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

Prof. Roberto Sebastiani
DISI, Università di Trento, Italy

2026.09.08

6305186

[COPY WITH SOLUTIONS]

1

Let

- $h(n)$ be a heuristic function representing the estimated cost from a node n to the solution,
- $h^*(n)$ represent the true cost from a node n to the solution,
- $c(n, a, n')$ represent the cost of passing from node n top node n' via action a.

For each of the following facts about heuristic search and A^*, say if it is true or false.

(a) $h(n)$ is admissible if and only if $h(n) \leq c(n, a, n') + h(n')$ for every successor n' of n generated by any action a
[Solution: false. (This is a definition of monotonic, s.t. the “only if” part is wrong..)]

(b) $h(n)$ is consistent (aka monotonic) if and only if $h(n) = h^*(n)$.
[Solution: false. It is monotonic if and only if $h(n) \leq c(n, a, n') + h(n')$ for every successor n' of n generated by any action a.]

(c) if $h(n)$ is admissible, then A^* tree search is optimal
[Solution: true]

(d) if $h(n)$ is consistent (aka monotonic), A^* tree graph is optimal
[Solution: true]

2

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

(*a*) $((A \wedge B) \rightarrow C)$ is equivalent to $(A \rightarrow (B \rightarrow C))$
[Solution: True]

(*b*) $((A \wedge B) \rightarrow C$ is equivalent to $(A \rightarrow C) \wedge (B \rightarrow C))$
[Solution: False]

(*c*) $((A \leftrightarrow B) \leftrightarrow C)$ is valid if and only if $(\neg A \leftrightarrow (\neg B \leftrightarrow \neg C))$ is unsatisfiable
[Solution: True]

(*d*) $((A \leftrightarrow B) \leftrightarrow C)$ is satisfiable if and only if $(\neg A \leftrightarrow (\neg B \leftrightarrow \neg C))$ is unsatisfiable
[Solution: False]

3

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 $\forall x.((x < 1) \wedge (x > 2))$ is unsatisfiable.
[Solution: false (in FOL “>”, “<”, “1” and “2” have no fixed interpretation)]

(b) The FOL formula $(2 < 4)$ is valid.
[Solution: false (in FOL “>”, “<”, “1” and “2” have no fixed interpretation)]

(c) The FOL formula $(\exists x_1 \forall x_2.P(x_1, x_2)) \rightarrow (\forall x_2 \exists x_1.P(x_1, x_2))$ is valid.
[Solution: true]

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

4

Consider (normal) modal logics. Let $\mathsf{IsOn}(\mathsf{Light}), \mathsf{IsRunning}(\mathsf{Pump})$ be possible facts, let $Ada, Chris$ be agents and let $\mathbf{K}_{Ada}, \mathbf{K}_{Chris}$ denote the modal operators “Ada knows that...” and “Chris knows that...” respectively.
For each of the following facts, say if it is true or false.

(*a*) If $\neg\mathbf{K}_{Ada}(\mathsf{IsOn}(\mathsf{Light}) \wedge \mathsf{IsRunning}(\mathsf{Pump}))$ holds, then $\neg\mathbf{K}_{Ada}\mathsf{IsOn}(\mathsf{Light}) \wedge \neg\mathbf{K}_{\mathsf{Ada}}\mathsf{IsRunning}(\mathsf{Pump})$ holds
[Solution: false]

(*b*) If $\neg\mathbf{K}_{Ada}\mathsf{IsOn}(\mathsf{Light}) \wedge \neg\mathbf{K}_{\mathsf{Ada}}\mathsf{IsRunning}(\mathsf{Pump})$ holds, then $\neg\mathbf{K}_{Ada}(\mathsf{IsOn}(\mathsf{Light}) \wedge \mathsf{IsRunning}(\mathsf{Pump}))$ holds
[Solution: true]

(*c*) If $\mathsf{IsOn}(\mathsf{Light}) \leftrightarrow \mathsf{IsRunning}(\mathsf{Pump})$ and $\mathbf{K}_{Chris}\mathsf{IsOn}(\mathsf{Light})$ hold, then $\mathbf{K}_{Chris}\mathsf{IsRunning}(\mathsf{Pump})$ holds
[Solution: false]

(*d*) If $\mathbf{K}_{Ada}(\mathsf{IsOn}(\mathsf{Light}) \rightarrow \mathbf{K}_{\mathsf{Chris}}\mathsf{IsRunning}(\mathsf{Pump}))$ and $\mathbf{K}_{Ada}\mathsf{IsOn}(\mathsf{Light})$ hold, then $\mathbf{K}_{Ada}\mathbf{K}_{Chris}\mathsf{IsRunning}(\mathsf{Pump}))$ holds
[Solution: true]

5

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

(a) If A and B are conditionally independent given C, then $\mathbf{P}(A \mid B, C) = \mathbf{P}(A \mid C)$.
[Solution: True:
$\mathbf{P}(A \mid B, C) = \dfrac{\mathbf{P}(A, B, C)}{\mathbf{P}(B, C)} = \dfrac{\mathbf{P}(A, B \mid C)\mathbf{P}(C)}{\mathbf{P}(B \mid C)\mathbf{P}(C)} = \dfrac{\mathbf{P}(A \mid C)\mathbf{P}(B \mid C)}{\mathbf{P}(B \mid C)} = \mathbf{P}(A \mid C).$]

(b) $\mathbf{P}(A, B \mid C) = \mathbf{P}(A \mid B, C)\,\mathbf{P}(B \mid C)\,\mathbf{P}(C)$.
[Solution: False: by the chain rule, $\mathbf{P}(A, B \mid C) = \mathbf{P}(A \mid B, C)\,\mathbf{P}(B \mid C)$.]

(c) If A and B are conditionally independent given C, and all the probabilities involved are strictly greater than 0, then $\dfrac{\mathbf{P}(A, C)\,\mathbf{P}(B, C)}{\mathbf{P}(A, B \mid C)} = \mathbf{P}(C)^2$.
[Solution: True: since A and B are conditionally independent given C, then
$\mathbf{P}(A, B \mid C) = \mathbf{P}(A \mid C)\,\mathbf{P}(B \mid C)$.
Also $\mathbf{P}(A, C) = \mathbf{P}(A \mid C)\,\mathbf{P}(C)$ and $\mathbf{P}(B, C) = \mathbf{P}(B \mid C)\,\mathbf{P}(C)$. Thus:

$$\frac{\mathbf{P}(A, C)\,\mathbf{P}(B, C)}{\mathbf{P}(A, B \mid C)} = \frac{\mathbf{P}(A \mid C)\,\mathbf{P}(C) \cdot \mathbf{P}(B \mid C)\,\mathbf{P}(C)}{\mathbf{P}(A \mid C)\,\mathbf{P}(B \mid C)} = \mathbf{P}(C)^2.$$

]

(d) Without any additional assumption, $\mathbf{P}(A, B \mid C) = \mathbf{P}(B \mid A, C)\,\mathbf{P}(A \mid C)$.
[Solution: True:

$$\mathbf{P}(B \mid A, C)\,\mathbf{P}(A \mid C) = \frac{\mathbf{P}(A, B, C)}{\mathbf{P}(A, C)} \cdot \frac{\mathbf{P}(A, C)}{\mathbf{P}(C)} = \frac{\mathbf{P}(A, B, C)}{\mathbf{P}(C)} = \mathbf{P}(A, B \mid C).$$

]

6

(*a*) Describe as Pseudo-Code the recursive DPLL SAT procedure.
[Solution:

function DPLL-SATISFIABLE?(s) **returns** *true* or *false*
inputs: s, a sentence in propositional logic

$clauses \leftarrow$ the set of clauses in the CNF representation of s
$symbols \leftarrow$ a list of the proposition symbols in s
return DPLL($clauses, symbols, \{\}$)

---

function DPLL($clauses, symbols, model$) **returns** *true* or *false*

if every clause in $clauses$ is true in $model$ **then return** *true*
if some clause in $clauses$ is false in $model$ **then return** *false*
$P, value \leftarrow$ FIND-PURE-SYMBOL($symbols, clauses, model$)
if P is non-null **then return** DPLL($clauses, symbols - P, model \cup \{P{=}value\}$)
$P, value \leftarrow$ FIND-UNIT-CLAUSE($clauses, model$)
if P is non-null **then return** DPLL($clauses, symbols - P, model \cup \{P{=}value\}$)
$P \leftarrow$ FIRST($symbols$); $rest \leftarrow$ REST($symbols$)
return DPLL($clauses, rest, model \cup \{P{=}true\}$) **or**
DPLL($clauses, rest, model \cup \{P{=}false\}$))

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

(*b*) Among the following sentences about this algorithm, only one is true. Say which one.

1 It performs non-deterministic steps before deterministic ones.

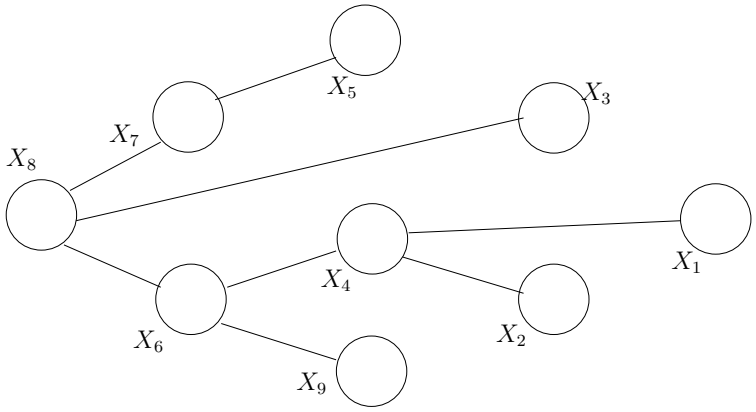
2 It requires polynomial space in worst-case.

3 It is less efficient than PL-Resolution

[Solution: 2.]

7

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:



$$
\begin{array}{lcccc}
X_1 = \{ & \text{Red} & , & & \} \\
X_2 = \{ & & , \text{Green} & , & \} \\
X_3 = \{ & & , & , \text{Blue} & \} \\
X_4 = \{ & \text{Red} & , \text{Green} & , \text{Blue} & \} \\
X_5 = \{ & & , & , \text{Blue} & \} \\
X_6 = \{ & \text{Red} & , \text{Green} & , \text{Blue} & \} \\
X_7 = \{ & & , \text{Green} & , \text{Blue} & \} \\
X_8 = \{ & \text{Red} & , \text{Green} & , \text{Blue} & \} \\
X_9 = \{ & & , \text{Green} & , & \}
\end{array}
$$

Using the Tree-Structured Algorithm, and considering the following node ordering: $\{X_8, X_7, X_5, X_3, X_6, X_9, X_4, X_2, X_1\}$.
Note: solving the problem by using any other algorithm will be considered incorrect.

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

[Solution:

Node :	*Operation* :	*Updated*	*Domain* :			
X_1 :	make-arc-consistent $\langle X_4, X_1 \rangle$:	$X_4 =$	{	, Green	, Blue	}
X_2 :	make-arc-consistent $\langle X_4, X_2 \rangle$:	$X_4 =$	{	,	, Blue	}
X_4 :	make-arc-consistent $\langle X_6, X_4 \rangle$:	$X_6 =$	{Red	, Green	,	}
X_9 :	make-arc-consistent $\langle X_6, X_9 \rangle$:	$X_6 =$	{Red	,	,	}
X_6 :	make-arc-consistent $\langle X_8, X_6 \rangle$:	$X_8 =$	{	, Green	, Blue	}
X_3 :	make-arc-consistent $\langle X_8, X_3 \rangle$:	$X_8 =$	{	, Green	,	}
X_5 :	make-arc-consistent $\langle X_7, X_5 \rangle$:	$X_7 =$	{	, Green	,	}
X_7 :	make-arc-consistent $\langle X_8, X_7 \rangle$:	$X_8 =$	{	,	,	} FAILURE!

]

(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 not solvable, because it produces deterministically an empty domain for X_8.]

(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: $X_5 : \{\ ,\ , \text{Blue}\}$)

8

Consider the following Horn formula in PL:

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

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: $N,\ D,\ E, \neg G,\ F$ (or, equivalently, $N,\ D,\ E,\ F, \neg G$).
The resulting formula is:

$$
\begin{array}{llll}
(\ I & \vee \neg B & \vee \neg C) & \wedge \\
(\neg B & \vee \ M & \vee \neg C) & \wedge
\end{array}
$$

$$
\begin{array}{llll}
(\neg B & \vee \ A & \vee \neg H) & \wedge \\
(\neg I & \vee \ B & \vee \neg A) & \wedge
\end{array}
$$

$$
\begin{array}{llll}
(& \vee \ H & \vee \neg M) & \wedge
\end{array}
$$

$$
\begin{array}{llll}
(& \vee \neg B & \vee \ C) &
\end{array}
$$

(ii) the resulting formula contains no empty clause, thus the original formula is satisfiable because it is possible to assign all other unassigned atoms to $\perp$ since the formuila is Horn.

(b) the model is $\{\ N,\ D,\ E, \neg G,\ F\} \cup \{\neg L, \neg M, \neg H, \neg I, \neg B, \neg C, \neg A\}$

]

9

Let $P()$, $Q()$, $R()$, $S()$, $T()$, $U()$, $V()$, $W()$, $X()$, $Y()$, $Z()$, denote predicates, a, b, c, d, e denote constants. Consider the following set of clauses:

$$
\begin{array}{l}
W(b) \\
\neg S(b) \\
\neg T(e) \vee S(b) \\
\neg S(b) \vee P(a) \\
U(d) \vee T(e) \\
\neg U(d) \vee X(c) \\
\neg P(a) \vee S(b) \\
P(a) \vee Y(d) \\
\neg R(a) \vee S(b) \vee T(e) \\
\neg Q(c) \vee T(e) \vee \neg U(d) \\
R(a) \vee S(b) \vee V(a) \\
R(a) \vee Q(c) \vee S(b)
\end{array}
$$

Build a refutation using the Hyper-Resolution strategy.

[Solution: The Hyper-Resolution strategy works by always resolving one electron (a clause with positive literals only) with a nucleus (a clause with at least one negative literal). Thus one possible refutation is:

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{R(a) \vee Q(c) \vee S(b) \quad \neg R(a) \vee S(b) \vee T(e)}{Q(c) \vee S(b) \vee T(e)} \quad \neg Q(c) \vee T(e) \vee \neg U(d)}{S(b) \vee T(e) \vee \neg U(d)} \quad U(d) \vee T(e)}{S(b) \vee T(e)} \quad \neg T(e) \vee S(b)}{S(b)} \quad \neg S(b)}{\bot}
$$

This can be merged into one single step:

$$
\dfrac{R(a) \vee Q(c) \vee S(b) \quad \neg R(a) \vee S(b) \vee T(e) \quad \neg Q(c) \vee T(e) \vee \neg U(d) \quad U(d) \vee T(e) \quad \neg T(e) \vee S(b) \quad \neg S(b)}{\bot}
$$

]

10

Given:

- a set of basic concepts: {Person, Man, Woman, Cat, Dog, Male, Female, Doctor, Engineer}
- a set of relations: {hasChild, hasPet}

with their standard English meaning (“hasChild” refers also to animals).
Write a $\mathcal{T}$-box in $\mathcal{ALCN}$ description logic defining the following concepts.

(a) C_1: a man with at least 3 cats or at least 3 dogs
[Solution:
$C_1 \equiv \mathsf{Man} \sqcap ((\geq 3)\mathsf{hasPet}.\mathsf{Cat}) \sqcup ((\geq 3)\mathsf{hasPet}.\mathsf{Dog})$
or any $\mathcal{T}$-box which is logically equivalent to it]

(b) C_2: a woman with at least 3 cats or dogs
[Solution:
$C_2 \equiv \mathsf{Woman} \sqcap (\geq 3)\mathsf{hasPet}.(\mathsf{Cat} \sqcup \mathsf{Dog})$
or any $\mathcal{T}$-box which is logically equivalent to it]

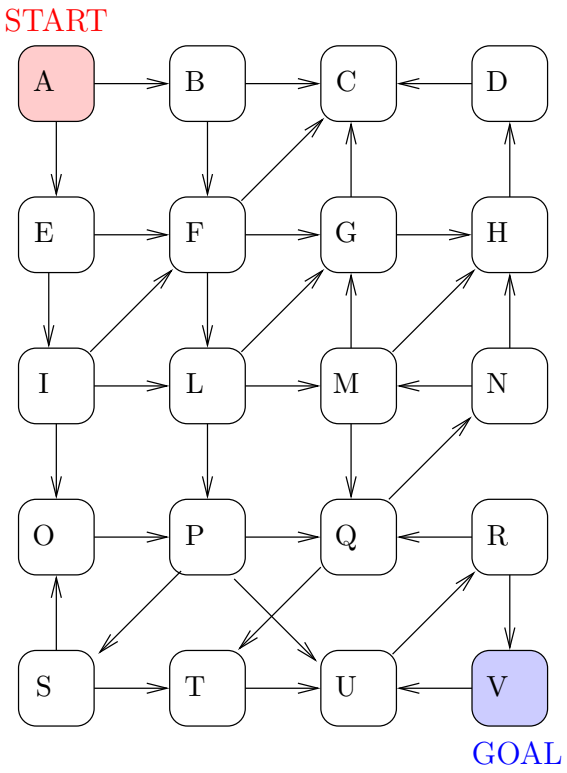
(c) C_3: a man with no female doctor children
[Solution:
$C_3 \equiv \mathsf{Man} \sqcap \neg\exists.\mathsf{hasChild}.(\mathsf{Female} \sqcap \mathsf{Doctor})$
or any $\mathcal{T}$-box which is logically equivalent to it]

(d) C_4: a woman whose children are all engineers or doctors
[Solution:
$C_4 \equiv \mathsf{Woman} \sqcap \forall\mathsf{hasChild}.(\mathsf{Engineer} \sqcup \mathsf{Doctor})$
or any $\mathcal{T}$-box which is logically equivalent to it]

11

A In the following state graph, apply *depth-first search* (graph version, with loop detection) and report for each step:

- the node extracted from the frontier;
- the nodes in the frontier after the step, listed from the oldest to the newest.



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

[Solution:

	$[\ A$	$]$
A	$[\ B, E$	$]$
E	$[\ B, F, I$	$]$
I	$[\ B, F, L, O$	$]$
O	$[\ B, F, L, P$	$]$
P	$[\ B, F, L, Q, S, U$	$]$
U	$[\ B, F, L, Q, S, R$	$]$
R	$[\ B, F, L, Q, S, Q, V$	$]$ Goal reached! (early goal test)

]

12

(a) Describe as Pseudo-Code the specialized solving procedure for tree-structured CSPs.
[Solution:

function TREE-CSP-SOLVER(csp) **returns** a solution, or failure
  inputs: csp, a CSP with components X, D, C

  $n \leftarrow$ number of variables in X
  $assignment \leftarrow$ an empty assignment
  $root \leftarrow$ any variable in X
  $X \leftarrow$ TOPOLOGICALSORT(X, $root$)
  for j = n **down to** 2 **do**
    MAKE-ARC-CONSISTENT(PARENT(X_j), X_j)
    if it cannot be made consistent **then return** $failure$
  for i = 1 **to** n **do**
    $assignment[X_i] \leftarrow$ any consistent value from D_i
    if there is no consistent value **then return** $failure$
  return $assignment$

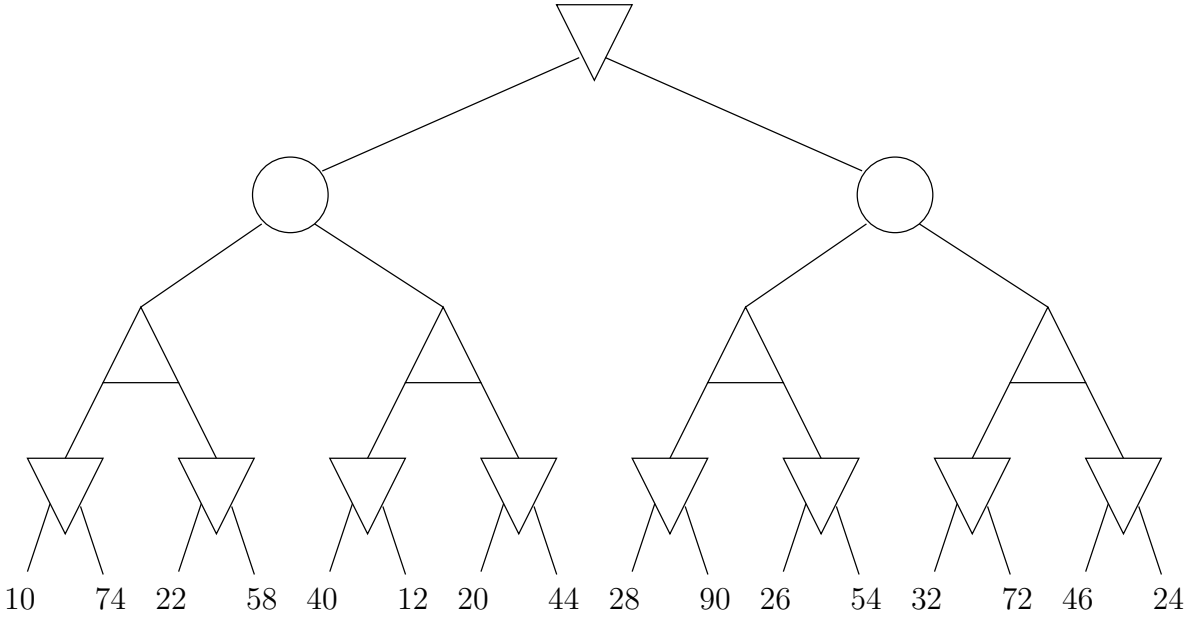
or any schema equivalent to the above one (from AIMA book).]

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

- It requires polynomial time in worst-case if the input constraint graph has no loops.

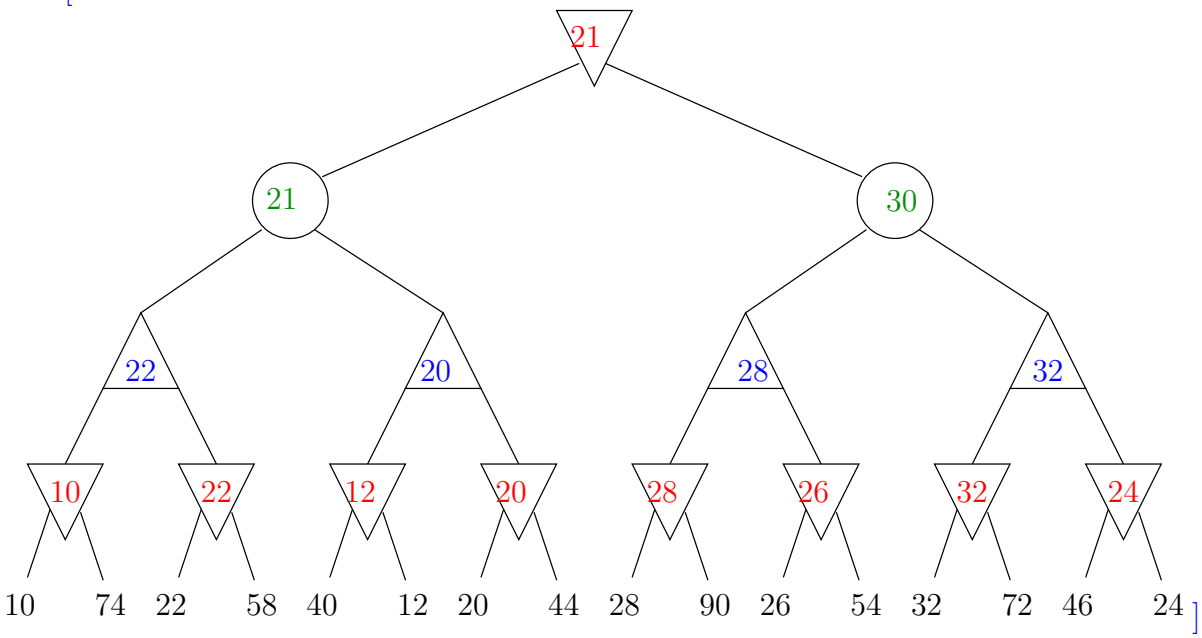
[Solution: True. It requires $O(nd^2)$ steps in worst case.]

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.

[Solution:

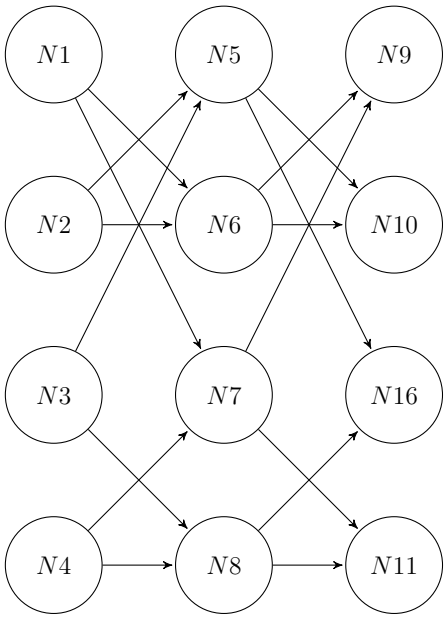


]

14

Consider the following actions with durations / dependencies:

a	duration(a)
N1	30
N2	10
N3	10
N4	10
N5	20
N6	30
N7	30
N8	20
N9	30
N10	20
N16	10
N11	20



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
N1	0	0
N2	0	20
N3	0	40
N4	0	20
N5	10	50
N6	30	30
N7	30	30
N8	10	50
N9	60	60
N10	60	70
N16	30	80
N11	60	70

Actions in CP: N1, N6, N7, N9
Makespan: 90]

15

The Happyness Corporation wants to find sad people and help them. They know from historical data that 18% of the people are sad. To identify who is sad, they rely on three indicators:

Indicator #1: the person is married;
Indicator #2: the person is rich;
Indicator #3: the person has a cat.

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

$p(\text{indicator \#1} \mid \quad sad)$	$= 0.60$
$p(\text{indicator \#1} \mid \neg sad)$	$= 0.70$
$p(\text{indicator \#2} \mid \quad sad)$	$= 0.10$
$p(\text{indicator \#2} \mid \neg sad)$	$= 0.70$
$p(\text{indicator \#3} \mid \quad sad)$	$= 0.10$
$p(\text{indicator \#3} \mid \neg sad)$	$= 0.80$

Consider a person who has a cat, but is **not** married and is **not** rich. Compute the probability that this person is sad.

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

$$p(Cause \mid Effect_1, \ldots, Effect_n) = \alpha\, p(Cause) \prod_{i=1}^{n} p(Effect_i \mid Cause),$$

for some normalization constant α. Thus, we have:

$$\begin{aligned} p(\quad sad \mid \text{ind.\#1 false}, \text{ind.\#2 false}, \text{ind.\#3 true}) &= \alpha * 0.18 * (1 - 0.60) * (1 - 0.10) * 0.10 \\ &= \alpha * 0.00648 \\ p(\neg sad \mid \text{ind.\#1 false}, \text{ind.\#2 false}, \text{ind.\#3 true}) &= \alpha * (1 - 0.18) * (1 - 0.70) * (1 - 0.70) * 0.80 \\ &= \alpha * 0.05904 \end{aligned}$$

Thus, after normalization:

$$\begin{aligned} p(\quad sad \mid \text{ind.\#1 false}, \text{ind.\#2 false}, \text{ind.\#3 true}) &= \quad 0.0989010989010989 \\ p(\neg sad \mid \text{ind.\#1 false}, \text{ind.\#2 false}, \text{ind.\#3 true}) &= \quad 0.9010989010989011 \end{aligned}$$

]

[1] The data here are pure fantasy and have no correspondence with any real-world study.