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

Prof. Roberto Sebastiani
DISI, Università di Trento, Italy

2026.01.13

862927

[COPY WITH SOLUTIONS]

1

Given:

- the constant symbols, representing proper names: $\{John\}$
- the unary predicate symbols, representing adjectives or categories: $\{Expensive, Jewel, Prince, Princess, Handsome, Pretty, Man, Woman, Person\}$
- the binary predicate symbols, representing verbs: $\{Owns, Spoils, Beats, Loves, Respects, Appreciate\}$

with their standard interpretation.

For each of the following pairs ⟨[English sentence], [FOL formula]⟩, say if [English sentence] is correctly encoded into [FOL formula].

(a) ⟨ "Every man is loved by some woman" ,
$\exists y.\{Woman(y) \wedge \forall x.[Man(x) \rightarrow Loves(x, y)]\}$ ⟩
[Solution: False. The above formula means "There is a woman who is loved by every man.". The correct encoding is $\forall x.\{Man(x) \rightarrow [\exists y.Woman(y) \wedge Loves(x, y)]\}$]

(b) ⟨ "Some pretty woman loves John",
$\exists x.(Woman(x) \wedge Pretty(x) \wedge Loves(x, John)$ ⟩
[Solution: True]

(c) ⟨ "Every prince owns some expensive jewel",
$\forall x.(Prince(x) \rightarrow \exists y.(Jewel(y) \wedge Expensive(y) \wedge Owns(x, y)))$ ⟩
[Solution: True]

(d) ⟨ "Every man who respects all persons is appreciated by some person." ,
$\forall x.\forall y.\{[Man(x) \wedge Person(y) \wedge Respects(x, y)] \rightarrow [\exists z.(Person(z) \wedge Appreciate(z, x))]\}$ ⟩
[Solution: False. The quantification of "y" should be part of the implicant, e.g.:
$\forall x.\{Man(x) \wedge [\forall y.(Person(y) \rightarrow Respects(x, y))] \rightarrow [\exists z.(Person(z) \wedge Appreciate(z, x))]\}$]

2

Consider (normal) modal logics. Let $\mathsf{IsOff}(\mathsf{Light}), \mathsf{IsIdle}(\mathsf{Pump})$ be possible facts, let $Claire, Paul$ be agents and let $\mathbf{K}_{Claire}, \mathbf{K}_{Paul}$ denote the modal operators "Claire knows that..." and "Paul knows that..." respectively.
For each of the following facts, say if it is true or false.

(a) If $\mathsf{IsOff}(\mathsf{Light}) \leftrightarrow \mathsf{IsIdle}(\mathsf{Pump})$ and $\mathbf{K}_{Paul}\mathsf{IsOff}(\mathsf{Light})$ hold, then $\mathbf{K}_{Paul}\mathsf{IsIdle}(\mathsf{Pump})$ holds
[Solution: false]

(b) If $\mathbf{K}_{Claire}(\mathsf{IsOff}(\mathsf{Light}) \rightarrow \mathbf{K}_{\mathsf{Paul}}\mathsf{IsIdle}(\mathsf{Pump}))$ and $\mathbf{K}_{Claire}\mathsf{IsOff}(\mathsf{Light})$ hold, then $\mathbf{K}_{Claire}\mathbf{K}_{Paul}\mathsf{IsIdle}(\mathsf{Pump}))$ holds
[Solution: true]

(c) If $\neg\mathbf{K}_{Claire}(\mathsf{IsOff}(\mathsf{Light}) \land \mathsf{IsIdle}(\mathsf{Pump}))$ holds, then $\neg\mathbf{K}_{Claire}\mathsf{IsOff}(\mathsf{Light}) \land \neg\mathbf{K}_{\mathsf{Claire}}\mathsf{IsIdle}(\mathsf{Pump})$ holds
[Solution: false]

(d) If $\neg\mathbf{K}_{Claire}\mathsf{IsOff}(\mathsf{Light}) \land \neg\mathbf{K}_{\mathsf{Claire}}\mathsf{IsIdle}(\mathsf{Pump})$ holds, then $\neg\mathbf{K}_{Claire}(\mathsf{IsOff}(\mathsf{Light}) \land \mathsf{IsIdle}(\mathsf{Pump}))$ holds
[Solution: true]

3

Given a generical search problem, assume time and space complexity are measured in terms of

b : maximum branching factor of the search tree

m : maximum depth of the state space (assume m is finite)

d : depth of the shallowest solution

Assume also that all steps cost are 1.

For each of the following facts about Iterative-Deepening Search, say if it is true or false.

(a) Iterative-Deepening Search is optimal
[Solution: true]

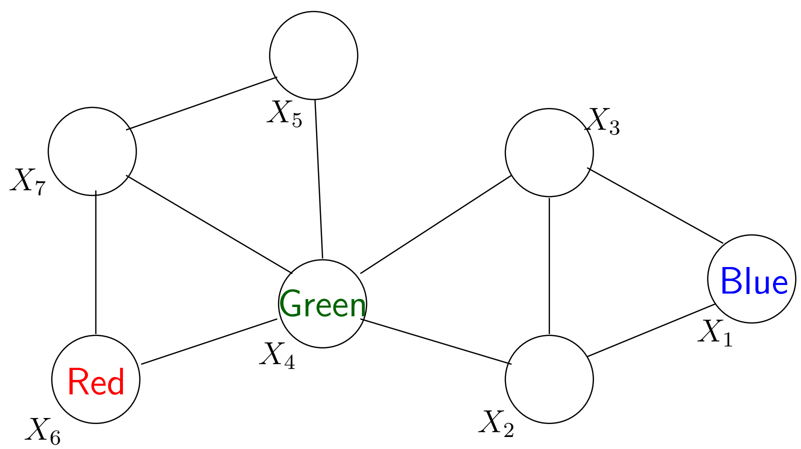
(b) Iterative-Deepening Search requires $O(b^d)$ memory to find a solution.
[Solution: false. It requirese $O(bd)$ memory.]

(c) Iterative-Deepening Search is complete.
[Solution: true]

(d) Iterative-Deepening Search requires $O(b^m)$ time to find a solution.
[Solution: false: it requires $O(b^d)$ time]

4

Consider the following constraint graph of a map coloring problem, with domain $D \stackrel{\text{def}}{=} \{$Green, Red, Blue$\}$, and consider the partial value assignment induced by the following unary constraints:
$\{X_1 =$ Blue, $X_4 =$ Green, $X_6 =$ Red, $\}$ (see figure).



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

(a) Forward checking reduces the domain of some variable to the empty set.
[Solution: false]

(b) AC-3 reduces the domain of some variable to the empty set.
[Solution: true]

(c) Forward Checking reduces the domain of X_5 to { , Red, }
[Solution: false]

(d) AC-3 reduces the domain of X_5 to { , Red, }
[Solution: true]

5

Consider propositional logic (PL); let F, C, D, E, B, A, G be atomic propositions
For each of the following statements, say if it is true or false.

(a) The subformula $(\neg F \wedge C)$ occurs only negatively in the formula $\neg((\neg F \wedge C) \rightarrow D)$
[Solution: False]

(b) The subformula $(F \vee \neg C)$ occurs both positively and negatively in the formula $\neg(D \leftrightarrow (F \vee \neg C))$
[Solution: True]

(c) The subformula $(F \rightarrow C)$ occurs only positively in the formula $(D \rightarrow (F \rightarrow C))$
[Solution: True]

(d) The subformula $(F \wedge \neg C)$ occurs only positively in the formula $((\neg D \vee E) \leftrightarrow (F \wedge \neg C))$
[Solution: False]

6

Let $R()$, $P()$, $Q()$ denote predicates, $h()$, $f()$, $g()$ denote functions, A, B denote constants, x, y, w, z denote variables, each possibly with suffixes.

For each of the following pair of clauses C_1, C_2:

- compute the most-general unifier (mgu) θ of their resolvent, or say "NULL" if none exists (represent the unifier as a list $\{variable_1/term_1, variable_2/term_2, ...\}$)
- compute the clause C resulting from resolving C_1 and C_2 with mgu θ, or say "NO SOLUTION" if resolution is not applicable

(a) $C_1 \stackrel{\text{def}}{=} R(h(x), y) \vee P(h(y), f(y))$,
$C_2 \stackrel{\text{def}}{=} R(h(A), g(z)) \vee \neg R(z, f(B))$
[Solution:
$\theta = \{z/h(x), y/f(B)\}$
$C = P(h(f(B)), f(f(B))) \vee R(h(A), g(h(x)))$]

(b) $C_1 \stackrel{\text{def}}{=} Q(h(A), h(z)) \vee \neg R(A, h(z))$,
$C_2 \stackrel{\text{def}}{=} R(y, z) \vee Q(x, g(y))$
[Solution:
$\theta =$ NULL ($h(z)$ cannot unify with z)
$C =$ NO SOLUTION]

(c) $C_1 \stackrel{\text{def}}{=} R(g(x), B) \vee Q(h(x), A)$
$C_2 \stackrel{\text{def}}{=} \neg R(A, y) \vee \neg Q(h(f(B)), y)$
[Solution:
$\theta = \{x/f(B), y/A\}$
$C = R(g(f(B)), B) \vee \neg R(A, A)$]

(d) $C_1 \stackrel{\text{def}}{=} R(x, g(x)) \vee P(x, h(z))$
$C_2 \stackrel{\text{def}}{=} \neg P(f(z), A) \vee R(f(B), y)$
[Solution:
$\theta =$ NULL ($h(z)$ cannot unify with A)
$C =$ NO SOLUTION]

7

Given the random propositional variables a, b, c and their joint probability distribution $\mathbf{P}(\mathbf{A}, \mathbf{B}, \mathbf{C})$ described as follows:

A	B	C	$\mathbf{P}(\mathbf{A}, \mathbf{B}, \mathbf{C})$
T	T	T	0.023
F	T	T	0.024
T	F	T	0.192
F	F	T	0.128
T	T	F	0.056
F	T	F	0.097
T	F	F	0.048
F	F	F	0.432

(a) Using marginalization, compute the probability $P(c)$

(b) Using normalization, compute the probability distribution $\mathbf{P}(\mathbf{A}|\neg b)$.
Note: Solving the problem by using any other technique will be considered incorrect.

[Solution:

(a) $P(c)$
$= P(a, b, c) + P(a, \neg b, c) + P(\neg a, b, c) + P(\neg a, \neg b, c) =$
$= 0.023 + 0.192 + 0.024 + 0.128 = 0.367.$

(b) Normalization: $\mathbf{P}(\mathbf{A}|\neg b) = \alpha \cdot \mathbf{P}(\mathbf{A}, \neg b)$, and then compute the value of α normalizing it. Thus:
$P(\ \ a|\neg b) = \alpha \cdot P(\ \ a, \neg b) = \alpha \cdot (P(\ \ a, \neg b, \ \ c) + P(\ \ a, \neg b, \neg c)) = \alpha \cdot (0.192 + 0.048) = \alpha \cdot 0.24$
$P(\neg a|\neg b) = \alpha \cdot P(\neg a, \neg b) = \alpha \cdot (P(\neg a, \neg b, \ \ c) + P(\neg a, \neg b, \neg c)) = \alpha \cdot (0.128 + 0.432) = \alpha \cdot 0.56$
that is, $\mathbf{P}(\mathbf{A}|\neg b) = \alpha \cdot \mathbf{P}(\mathbf{A}, \neg b) = \alpha \cdot \langle 0.24, 0.56 \rangle$

Then $\alpha = 1/(0.24 + 0.56) = 1.25.$
Thus $\mathbf{P}(\mathbf{A}|\neg b) = 1.25 \cdot \langle 0.24, 0.56 \rangle = \langle 0.3, 0.7 \rangle$

]

8

(*a*) Describe as Pseudo-Code the generic procedure for minimization via Simulated Annealing, assuming a function Schedule(t) returning a "temperature" parameter which slowly decreases with time step t, and a function Value(s) that measures the value of a state s to be minimized.

[Solution:

```
function SIMULATED-ANNEALING(problem, schedule) returns a solution state
  current ← problem.INITIAL
  for t = 1 to ∞ do
    T ← schedule(t)
    if T = 0 then return current
    next ← a randomly selected successor of current
    ΔE ← VALUE(current) – VALUE(next)
    if ΔE > 0 then current ← next      // if improved, then it is accepted
    else current ← next only with probability e^(ΔE/T)   //if worse, accepted with probability
                                                          // which decreases with time
```

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

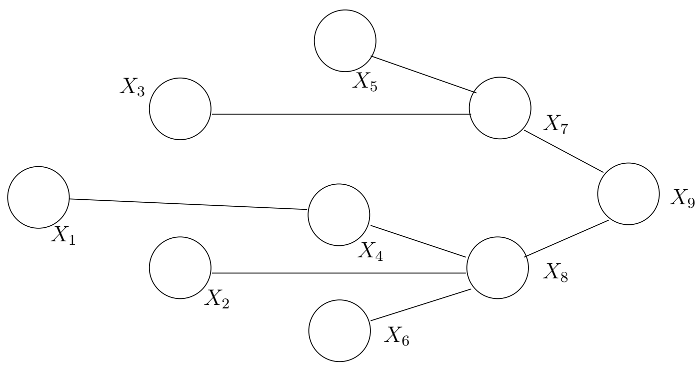
(*b*) Say if the following sentence is true or false, and briefly explain why.

- The probability of picking a "bad move" decreases as the "temperature" decreases.

[Solution: True.]

9

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



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

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

(a) Show the ordered progression of domain updates induced by the algorithm.

[Solution:

Node :	*Operation :*	*Updated*	*Domain :*			
X_1 :	make-arc-consistent $\langle X_4, X_1\rangle$:	$X_4 =$	$\{$	, Red	,	$\}$
X_2 :	make-arc-consistent $\langle X_8, X_2\rangle$:	$X_8 =$	$\{$Green	,	, Blue	$\}$
X_4 :	make-arc-consistent $\langle X_8, X_4\rangle$:	$X_8 =$	$\{$Green	,	, Blue	$\}$
X_6 :	make-arc-consistent $\langle X_8, X_6\rangle$:	$X_8 =$	$\{$Green	,	,	$\}$
X_8 :	make-arc-consistent $\langle X_9, X_8\rangle$:	$X_9 =$	$\{$	, Red	, Blue	$\}$
X_3 :	make-arc-consistent $\langle X_7, X_3\rangle$:	$X_7 =$	$\{$	, Red	,	$\}$
X_5 :	make-arc-consistent $\langle X_7, X_5\rangle$:	$X_7 =$	$\{$	, Red	,	$\}$
X_7 :	make-arc-consistent $\langle X_9, X_7\rangle$:	$X_9 =$	$\{$	,	, Blue	$\}$
X_9 :	*no parent*					

]

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

[Solution: It is solvable, because it produces no empty domain. The unique solution is thus: $\{X_9$: Blue, X_7 : Red, X_5 : Blue, X_3 : Blue, X_8 : Green, X_6 : Blue, X_4 : Red, X_2 : Red, X_1 : Green$\}$.]

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

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

Note: Solving the problem by using any other algorithm or any other ordering will be considered incorrect.
Notation: to represent the current domain of a node X_i, substitute with a blank " " any value in $\{\text{Green}, \text{Red}, \text{Blue}\}$ which cannot be assigned. (Ex: in current graph: $X_5 : \{\ ,\ , \text{Blue}\}$)

10

Consider the following Horn formula in PL:

$$
\begin{array}{llll}
(\neg E & \vee \neg D & \vee \ \ N) & \wedge \\
(\ \ D & \vee \neg C & \vee \neg B) & \wedge \\
(\neg C & \vee \ \ L & \vee \neg B) & \wedge \\
(\neg N & \vee \ \ I &) & \wedge \\
(\neg H & \vee \neg G & \vee \ \ N) & \wedge \\
(\ \ F & \vee \neg N & \vee \neg H) & \wedge \\
(\neg C & \vee \ \ A & \vee \neg E) & \wedge \\
(\neg D & \vee \ \ C & \vee \neg A) & \wedge \\
(\ \ H & \vee \neg N & \vee \neg I) & \wedge \\
(\ \ N &) & & \wedge \\
(\neg H & \vee \neg N & \vee \ \ G) & \wedge \\
(\neg N & \vee \ \ E & \vee \neg L) & \wedge \\
(\ \ H & \vee \neg F & \vee \neg N) & \wedge \\
(\neg N & \vee \neg C & \vee \ \ B) &
\end{array}
$$

Using the simple polynomial procedure for Horn formulas,

(a) decide if the formula is satisfiable or not

(b) if satisfiable, return the satisfying total truth assignment;
if unsatisfiable, return the falsified clause.

Note: Solving the problem by using any other procedure will be considered incorrect.

[Solution:

(a) (i) run unit propagation: N, I, H, F, G or, equivalently, N, I, H, G, F.
The resulting formula is:

$$
\begin{array}{llll}
(\ \ D & \vee \neg C & \vee \neg B) & \wedge \\
(\neg C & \vee \ \ L & \vee \neg B) & \wedge \\
 & & & \\
 & & & \\
(\neg C & \vee \ \ A & \vee \neg E) & \wedge \\
(\neg D & \vee \ \ C & \vee \neg A) & \wedge \\
 & & & \\
 & & & \\
(& \vee \ \ E & \vee \neg L) & \wedge \\
 & & & \\
(& \vee \neg C & \vee \ \ B) &
\end{array}
$$

(ii) the resulting formula contains no empty clause, and all clauses contain at least one negative literal by construction, thus the original formula is satisfiable.

(b) the satisfying truth assignment is $\{ N, I, H, F, G, \neg L, \neg E, \neg D, \neg C, \neg B, \neg A\}$.

]

11

Apply a *genetic algorithm* to the following optimization problem on ternary ($p_i \in \{0, 1, 2\}$) strings of length 4. The fitness function of individuals $\mathbf{p} = p_1p_2p_3p_4$ is:

$$f(\mathbf{p}) = \sum_{i=1}^{4} p_i \qquad \text{(e.g. } f(1210) = 4 \text{)}$$

Parents are selected from the available individuals with *probability directly proportional to their fitness value*.
There are 3 possible split points:

$$S1 = p_1|p_2p_3p_4$$
$$S2 = p_1p_2|p_3p_4$$
$$S3 = p_1p_2p_3|p_4$$

selected with *equal probability* 1/3. Mutations happen with probability 0.4 to each element independently.

$$[m(p_i) = p_i : 0.6, \qquad m(p_i) = (p_i + 1) \texttt{ mod } 3 : 0.4]$$

Apply a genetic algorithm until 4 new individuals are generated, report every step of the generation.

The *initial population* (iteration 0) is:

$$1201, 2000, 1021, 0002$$

The *choice vector* is:

$$[99/100, 1/2, 3/4]$$

Report the all the individuals of the population resulting from one full iteration of the algorithm, as well as the intermediate steps (i.e., for each new individual, how the indices of the parents and split point were selected).

[Solution:

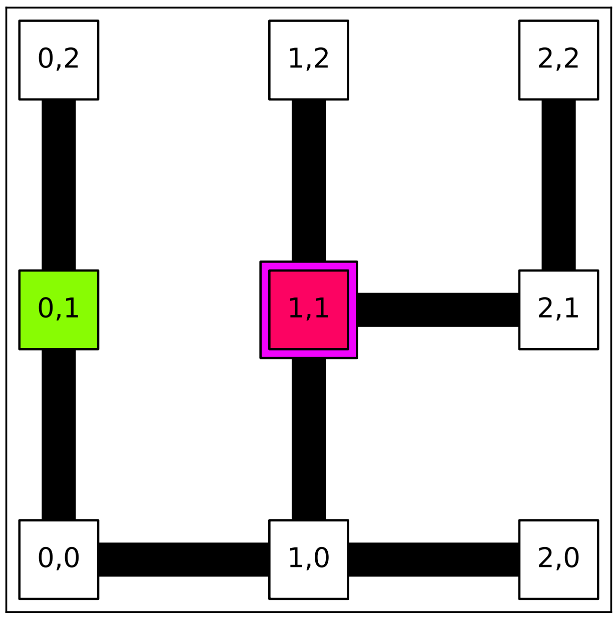
Resulting population:

1011 $[P_1 = 4(99/100), P_2 = 2(1/2), S = 3(3/4), mutations : M - MM(99/100, 1/2, 3/4, 99/100),$
2111 $[P_1 = 2(1/2), P_2 = 3(3/4), S = 3(99/100), mutations : -MM - (1/2, 3/4, 99/100, 1/2),$
2100 $[P_1 = 3(3/4), P_2 = 4(99/100), S = 2(1/2), mutations : MM - M(3/4, 99/100, 1/2, 3/4),$
1011 $[P_1 = 4(99/100), P_2 = 2(1/2), S = 3(3/4), mutations : M - MM(99/100, 1/2, 3/4, 99/100),$

]

12

In the following state graph, apply *online DFS* and report the list of visited states, including repetitions (e.g. $(0,1) \rightarrow (0,0) \rightarrow (0,1) \rightarrow ...$). The order of (untried) actions is $[up, right, down, left]$.



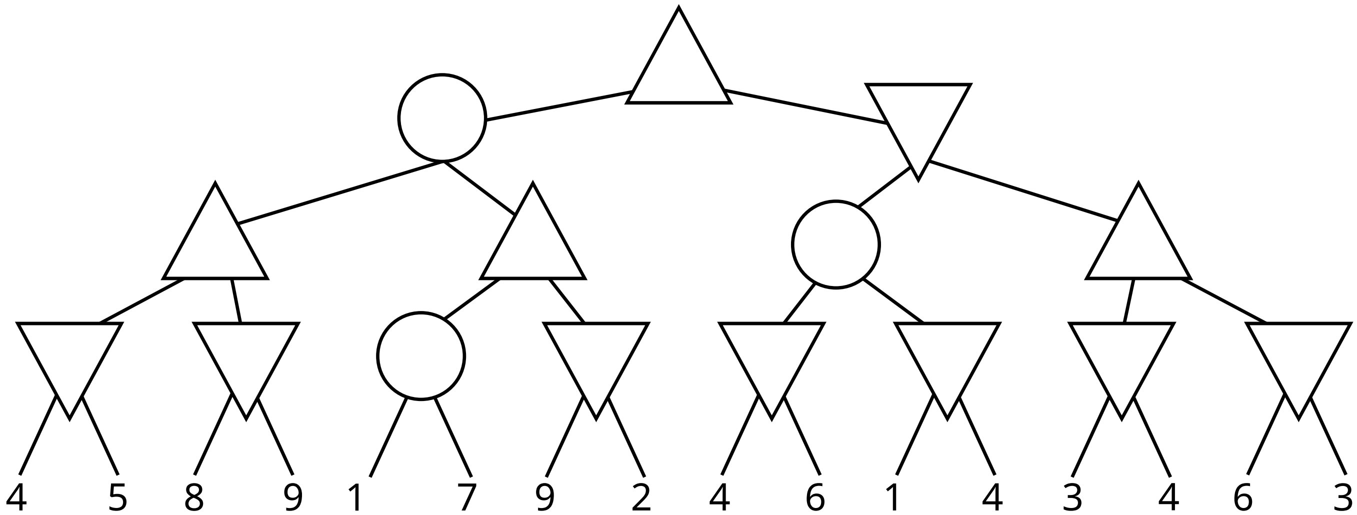
Start: $(1,1)$ - Goal: $(0,1)$

[Solution:

$(1,1)$, $(1,2)$, $(1,1)$, $(2,1)$, $(2,2)$, $(2,1)$, $(1,1)$, $(1,0)$, $(1,1)$, $(1,0)$, $(2,0)$, $(1,0)$, $(0,0)$, $(0,1)$
]

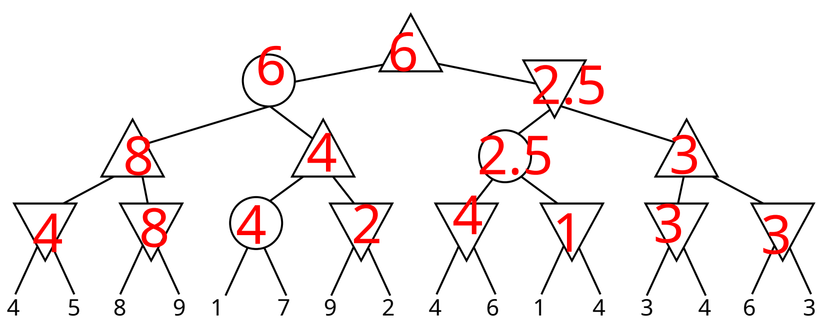
13

Use *ExpectiMinimax* to propagate the expected utilities from the leaves to the root of the tree. Chance nodes assign uniform probabilities to their children.



For each MIN/MAX/CHANCE node, write inside the expected utility.

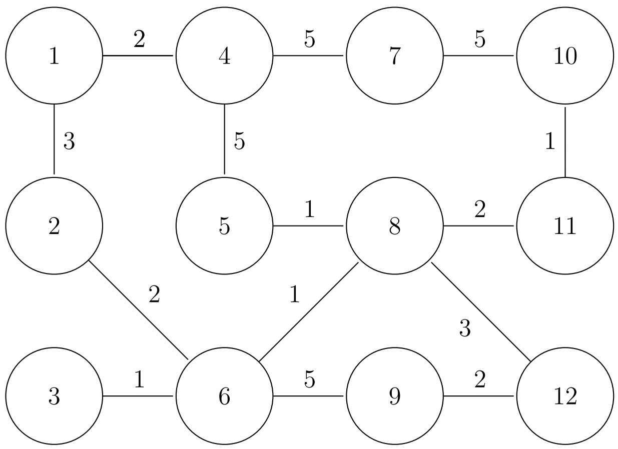
[Solution:



]

14

Apply A* to the following search problem:



Initial state: 9
Goal state: 7

All actions can be reverted and their cost is denoted on the edges.

- Actions are sorted according to the destination state with ascending numerical order

$$\text{e.g. } next(s, a) = 7 \wedge next(s, a') = 3 \Rightarrow a' < a$$

- The heuristic function is the Manhattan distance from the goal relative to the position in the grid (e.g., if the goal is in position $(2, 2)$, then the node n in position $(1, 0)$ has $h(n) = 3$).

[10 pt. /100] Is the heuristic function admissible? (Motivate your answer).
[90 pt. /100] For each step of the execution, report:

- The node extracted from the priority queue.
- The elements added to the priority queue *after* expanding.

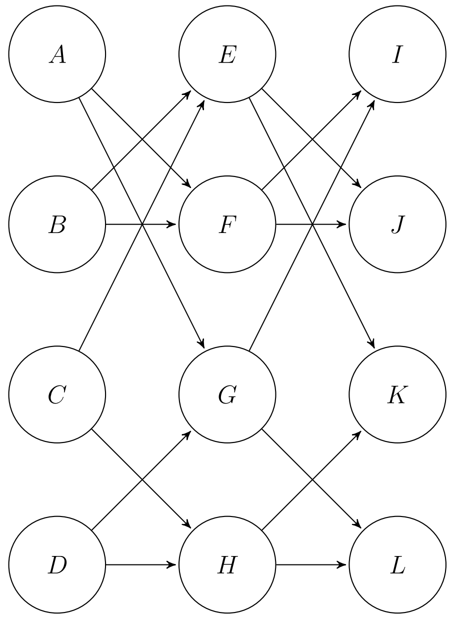
Then report the the path returned by A*.

[Solution:
Extracted: 9 Added to frontier [-6$(5+3)$, -12$(2+3)$]
Extracted: 9-12 Added to frontier [-8$(5+1)$]
Extracted: 9-12-8 Added to frontier [-5$(6+2)$, -11$(7+2)$]
Extracted: 9-6 Added to frontier [-2$(7+3)$, -3$(6+4)$]
Extracted: 9-12-8-5 Added to frontier [-4$(11+1)$]
Extracted: 9-12-8-11 Added to frontier [-10$(8+1)$]
Extracted: 9-12-8-11-10 Added to frontier [-7$(13+0)$]
Extracted: 9-6-2 Added to frontier [-1$(10+2)$]
Extracted: 9-6-3 Added to frontier []
Extracted: 9-12-8-5-4 Added to frontier []
Extracted: 9-6-2-1 Added to frontier []
Extracted and returned: 9-12-8-11-10-7]

15

Consider the following actions with durations / dependencies:

a	duration(a)
A	2
B	3
C	1
D	3
E	2
F	3
G	3
H	1
I	2
J	2
K	1
L	2



Using the Critical Path method:

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

[Solution:

a	ES	LS
A	0	1
B	0	0
C	0	3
D	0	0
E	3	4
F	3	3
G	3	3
H	3	5
I	6	6
J	6	6
K	5	7
L	6	6

Actions in CP: B, D, F, G, I, J, L
Makespan: 8]