Course “**Fundamentals of Artificial Intelligence**”
EXAM TEXT

Prof. Roberto Sebastiani
DISI, Università di Trento, Italy

2026.06.10

6305186

[COPY WITH SOLUTIONS]

1

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

(a) Shortest-path search problems adopt an *atomic* representation of states.
[Solution: True]

(b) Propositional-logic problems adopt an *atomic* representation of states.
[Solution: False. They adopt a *factored* representation.]

(c) CSP problems adopt a *factored* representation of states.
[Solution: True]

(d) First-order-logic problems adopt an *structured* representation of states.
[Solution: True]

2

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 $h(n) = h^*(n)$.
[Solution: false. It is admissible if and only if $h(n) \leq h^*(n)$]

(b) $h(n)$ 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
[Solution: true]

(c) if $h(n)$ is admissible, then $h(n)$ is monotonic, but the vice versa does not necessarily hold
[Solution: false]

(d) if $h(n)$ is monotonic, then $h(n)$ is admissible, but the vice versa does not necessarily hold
[Solution: true]

3

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

(a) If $\mathbf{K}_{Ed}\mathsf{IsIdle}(\mathsf{Pump}) \vee \mathbf{K}_{\mathsf{Ed}}\mathsf{IsOn}(\mathsf{Light})$holds, then $\mathbf{K}_{Ed}(\mathsf{IsIdle}(\mathsf{Pump}) \vee \mathsf{IsOn}(\mathsf{Light}))$ holds
[Solution: true]

(b) $\mathbf{K}_{Ed}\mathsf{IsIdle}(\mathsf{Pump}) \wedge \mathbf{K}_{\mathsf{Ed}}\mathsf{IsOn}(\mathsf{Light})$ if and only if $\mathbf{K}_{Ed}(\mathsf{IsIdle}(\mathsf{Pump}) \wedge \mathsf{IsOn}(\mathsf{Light}))$ holds
[Solution: true]

(c) If $\mathbf{K}_{Eve}\mathsf{IsIdle}(\mathsf{Pump})$ and $\mathsf{IsIdle}(\mathsf{Pump}) \rightarrow \mathsf{IsOn}(\mathsf{Light})$ hold, then $\mathbf{K}_{Eve}\mathsf{IsOn}(\mathsf{Light})$ holds
[Solution: false]

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

4

Given the following symbols, representing concept, relation and individual names in the alien language of the remote planet Gronk, and the following $\mathcal{ALCQ}$ $\mathcal{T}$-box $\mathcal{T}$ and $\mathcal{A}$-box $\mathcal{A}$:

- a set of primitive $\mathcal{ALCQ}$ concept names: {Pero, Mall, Fell, Doto, Eger}
- a set of $\mathcal{ALCQ}$ relation names: {haCilo}
- a set of $\mathcal{ALCQ}$ individual names: {Ja, Ju, Mo, Do, Ad, Bi, Ev, Gl}

$\mathcal{T}$-box $\mathcal{T}$		
Pero		⟨*primitive concept*⟩
Fell		⟨*primitive concept*⟩
Mall		⟨*primitive concept*⟩
Doto		⟨*primitive concept*⟩
Eger		⟨*primitive concept*⟩
Wonm	$\equiv$	Pero $\sqcap$ Fell
Mnam	$\equiv$	Pero $\sqcap$ Mall
Moet	$\equiv$	Wonm $\sqcap$ $\exists$haCilo.Pero
Faet	$\equiv$	Mnam $\sqcap$ $\exists$haCilo.Pero
Part	$\equiv$	Pero $\sqcap$ haCilo.Pero
Gamr	$\equiv$	Moet $\sqcap$ haCilo.Part
Gafr	$\equiv$	Faet $\sqcap$ haCilo.Part
Gpar	$\equiv$	Part $\sqcap$ haCilo.Part

$\mathcal{A}$-box $\mathcal{A}$
Mo : Wonm; Ev : Wonm; Gl : Wonm; Ja : Wonm;
Ju : Mnam; Ad : Mnam; Bi : Mnam; Do : Mnam;
Ad : Doto; Mo : Doto
Ev : Eger; Gl : Eger; Ja : Eger
⟨Ju, Mo⟩ : haCilo; ⟨Ju, Ad⟩ : haCilo; ⟨Ju, Ev⟩ : haCilo;
⟨Mo, Gl⟩ : haCilo; ⟨Mo, Bi⟩ : haCilo; ⟨Mo, Do⟩ : haCilo; ⟨Mo, Ja⟩ : haCilo;

For each of the following $\mathcal{ALCQ}$ queries to $\mathcal{T} \cup \mathcal{A}$, say if it is true or false.

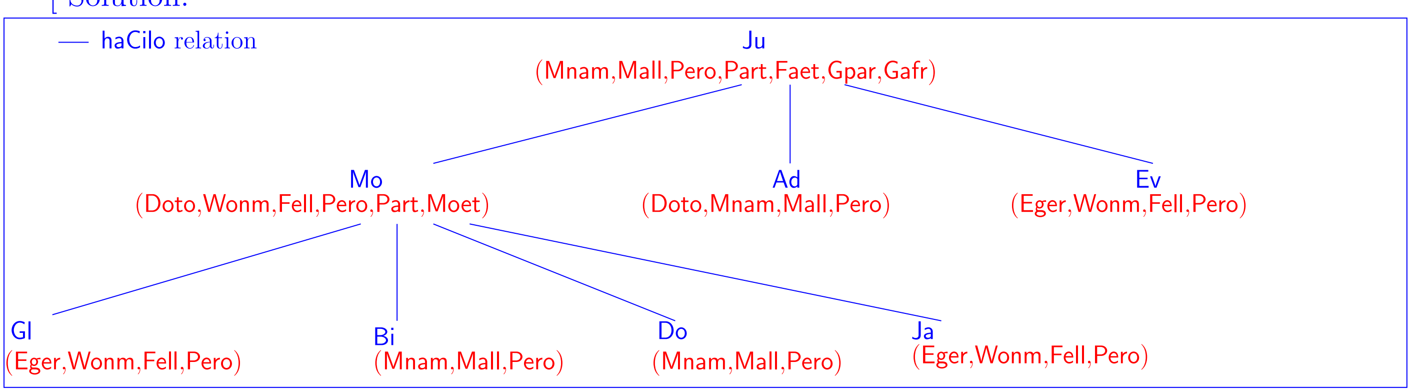
(a) Ju : Gafr $\sqcap$ $\exists$haCilo.Moet [Solution: true]

(b) Ju : Part $\sqcap$ (≤ 1)haCilo.Doto [Solution: false]

(c) Ju : Faet $\sqcap$ $\forall$haCilo.(Doto $\sqcup$ Eger) [Solution: true]

(d) Mo : Part $\sqcap$ (≥ 2)haCilo.Mall [Solution: true]

[Solution:



]

5

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

(*a*) If A and B are conditionally independent given C, then $\mathbf{P}(A|B, C) = \mathbf{P}(A|C)$
[Solution: true, by definition of conditional independence]

(*b*) If A and B are conditionally independent given C, then $\mathbf{P}(A, B|C) = \mathbf{P}(A|C)\mathbf{P}(B|C)$
[Solution: true, by definition of conditional independence]

(*c*) If A and B are conditionally independent given C, then $\mathbf{P}(A|B, C) = \mathbf{P}(A|B)$
[Solution: False, $\mathbf{P}(A|B, C) = \mathbf{P}(A|C)$ by definition of conditional independence]

(*d*) If A and B are conditionally independent given C, then $\mathbf{P}(A, B, C) = \mathbf{P}(A)\mathbf{P}(B)\mathbf{P}(C)$
[Solution: false: $\mathbf{P}(A, B, C) = \mathbf{P}(A, B|C)\mathbf{P}(C) = \mathbf{P}(A|C)\mathbf{P}(B|C)\mathbf{P}(C) \neq \mathbf{P}(A)\mathbf{P}(B)\mathbf{P}(C)$]

6

(*a*) Describe as Pseudo-Code the WalkSAT local-search SAT procedure.

[Solution:

function WALKSAT(*clauses*, *p*, *max_flips*) **returns** a satisfying model or *failure*
  inputs: *clauses*, a set of clauses in propositional logic
    p, the probability of choosing to do a "random walk" move, typically around 0.5
    max_flips, number of flips allowed before giving up

  model ← a random assignment of *true*/*false* to the symbols in *clauses*
  for $i = 1$ **to** *max_flips* **do**
    if *model* satisfies *clauses* **then return** *model*
    clause ← a randomly selected clause from *clauses* that is false in *model*
    with probability *p* flip the value in *model* of a randomly selected symbol from *clause*
    else flip whichever symbol in *clause* maximizes the number of satisfied clauses
  return *failure*

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

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

1 the input formula is unsatisfiable

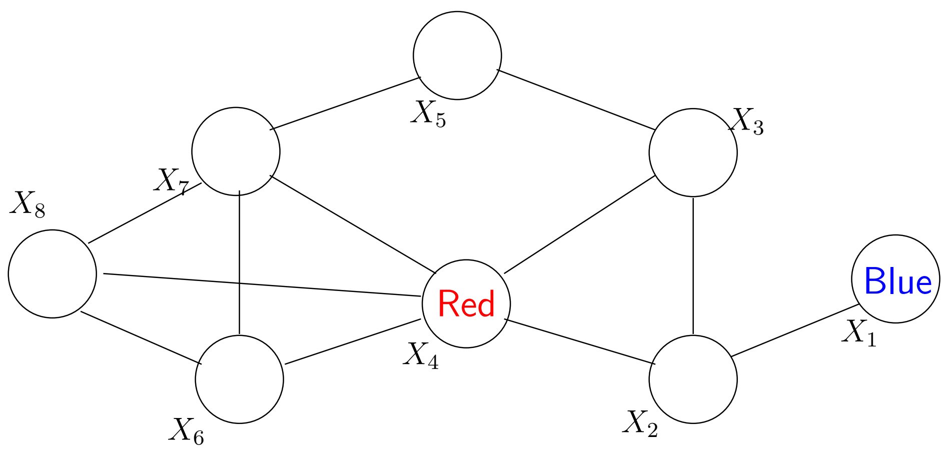
2 we can conclude nothing about the satisfiablity of the input formula

(Say if 1. or 2.)

[Solution: 2.]

7

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



(i) Describe the domain of the unassigned nodes after applying the forward-checking algorithm to the current status of the graph.

[Solution:

$X_2 = \{$, Green , $\}$
$X_3 = \{$, Green , Blue $\}$
$X_5 = \{$ Red , Green , Blue $\}$
$X_6 = \{$, Green , Blue $\}$
$X_7 = \{$, Green , Blue $\}$
$X_8 = \{$, Green , Blue $\}$]

(ii) After running forward-checking, is the resulting status of the problem ARC-consistent?

[Solution: No. E.g., the Green value in X_3 violates the arc-consistency wrt. X_2.]

(iii) Can forward-checking alone detect any inconsistency in this graph? In either case explain why.

[Solution: NO. Although the current status of X_6, X_7, X_8 is unsatisfiable, forward-checking cannot detect it because it does not compare unassigned nodes.]

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: $X_1 : \{ \ , \ , \text{Blue}\}$)

8

Consider the following propositional formula φ:

$$(A_5 \leftrightarrow A_6) \wedge \neg((A_3 \vee \neg A_2) \wedge (\neg A_1 \vee A_4))$$

1. Using the "classic" CNF conversion, produce the CNF formula $CNF(\varphi)$.

   [Solution: The first conjuct can be split into two implications and then easily converted. For the second conjunct, we first transform it into NNF by pushing the negation down to literal level, and then we apply iteratively the De Morgan rule to it.

$$(A_5 \rightarrow A_6) \wedge (A_5 \leftarrow A_6) \wedge ((\neg A_3 \wedge A_2) \vee (A_1 \wedge \neg A_4))$$

$$\begin{array}{l} (\neg A_5 \vee A_6) \wedge \\ (A_5 \vee \neg A_6) \wedge \\ (\neg A_3 \vee A_1) \wedge \\ (\neg A_3 \vee \neg A_4) \wedge \\ (A_2 \vee A_1) \wedge \\ (A_2 \vee \neg A_4) \end{array} .$$

   ]

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

   (i) φ and $CNF(\varphi)$ are equivalent. [Solution: True]

   (ii) φ and $CNF(\varphi)$ are equi-satisfiable, but not necessarily equivalent. [Solution: false]

   (iii) There is no relation between the satisfiablity of φ and that of $CNF(\varphi)$. [Solution: False]

9

Consider the following Horn formula in PL:

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

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

$$\begin{array}{llll}(\ I & \vee\neg L & \vee\neg M) & \wedge\\ (\neg L & \vee\ C & \vee\neg M) & \wedge\\ \\ \\ \\ (\neg L & \vee\ N & \vee\neg H) & \wedge\\ (\neg I & \vee\ L & \vee\neg N) & \wedge\\ \\ \\ \\ (\neg A & \vee\ H & \vee\neg C) & \wedge\\ \\ (\neg A & \vee\neg L & \vee\ M) & \end{array}$$

(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 $\bot$ since the formuila is Horn.

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

]

10

Let $P()$, $Q()$, $R()$, $S()$, $T()$, $U()$ denote predicates, a, b, c, d, e denote constants.

Consider the following set of clauses:

$$
\begin{array}{l}
P(a) \lor Q(c) \lor R(b) \\
\neg U(a) \lor R(b) \\
\neg P(a) \lor R(b) \lor T(e) \\
\neg Q(c) \lor T(e) \lor \neg S(d) \\
\neg R(b) \lor U(a) \\
S(d) \lor T(e) \\
\neg T(e) \lor R(b) \\
\neg R(b)
\end{array}
$$

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:

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

This can be merged into one single step:

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

]

11

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

[Solution: (2 possible versions, AIMA3 version:

```
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 AIMA4 version:

```
function UNIFORM-COST-SEARCH(problem) returns a solution node, or failure
  return BEST-FIRST-SEARCH(problem, PATH-COST)
```

```
function BEST-FIRST-SEARCH(problem, f) returns a solution node or failure
  node ← NODE(STATE=problem.INITIAL)
  frontier ← a priority queue ordered by f, with node as an element
  reached ← a lookup table, with one entry with key problem.INITIAL and value node
  while not IS-EMPTY(frontier) do
    node ← POP(frontier)
    if problem.IS-GOAL(node.STATE) then return node     // late test
    for each child in EXPAND(problem, node) do
      s ← child.STATE          // re-added if now reached by another path with lower cost
      if s is not in reached or child.PATH-COST < reached[s].PATH-COST then
        reached[s] ← child
        add child to frontier
  return failure
```

```
function EXPAND(problem, node) yields nodes     // generates a sequence of nodes
  s ← node.STATE
  for each action in problem.ACTIONS(s) do
    s′ ← problem.RESULT(s, action)
    cost ← node.PATH-COST + problem.ACTION-COST(s, action, s′)
    yield NODE(STATE=s′, PARENT=node, ACTION=action, PATH-COST=cost)
      // add a node to the sequence
```

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

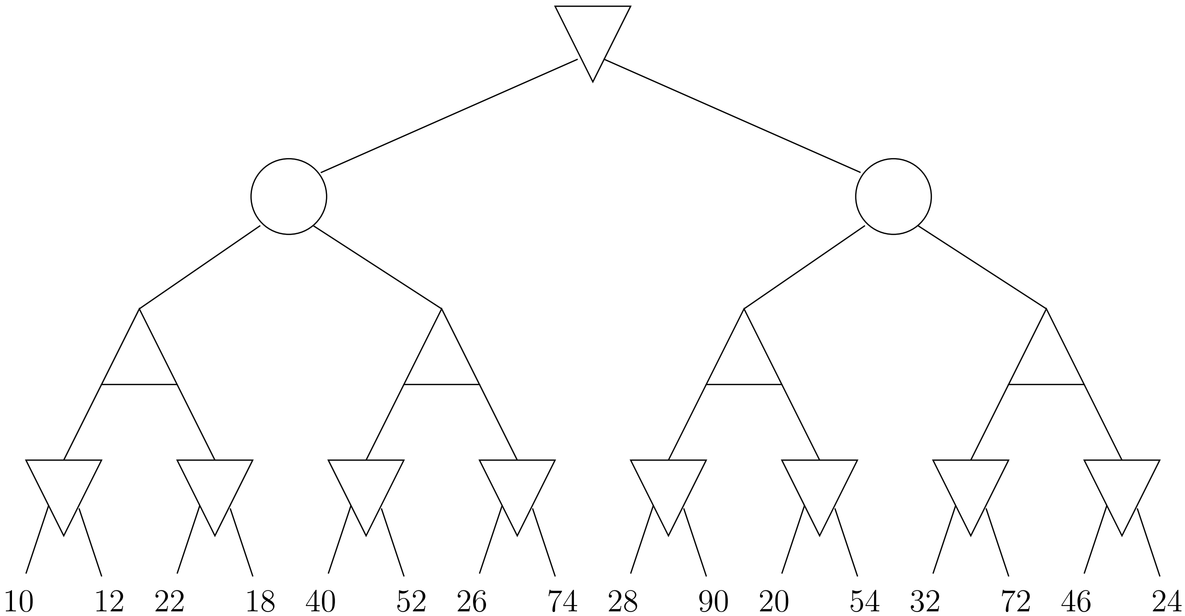
(*b*) When is the goal test applied to a node? (Say if 1. or 2.)

1 when the node is selected for expansion

2 when the node is first generated

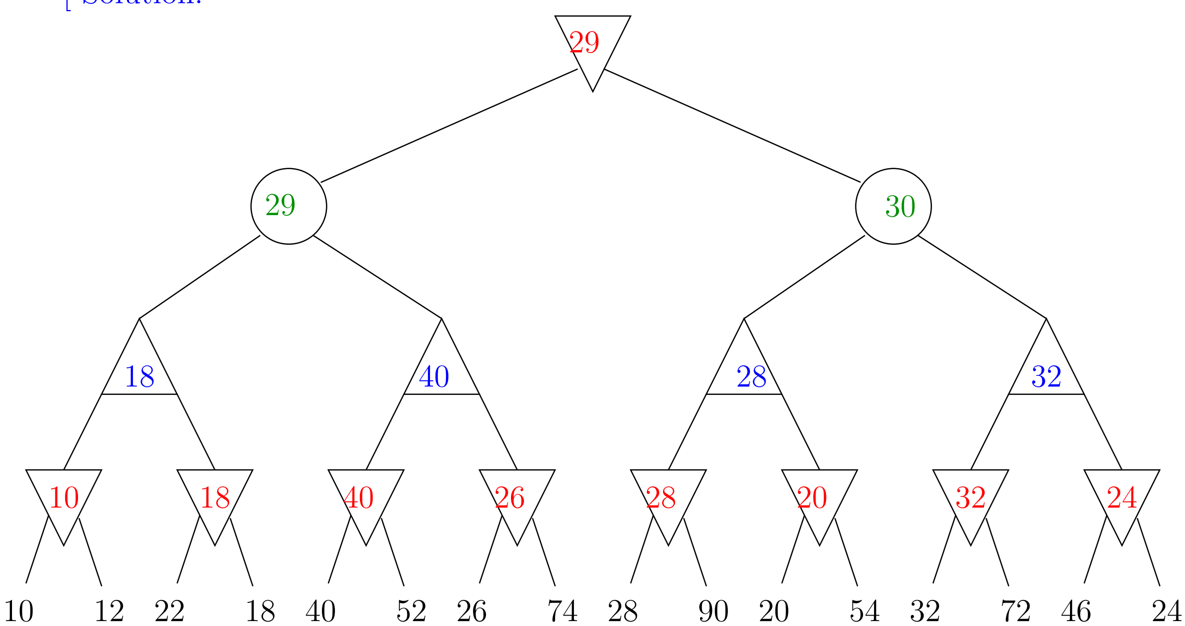
[Solution: 1.]

12



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:



]

13

Consider the following actions with durations / dependencies:

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

$A \prec E \quad A \prec F$
$B \prec E \quad B \prec F \quad B \prec H$
$C \prec E \quad C \prec G$
$D \prec F \quad D \prec G \quad D \prec H$
$E \prec I \quad E \prec J \quad E \prec K$
$F \prec I \quad F \prec J$
$G \prec I \quad G \prec K \quad G \prec L$
$H \prec J \quad H \prec K$

Using the Critical Path method:

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

[Solution:

a	ES	LS
A	0	3
B	0	2
C	0	0
D	0	1
E	3	4
F	2	5
G	3	3
H	2	4
I	6	6
J	5	7
K	6	6
L	6	6

Actions in CP: C, G, I, K, L
Makespan: 9
]

14

Assume the following facts are known from medicine literature: [1]

- 4 persons over 1000 suffer of flu
- 5 % of persons have high_temperature
- one person with flu has high_temperature with probability 0.8

Given that a person has high_temperature, compute the probability of having flu.

[Solution:
From the data, we have that:

$$\begin{aligned} P(\mathsf{flu}) &= \tfrac{4}{1000} &= 0.004 \\ P(\mathsf{high\_temperature}) &= \tfrac{5}{100} &= 0.05 \\ P(\mathsf{high\_temperature}|\mathsf{flu}) & &= 0.8 \end{aligned}$$

Using Bayes' rule:

$$P(\mathsf{flu}|\mathsf{high\_temperature}) = \frac{P(\mathsf{high\_temperature}|\mathsf{flu}) \cdot P(\mathsf{flu})}{P(\mathsf{high\_temperature})} = \frac{0.8 \cdot 0.004}{0.05} = 0.064$$

]

[1] These data are pure fantasy and have no correspondence with real-world medicine.

15

An experienced doctor has to cope with an epidemic of dengue, where 20% of people of the area have been infected. She considers the following possible symptoms:

Symptom #1: high temperature;
Symptom #2: headache;
Symptom #3: vomit.

She models the cause-effect relation as a Naive Bayes Model scenario, s.t the effects are considered conditionally independent given the cause, and she knows from statistics the following data: [2]

$P(high\ temperature \mid\ dengue)$	$= 0.7$
$P(high\ temperature \mid \neg dengue)$	$= 0.1$
$P(headache \mid\ dengue)$	$= 0.3$
$P(headache \mid \neg dengue)$	$= 0.2$
$P(vomit \mid\ dengue)$	$= 0.6$
$P(vomit \mid \neg dengue)$	$= 0.2$

She is informed that one patient has headache and vomit but not high temperature. Compute the probability that such patient has contracted dengue.

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

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

for some normalization constant α. Thus, we have:

$$\begin{array}{lll} P(\ dengue|\neg high\ temperature \wedge headache \wedge vomit) & = \alpha * 0.2 * (1-0.7) * 0.3 * 0.6 & = \alpha * 0.0108 \\ P(\neg dengue|\neg high\ temperature \wedge headache \wedge vomit) & = \alpha * (1-0.2) * (1-0.1) * 0.2 * 0.2 & = \alpha * 0.0288 \end{array}$$

Thus, after normalization:

$$\begin{array}{ll} P(\ dengue|\neg high\ temperature \wedge headache \wedge vomit) = & 0.2727272727272727 \\ P(\neg dengue|\neg high\ temperature \wedge headache \wedge vomit) = & 0.7272727272727273 \end{array}$$

]

[2] The data here are pure fantasy and are not supposed to correspond to actual medical data.