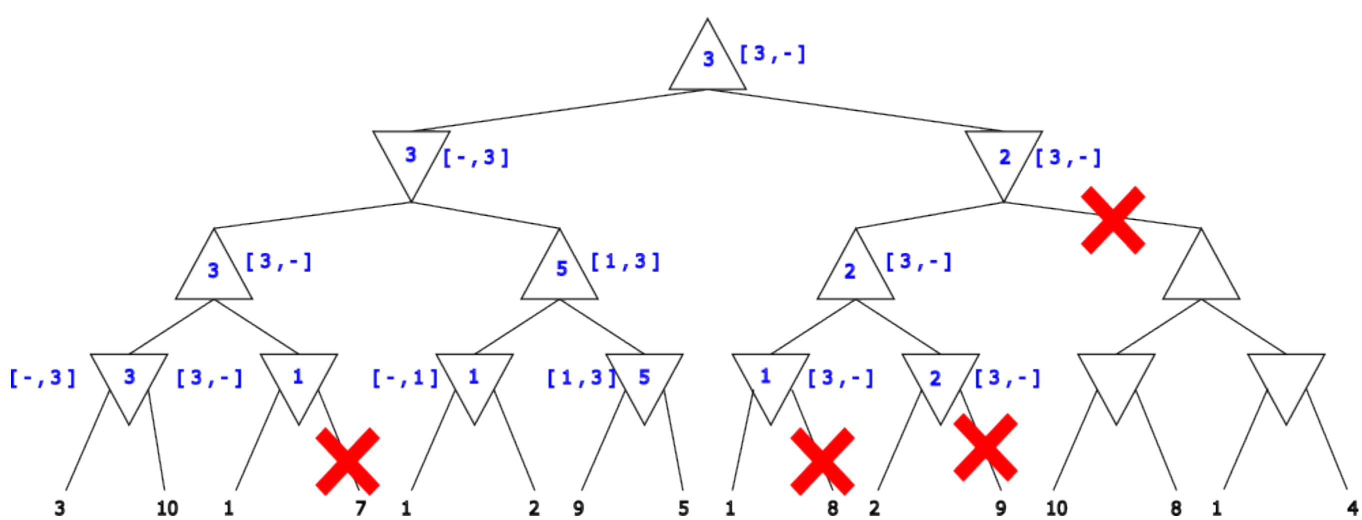



For each visited non-terminal node, write:

- its return value (inside the triangle)
- its $\alpha$ and $\beta$ values right before returning (outside the triangle)

Additionally, clearly cross the pruned subtrees (not visited by the algorithm).

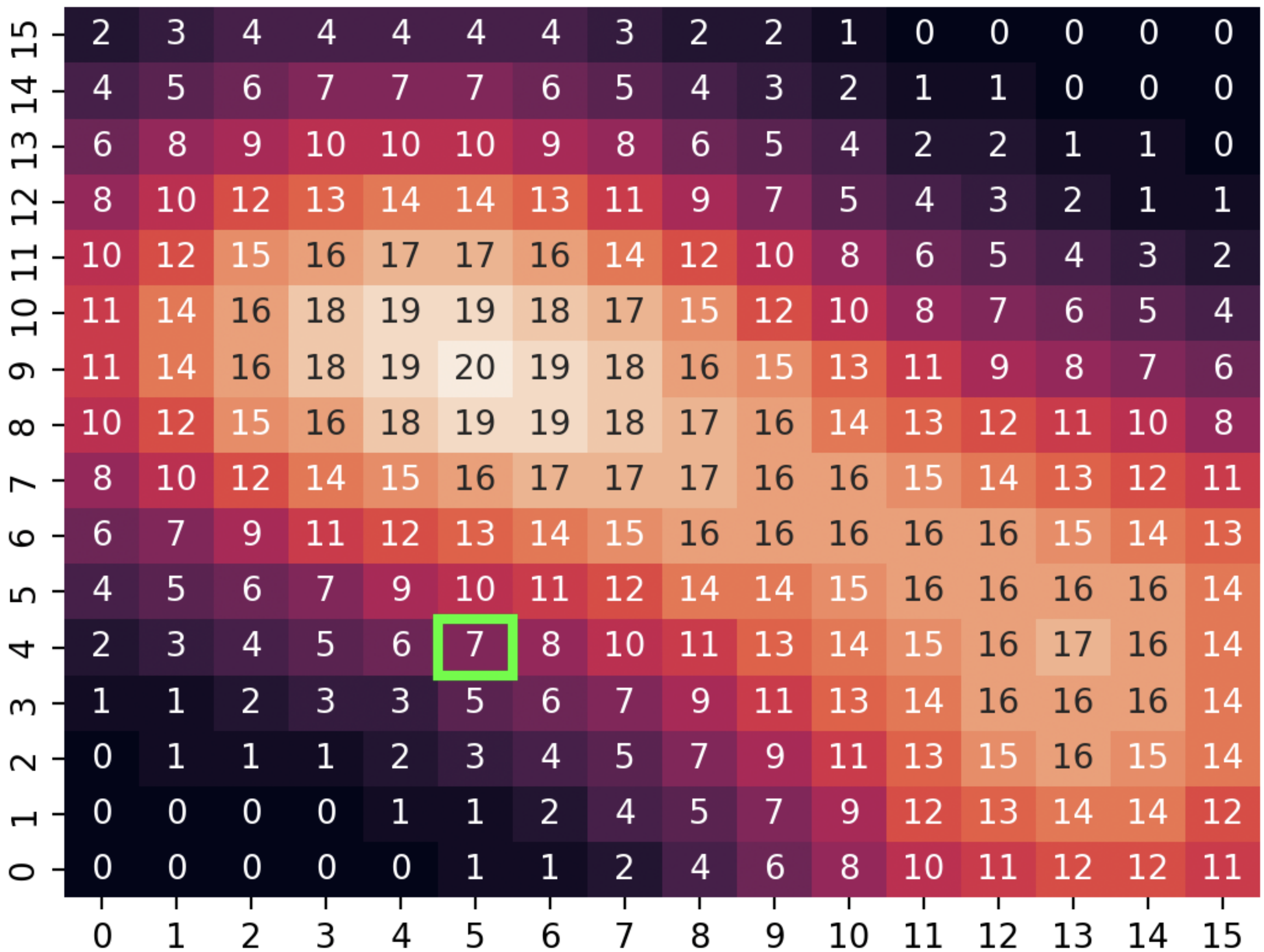
[ Solution:




]

## 13

Consider the landscape of the following *maximization* problem:




with starting solution: $(5,4)$

Compute the sequence of solutions generated with *steepest* Hill Climbing. The neighbours order is: SouthWest, South, SouthEast, West, East, NorthWest, North, NorthEast.

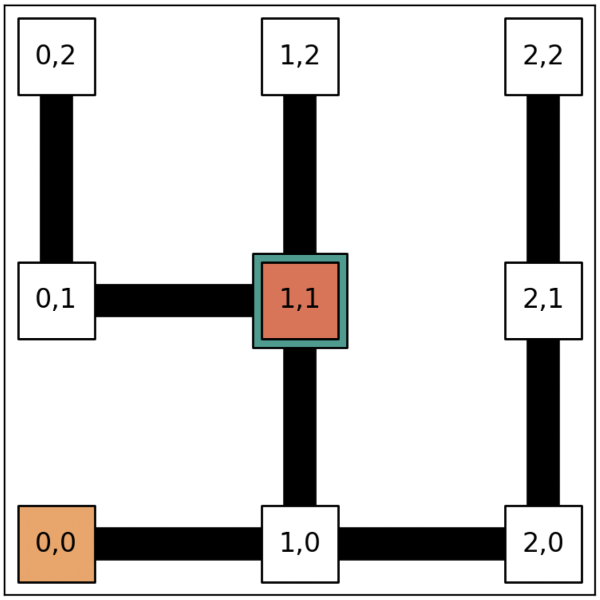
[ Solution:

$(5,4)$, $(6,5)$, $(7,6)$, $(8,7)$, $(7,8)$, $(6,9)$, $(5,9)$

]

# 14

Consider the following state graph, with undirected edges having uniform cost:




Start: (1,1)
Goal: (0,0)
Action order (when equal est. cost):
North, East, South, West

Compute the sequence of visted nodes by an agent using $LRTA^*$.

[ Solution:

$$(1,1),\ (1,2),\ (1,1),\ (1,0),\ (1,1),\ (1,0),\ (2,0),\ (2,1),\ (2,2),\ (2,1),\ (2,0),\ (1,0),\ (0,0)$$

]

# 15

Consider the following actions with durations / dependencies:

| a | d(a) |
|---|---|
| A | 5 |
| B | 2 |
| C | 3 |
| D | 10 |
| E | 2 |
| F | 9 |

$A \prec C$

$B \prec C$

$C \prec E$

$D \prec E$

Using the Critical Path method:

- Compute the earliest / latest possible start time (ES/LS) for each action.
- Indicate which actions are in the critical path and the minimum makespan.

[ Solution:

| a | ES(a) | LS(a) |
|---|---|---|
| A | 0 | 2 |
| B | 0 | 5 |
| C | 5 | 7 |
| D | 0 | 0 |
| E | 10 | 10 |
| F | 0 | 3 |

Critical path: D, E.
Makespan: 12
]