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

Prof. Roberto Sebastiani
DISI, Università di Trento, Italy

2026.07.15

6305286

[COPY WITH SOLUTIONS]

1

Consider first-order-logic (FOL); let P, R, Q be predicates; let y, x be variables.
For each of the following statements, say if it is true or false.

(a) $\forall y.P(y) \wedge \forall x.R(x) \models \forall y.(P(y) \wedge R(y))$
[Solution: True]

(b) $\exists y \forall x \exists z.(P(y,x) \rightarrow R(y,z))$ is equivalent to $\neg P(C_1, x) \vee R(C_1, F_1(x))$ for some Skolem constant C_1 and function F_1
[Solution: False (it is only equivalently satisfiable to it)]

(c) $\exists y.(P(y) \vee R(y))$ is equivalent to $(\exists y.P(y)) \vee (\exists x.R(x))$
[Solution: True]

(d) $\forall y \exists x.Q(y,x) \models \exists x \forall y.Q(y,x)$
[Solution: False]

2

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

(*a*) $\mathbf{P}(D \mid E) = \mathbf{P}(E|D)$ if and only if $\mathbf{P}(D) = \mathbf{P}(E)$.
[Solution: True: $P(D \mid E) = P(E \mid D)$ if and only if $\frac{P(E|D)\,P(D)}{P(E)} = P(E \mid D)$,
if and only if $P(E \mid D)\,P(D) = P(E \mid D)\,P(E)$,
if and only if $P(D) = P(E)$]

(*b*) If D and E are conditionally independent given F, then $\mathbf{P}(D, E, F) = \mathbf{P}(D)\mathbf{P}(E \mid F)\mathbf{P}(F)$
[Solution: False: by the chain rule and D conditionally independent of E given F,
$P(D, E, F) = P(D \mid F)P(E \mid F)P(F)$. The proposed expression has $P(D)$ in place of $P(D \mid F)$.
]

(*c*) If D and E are conditionally independent given F, then $\mathbf{P}(D \mid E, F) = \mathbf{P}(D \mid F)$.
[Solution: True: $\mathbf{P}(D \mid E, F)) = \frac{\mathbf{P}(D,E,F)}{\mathbf{P}(E,F)} = \frac{\mathbf{P}(D,E|F)\mathbf{P}(F)}{\mathbf{P}(E|F)\mathbf{P}(F)} = \frac{\mathbf{P}(D|F)\mathbf{P}(E|F)}{\mathbf{P}(E|F)} = \mathbf{P}(D \mid F)$.]

(*d*) $\mathbf{P}(D, E) = \mathbf{P}(D|E)\mathbf{P}(E)$ always 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 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 $(\neg F \vee C)$ occurs only positively in the formula $\neg((\neg F \vee C) \rightarrow D)$
[Solution: True]

(*b*) The subformula $\neg D$ occurs only positively in the formula $((\neg F \vee E) \rightarrow \neg D)$
[Solution: True]

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

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

5

Consider the semantics of "$\alpha \to \beta$" ("α implies β"). Consider:

- the unary predicates $Italian()$, $French()$, $Russian()$, $American()$, denoting the corresponding nationalities
- the unary predicates $Odd()$, $Even()$ denoting the corresponding mathematical concepts,
- the constants $0, 1, 2, 3, 4, 5, 6, 7, ...$ denoting the corresponding integer numbers
- the constants $SergioMattarella$, $EmmanuelMacron$, $VladimirPutin$, $DonaldTrump$, denoting names of famous presidents (of Italy, France, Russia and USA respectively).

For each of the following FOL ground formulas, say whether it is true or false under a model which gives the standard interpretation of the above predicates and constants (that is, the interpretation which maps the constant "VladimirPutin" into the Russian president, the predicate "Italian()" into the property of being Italian, etc.).

(a) $Even(4) \to American(SergioMattarella)$.
[Solution: false]

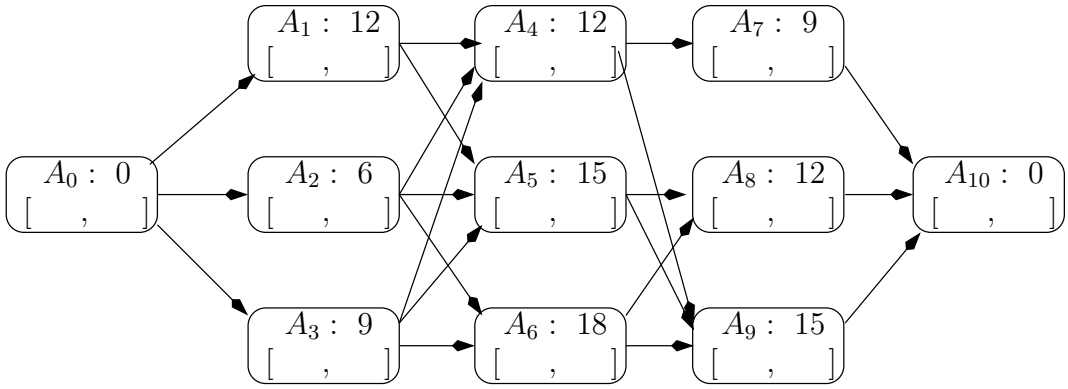
(b) $American(SergioMattarella) \to Even(4)$
[Solution: true]

(c) $American(SergioMattarella) \to Even(5)$.
[Solution: true]

(d) $French(EmmanuelMacron) \to Even(4)$.
[Solution: true]

6

Consider the following actions with durations and dependencies:



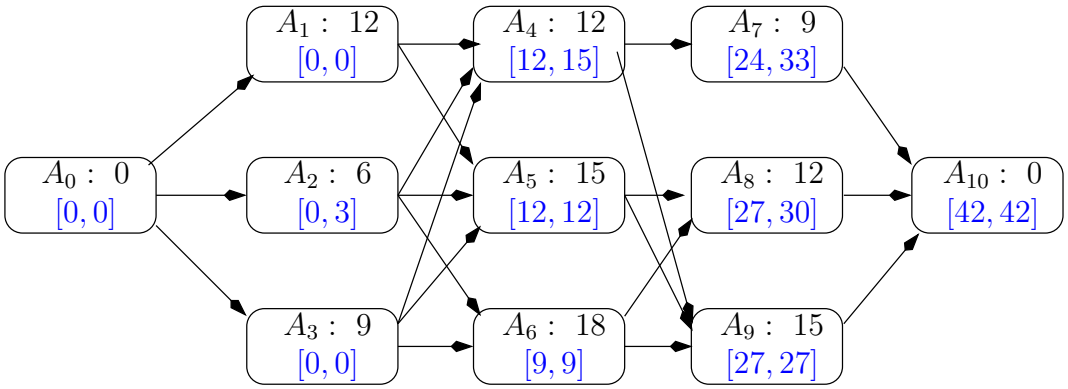
Using the Critical Path method:

(a) Compute the earliest / latest possible start time [ES, LS] for each action.
(Fill them into the "[,]" fields in the nodes.).

(b) Indicate all the actions that are in (at least one) critical path and the minimum makespan.

[Solution:

(a)



(b) The action involved in (at least one) critical path are: $\{A_0, A_1|A_3, A_5|A_6, A_9, A_{10}\}$.
The makespan is 42.

]

7

An email provider is tuning a spam filter. From historical data, 25% of incoming emails are spam. For each incoming email, the filter checks three simple indicators:

Indicator #1: the email contains the word “free”;
Indicator #2: the email contains a hyperlink;
Indicator #3: the email comes from an unknown sender.

The filter treats the three indicators as a Naive Bayes model, i.e. it assumes they are conditionally independent given whether the email is spam or not, and from past data it has learned the following statistics: [1]

$p(\text{indicator \#1}\mid \ spam)$	$= 0.55$
$p(\text{indicator \#1}\mid \neg spam)$	$= 0.05$
$p(\text{indicator \#2}\mid \ spam)$	$= 0.65$
$p(\text{indicator \#2}\mid \neg spam)$	$= 0.15$
$p(\text{indicator \#3}\mid \ spam)$	$= 0.75$
$p(\text{indicator \#3}\mid \neg spam)$	$= 0.25$

A new email arrives: it contains a hyperlink and comes from an unknown sender, but it does **not** contains the word “free”. Compute the probability that this email is spam.

[Solution: With a Naive Bayes Model scenario, we have that:

$$p(Cause|Effect_1, ..., Effect_n) = \alpha\, p(Cause) * \prod_{i=1}^{n} p(Effect_i|Cause).$$

for some normalization constant α. Thus, we have:

$$\begin{aligned} p(\ spam|\text{ind.\#1 false, ind.\#2 true, ind.\#3 true}) &= \alpha * 0.25 * (1 - 0.55) * 0.65 * 0.75 &&= \alpha * 0.0548437 \\ p(\neg spam|\text{ind.\#1 false, ind.\#2 true, ind.\#3 true}) &= \alpha * (1 - 0.25) * (1 - 0.05) * 0.15 * 0.25 &&= \alpha * 0.0267187 \end{aligned}$$

Thus, after normalization:

$$\begin{aligned} p(\ spam|\text{ind.\#1 false, ind.\#2 true, ind.\#3 true}) &= 0.6724137931034483 \\ p(\neg spam|\text{ind.\#1 false, ind.\#2 true, ind.\#3 true}) &= 0.3275862068965517 \end{aligned}$$

]

[1] The data here are pure fantasy and are not supposed to correspond to the behaviour of any real spam filter.

8

(*a*) Describe as Pseudo-Code the Recursive Depth-first Tree-based And-Or-Search procedure for returning a conditional plan with non-deterministic actions.

[Solution:

function AND-OR-SEARCH(*problem*) **returns** a conditional plan, or *failure*
  return OR-SEARCH(*problem*, *problem*.INITIAL, [])

function OR-SEARCH(*problem*, *state*, *path*) **returns** *a conditional plan, or failure*
  if *problem*.IS-GOAL(*state*) **then return** the empty plan
  if IS-CYCLE(*state*, *path*) **then return** *failure*
  for each *action* **in** *problem*.ACTIONS(*state*) **do**
    plan ← AND-SEARCH(*problem*, RESULTS(*state*, *action*), [*state*] + [*path*])
    if *plan* ≠ *failure* **then return** [*action*] + [*plan*]
  return *failure*

function AND-SEARCH(*problem*, *states*, *path*) **returns** *a conditional plan, or failure*
  for each s_i **in** *states* **do**
    $plan_i$ ← OR-SEARCH(*problem*, s_i, *path*)
    if $plan_i$ = *failure* **then return** *failure*
  return [**if** s_1 **then** $plan_1$ **else if** s_2 **then** $plan_2$ **else** ... **if** s_{n-1} **then** $plan_{n-1}$ **else** $plan_n$]

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

(*b*) If, for some current path and state, a recursive call to OR-Search returns “failure”, then this means that, starting from that state and path:

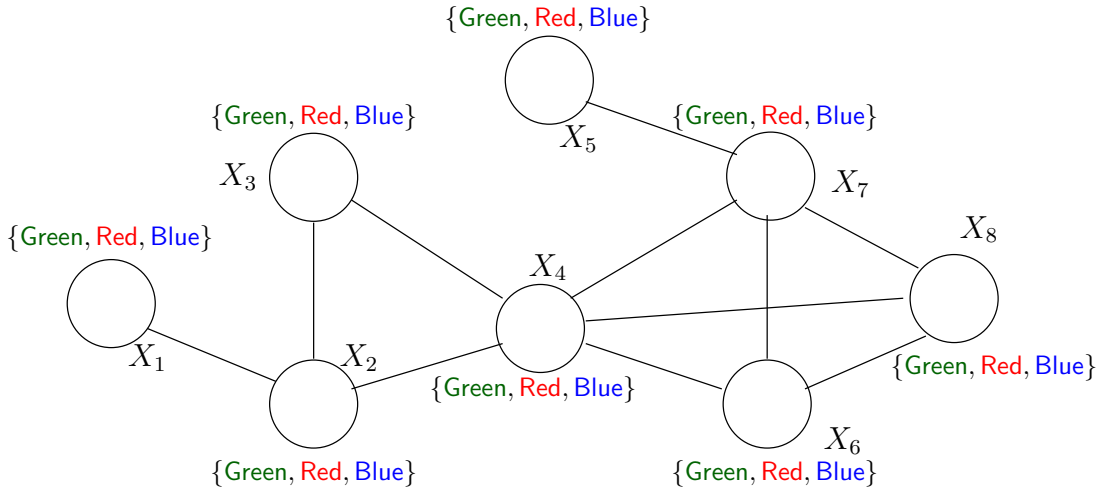
(i) there is no solution

(ii) if there is a non-cyclic solution, then it must be reachable from the earlier incarnation of the current state

[Solution: (ii)]

9

Consider the following constraint graph of a map coloring problem, with domain $D \stackrel{\text{def}}{=} \{\text{Green}, \text{Red}, \text{Blue}\}$.



Repeat the following steps:

1. Pick a node and a value according to the following criteria:
   use the Minimum Remaining Values (MRV) heuristic; in suborder, as a tie breaker, use the degree (DH) heuristic; in suborder, as a tie breaker, pick the leftmost node.

   Pick a value according to the Least Constraining Value (LCS) heuristic. In suborder, as a tie breaker, use the following order: {Green, Red, Blue}.

2. Apply the formal Forward Checking procedure.

until either you find one solution, or some node has an empty domain.
At every step, show the domain of the nodes X_i whose domain has been updated.

[Solution:

1. All nodes have the same domain, thus by MRV pick all nodes and by DH pick X_4;
   All values are equally constraining, thus by LCS pick all values, thus set $X_4 = \{\text{Green}, \ , \ \}$.

2. By forward checking, we have that $X_2 = X_3 = X_6 = X_7 = X_8 = \{\ , \text{Red}, \text{Blue}\}$.

There is still no solution and no empty-domain node.

1. By MRV pick $\{X_2, X_3, X_6, X_7, X_8\}$, then by DH pick X_7;
   By LCS pick both values in $\{\ , \text{Red}, \text{Blue}\}$, thus set $X_7 = \{\ , \text{Red}, \ \}$.

2. By forward checking, we have that $X_5 = \{\text{Green}, \ , \text{Blue}\}$ and $X_6 = X_8 = \{\ , \ , \text{Blue}\}$.

There is still no solution and no empty-domain node.

1. By MRV pick $\{X_6, X_8\}$, then by DH pick $\{X_6, X_8\}$, thus pick the leftmost: X_6;
   By LCS pick Blue, thus we set $X_6 = \{\ , \ , \text{Blue}\}$.

2. By forward checking, we have that $X_8 = \{\ , \ , \ \}$.

]

Notation: to represent the current domain of a node X_i, substitute with a blank “ ” any value in {Green, Red, Blue} which cannot be assigned.

10

Given the following set of propositional clauses Γ:

$$
\begin{array}{l}
(\ \ F \vee \neg B \vee \ \ G) \\
(\ \ C \vee \neg D) \\
(\neg D \vee \ \ G \vee \ \ B) \\
(\ \ F \vee \ \ C) \\
(\ \ D \vee \ \ H \vee \neg I) \\
(\neg C \vee \neg D \vee \ \ E) \\
(\neg F \vee \neg C) \\
(\ \ C \vee \neg H \vee \ \ I) \\
(\ \ F \vee \neg C \vee \ \ D) \\
(\ \ F \vee \neg C \vee \neg G) \\
(\neg F)
\end{array}
$$

Produce a PL-resolution proof that Γ is unsatisfiable, based on unit-resolution only.
Such proof must be written as a sequence of resolution steps in the form:

$[Clause_{11},\ Clause_{12}] \Longrightarrow Resolvent\_Clause_1$;
$[Clause_{21},\ Clause_{22}] \Longrightarrow Resolvent\_Clause_2$;
...;
$[Clause_{k1},\ Clause_{k2}] \Longrightarrow Resolvent\_Clause_k$;

s.t. $Resolvent\_Clause_k$ is the empty clause, and each $Clause_{ij}$ is either in Γ or is a resolvent clause $Resolvent\_Clause_m$ resulting from previous steps, i.e. s.t. $m < i$.
WRITE THE CLAUSES EXPLICITLY, NOT SIMPLY THEIR INDEXES!!

[Solution: Γ can be proved unsatisfiable by unit-resolution steps only:
$[(\neg F),\ (\ \ F \vee \neg B \vee \ \ G)] \Longrightarrow (\neg B \vee \ \ G)$;
$[(\neg F),\ (\ \ F \vee \ \ C)] \Longrightarrow (\ \ C)$;
$[(\neg F),\ (\ \ F \vee \neg C \vee \ \ D)] \Longrightarrow (\neg C \vee \ \ D)$;
$[(\neg F),\ (\ \ F \vee \neg C \vee \neg G)] \Longrightarrow (\neg C \vee \neg G)$;

$[(\ \ C),\ (\neg C \vee \neg D \vee \ \ E)] \Longrightarrow (\neg D \vee \ \ E)$;
$[(\ \ C),\ (\neg C \vee \ \ D)] \Longrightarrow (\ \ D)$;
$[(\ \ C),\ (\neg C \vee \neg G)] \Longrightarrow (\neg G)$;

$[(\ \ D),\ (\neg D \vee \ \ G \vee \ \ B)] \Longrightarrow (\ \ G \vee \ \ B)$;
$[(\ \ D),\ (\neg D \vee \ \ E)] \Longrightarrow (\ \ E)$;

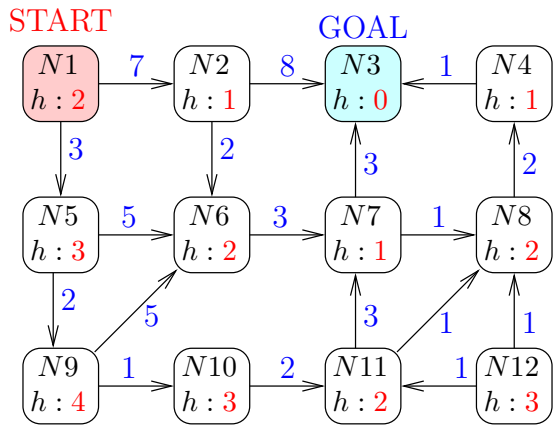
$[(\neg G),\ (\neg B \vee \ \ G)] \Longrightarrow (\neg B)$;
$[(\neg G),\ (\ \ G \vee \ \ B)] \Longrightarrow (\ \ B)$;

$[(\neg B),\ (\ \ B)] \Longrightarrow ()$;]

11

Apply A* to the following search problem, where

- the costs of the actions is denoted on the edges;
- the heuristic function is the Manhattan distance to 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 $h(n) = 3$);
- in case of same-score nodes, the oldest one is extracted from the frontier.



(a) **[10 pt. /100]** Is the heuristic function admissible? (Motivate your answer).

(b) **[90 pt. /100]** For each step of the execution, report:

- The node extracted from the priority queue.
- The nodes which are added to the priority queue *after* expanding.

Then report the path returned by A*.

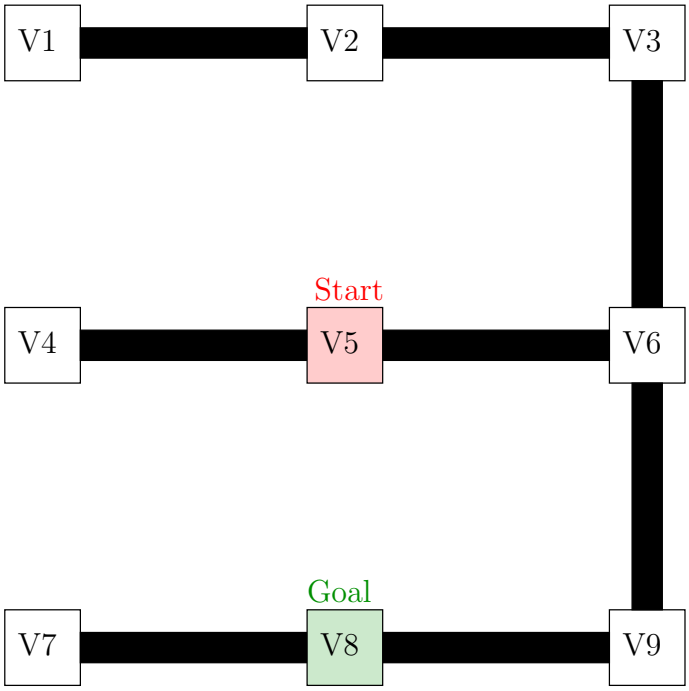
[Solution:

Extracted node	Nodes which are added to the priority queue after expanding
N1 (0+2)	N2 (1+7); N5 (3+3)
N5 (3+3)	N6 (8+2); N9 (5+4)
N2 (1+7)	N3 (15+0); (note: N6 (9+2) is not added)
N9 (5+4)	N10 (6+3); (note: N6 (10+2) is not added) ;
N10 (6+3)	N11 (8+2);
N6 (8+2)	N7 (11+1);
N11 (8+2)	N8 (9+2); (note: N7 (11+1)is not added);
N8 (9+2)	N4 (11+1);
N7 (11+1)	N3 (14+0); (note: better than N3 (15+0))
N4 (11+1);	N3 (12+0);
N3 (12+0);	Goal reached with optimum path!

Path: N1 , N5 , N9 , N10 , N11 , N8 , N4 , N3]

12

In the following state graph, apply *online DFS* and report the sequence of visited states, including repetitions (e.g. “$V1 \rightarrow V2 \rightarrow V3 \Rightarrow V2$ ... ”.).
The start if V5 , the goal is V8 .
The order of (untried) actions is $[up, left, down, right]$.



Important: *in the reported sequence, you must distinguish exploration steps from backtracking steps.*

- *mark as a single-line arrow “$\rightarrow$” an exploration step*
- *mark as a double-line arrow “$\Rightarrow$” a backtracking step.*

For instance, the sequence “$V1 \rightarrow V2 \rightarrow V3 \Rightarrow V2$” means:
“Pass from V1 to V2 and then to V3 via exploration steps, and then backtrack to V2 ”.
[Solution:

$V5 \rightarrow V4 \rightarrow V5 \rightarrow V6 \rightarrow V3 \rightarrow V2 \rightarrow V1 \rightarrow V2 \rightarrow V3 \rightarrow V6 \rightarrow V5 \Rightarrow V6 \rightarrow V9 \rightarrow V6 \Rightarrow V9 \rightarrow V8$

]

13

(a) Describe as Pseudo-Code the CSP Min-Conflicts local-search heuristic procedure.
[Solution:

function MIN-CONFLICTS(csp, max_steps) **returns** a solution or failure
inputs: csp, a constraint satisfaction problem
max_steps, the number of steps allowed before giving up

$current \leftarrow$ an initial complete assignment for csp
for i = 1 to max_steps **do**
if $current$ is a solution for csp **then return** $current$
$var \leftarrow$ a randomly chosen conflicted variable from csp.VARIABLES
$value \leftarrow$ the value v for var that minimizes CONFLICTS($var, v, current, csp$)
set $var = value$ in $current$
return *failure*

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

(b) If the procedure fails to find a satisfying solution, this means that:

1 the input CSP problem has no solution

2 we can conclude nothing about the existence of a solution for the input CSP problem.

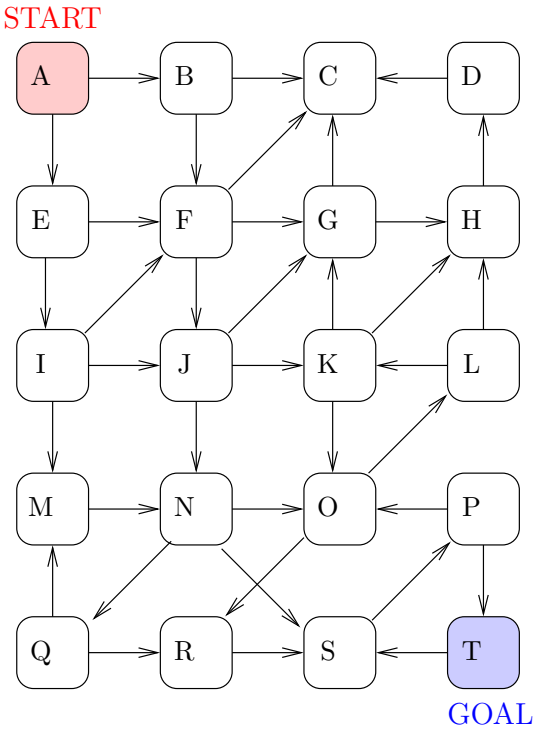
(Say if 1. or 2.)

[Solution: 2.]

14

In the following state graph, apply *breadth-first search* (graph version) and report for each step:

- the node extracted from the frontier;
- the nodes added to the frontier in the current step.



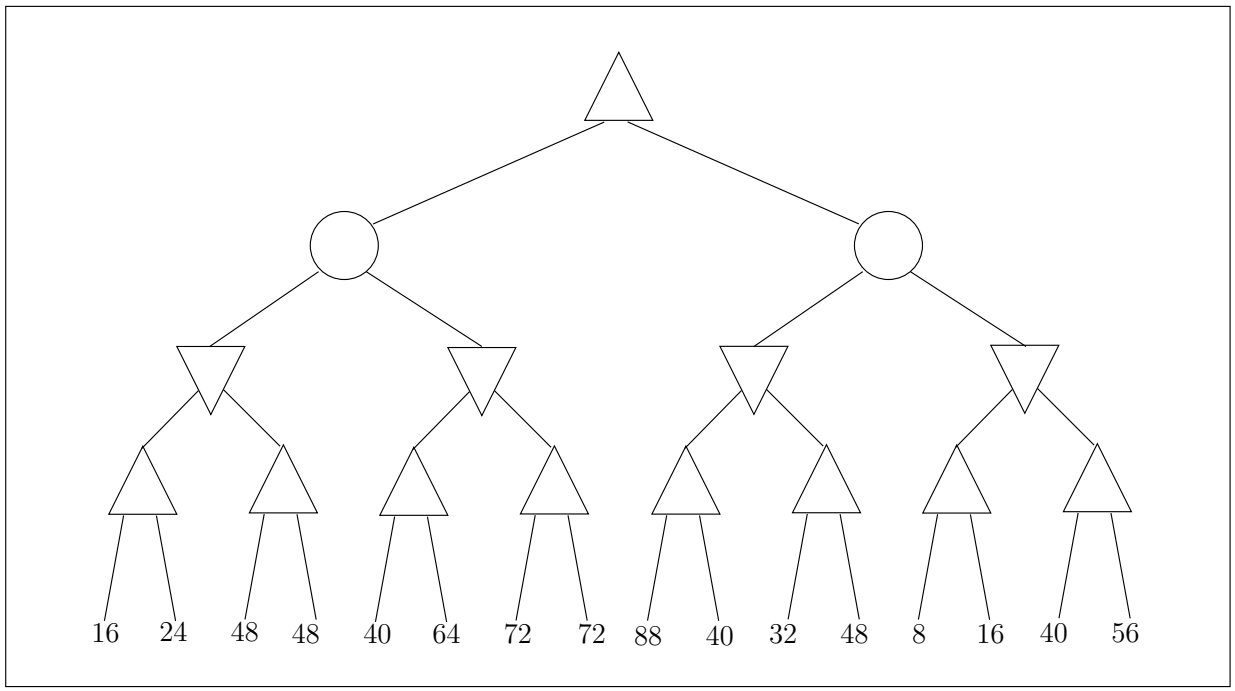
Actions are sorted according to the (ascending) alphabetical order of the destination.

[Solution:

A	[B, E	]
B	[C, F	]
E	[I	]
C	[	]
F	[G, J	]
I	[M	]
G	[H	]
J	[K, N	]
M	[	]
H	[D	]
K	[O	]
N	[Q, S	]
D	[	]
O	[L, R	]
Q	[	]
S	[P	]
L	[	]
R	[	]
P	[T	] Goal reached! (early goal test)

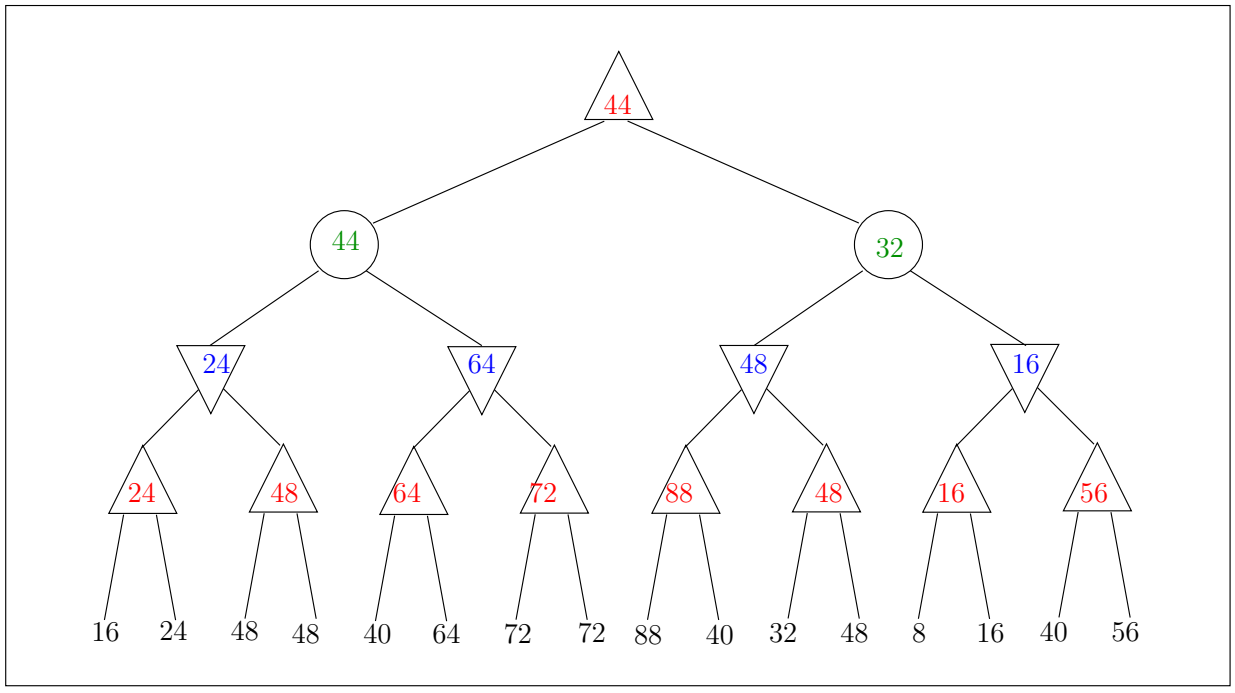
]

15



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

[Solution:



]