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

Prof. Roberto Sebastiani
DISI, Università di Trento, Italy

2026.02.19

832917

[COPY WITH SOLUTIONS]

1

In the following FOL formulas, let Q, R, P, and $>$, $\leq$, $<$, $\geq$ denote predicates,
g, h, f, 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 $(2 > 4)$ is unsatisfiable.
[Solution: false (in FOL “>”, “2”, “4” have no fixed interpretation)]

(b) The FOL formula $\forall x.((x > 2) \rightarrow (x > 1))$ is valid.
[Solution: false (in FOL “>”, “2”, “1” have no fixed interpretation)]

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

(d) The FOL formula $(\forall x_2 \exists x_1.Q(x_1, x_2)) \rightarrow (\exists x_1 \forall x_2.Q(x_1, x_2))$ is valid.
[Solution: false]

2

Given the following symbols, representing concept, relation and individual names in the alien language of the remote planet Sgotz:

- a set of primitive $\mathcal{ALCQ}$ concept names: {Perun, Malle, Feale, Dotot, Egier}
- a set of $\mathcal{ALCQ}$ relation names: {haCidod}
- a set of $\mathcal{ALCQ}$ individual names: {Lok, Jud, Etu, Dib, Ate, Bek, Mop, Gop}

and the following $\mathcal{ALCQ}$ $\mathcal{T}$-box $\mathcal{T}$ and $\mathcal{A}$-box $\mathcal{A}$:

$\mathcal{T}$		
Perun		⟨*primitive concept*⟩
Feale		⟨*primitive concept*⟩
Malle		⟨*primitive concept*⟩
Dotot		⟨*primitive concept*⟩
Egier		⟨*primitive concept*⟩
Woana	$\equiv$	Perun $\sqcap$ Feale
Mnana	$\equiv$	Perun $\sqcap$ Malle
Moeth	$\equiv$	Woana $\sqcap$ $\exists$haCidod.Perun
Faeth	$\equiv$	Mnana $\sqcap$ $\exists$haCidod.Perun
Panet	$\equiv$	Perun $\sqcap$ haCidod.Perun
Gamae	$\equiv$	Moeth $\sqcap$ haCidod.Panet
Gafae	$\equiv$	Faeth $\sqcap$ haCidod.Panet
Gapan	$\equiv$	Panet $\sqcap$ haCidod.Panet

$\mathcal{A}$

Etu : Woana; Mop : Woana; Gop : Woana; Lok : Woana;

Jud : Mnana; Ate : Mnana; Bek : Mnana; Dib : Mnana;

Ate : Dotot; Etu : Dotot

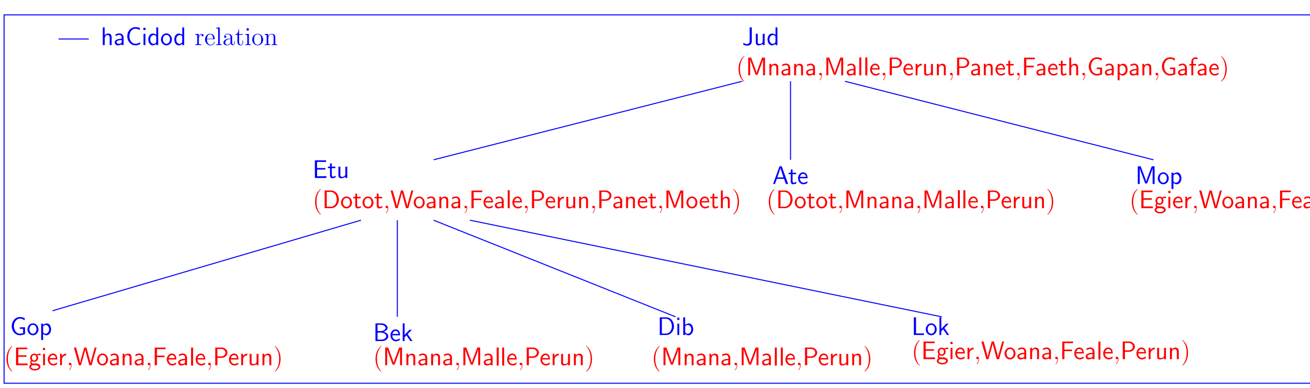
Mop : Egier; Gop : Egier; Lok : Egier

⟨Jud, Etu⟩ : haCidod; ⟨Jud, Ate⟩ : haCidod;
⟨Jud, Mop⟩ : haCidod;

⟨Etu, Gop⟩ : haCidod; ⟨Etu, Bek⟩ : haCidod;
⟨Etu, Dib⟩ : haCidod; ⟨Etu, Lok⟩ : haCidod;

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

(a) Jud : Gapan $\sqcap$ $\forall$haCidod.Moeth [Solution: false]
(b) Jud : Faeth $\sqcap$ $\exists$haCidod.Panet [Solution: true]
(c) Jud : Faeth $\sqcap$ (≥ 2)haCidod.Dotot [Solution: true]
(d) Etu : Moeth $\sqcap$ (≥ 3)haCidod.Egier [Solution: false]

[Solution: 

]

3

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

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

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

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

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

4

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)

s : depth of the shallowest solution

Assume also that all steps cost are 1.

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

(a) Depth-First Search with loop-prevention requires $O(b^s)$ memory to find a solution.
[Solution: false, requires $O(bm)$ memory]

(b) Depth-First Search with loop-prevention requires $O(b^s)$ steps to find a solution.
[Solution: false, requires $O(b^m)$ steps]

(c) Breadth-First Search is optimal
[Solution: true]

(d) Breadth-First Search requires $O(b^m)$ memory to find a solution.
[Solution: false, requires $O(b^s)$ memory]

5

Consider propositional logic (PL); let $G,\ F,\ E,\ D,\ C,\ B,\ A$ be atomic propositions
We adopt the set notation for resolution rules, s.t. Γ denotes a set of clauses.

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

(*a*) The following is a correct application of the PL general resolution rule:

$$\frac{\Gamma,\quad (\ B \vee \neg E \vee\ \ G),\quad (\neg B \vee \neg G \vee\ \ D)}{\Gamma,\quad (\neg E \vee\ \ D)}$$

[Solution: False]

(*b*) The following is a correct application of the PL general resolution rule:

$$\frac{\Gamma,\quad (\ E \vee \neg D \vee \neg B),\quad (\neg G \vee \neg E \vee \neg D)}{\Gamma,\quad (\neg D \vee \neg G \vee \neg B)}$$

[Solution: true]

(*c*) The following is a correct application of the PL clause-subsumption rule:

$$\frac{\Gamma,\quad (\ B \vee\ \ G),\quad (\ B \vee \neg E \vee\ \ G)}{\Gamma,\quad (\ B \vee \neg E \vee\ \ G)}$$

[Solution: false]

(*d*) The following is a correct application of the PL unit-resolution rule:

$$\frac{\Gamma,\quad (\ E),\quad (\ B \vee \neg E \vee\ \ G)}{\Gamma,\quad (\ E),\quad (\ B \vee\ \ G)}$$

[Solution: true]

6

Given the following Sudoku scenario:

1	2	3	4	5	6	7	8	9	
1		8							A
2				5					B
9						4	6		C
8								6	D
7								4	E
5								3	F
6	8	1							G
	7	2		3					H
		9							I

(a) Apply the AC-3 algorithm. Describe in the right sequence the domains of unassigned nodes whose domains become unary after one run of AC-3. (E.g.:
$D_{A1} := \{3\}$,
$D_{B1} := \{7\}$,
...)

(b) Can AC-3 reduce to unary the domains of nodes $B3$ and $C3$?

(c) After one run of AC-3, is the resulting graph arc-consistent?

[Solution:

(a) We have, in sequence
$D_{H1} := \{4\}$,
$D_{I1} := \{3\}$,
$D_{I2} := \{5\}$,
$D_{C2} := \{3\}$

(b) No. Decent Sudoku players can infer also $A2 = B2 = \{6, 4\}$, so that, by path-consistency (not arc-consistency!), we have $B3 := \{7\}$, $C3 := \{5\}$, but this cannot be performed by AC-3.

(c) Yes. AC-3 always makes a graph arc-consistent.

]

7

Consider the following CNF formula in PL:

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

Consider the WalkSAT algorithm, with probability parameter $p = 0.2$. Suppose at a given step the current assignment is

$$\{ \ A, \ B, \ C, \ D, \neg E, \neg F, \ G, \neg H, \ I, \neg L, \neg M, \neg N\}.$$

Assuming the most-likely event happens, describe what the assignment is after the next step.

[Solution:

The current assignments makes only the 1^{st} clause unsatisfied. Since $p = 0.2$, the most likely event is that the algorithm flips the symbol in the first clause which maximizes the number of satisfied clause at next step.

We notice that flipping B would cause no other clause to become unsatisfied, whereas flipping either of $\neg E, \neg N$ would cause other clauses to become unsatisfied. Thus WalkSAT flips B, obtaining the assignment

$$\{ \ A, \neg B, \ C, \ D, \neg E, \neg F, \ G, \neg H, \ I, \neg L, \neg M, \neg N\}.$$

which satisfies all clauses.]

8

An experienced doctor has to cope with an epidemic of covid19, where 70% of people of the area have been infected. She considers the following possible symptoms:

Symptom #1: nausea;
Symptom #2: headache;
Symptom #3: fever.

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: [1]

$P(nausea\vert\ covid19)$	$= 0.7$
$P(nausea\vert\neg covid19)$	$= 0.1$
$P(headache\vert\ covid19)$	$= 0.3$
$P(headache\vert\neg covid19)$	$= 0.2$
$P(fever\vert\ covid19)$	$= 0.6$
$P(fever\vert\neg covid19)$	$= 0.2$

She is informed that one patient has headache and fever but not nausea. Compute the probability that such patient has contracted covid19.

Notice: *the problem must be solved my using the Naive Bayes Model scenario. Any attempt to use any other technique will be considered incorrect.*

[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:

$$\begin{array}{rl} & P(\ covid19|\neg nausea \wedge headache \wedge fever) \\ = & \alpha * P(\ covid19) * P(\neg nausea|\ covid19) * P(\ headache|\ covid19) * P(\ fever|\ covid19) \\ = & \alpha * 0.7 * (1 - 0.7) * 0.3 * 0.6 \\ = & \alpha * 0.0378 \\ & P(\neg covid19|\neg nausea \wedge headache \wedge fever) \\ = & \alpha * P(\neg covid19) * P(\neg nausea|\neg covid19) * P(\ headache|\neg covid19) * P(\ fever|\neg covid19) \\ = & \alpha * (1 - 0.7) * (1 - 0.1) * 0.2 * 0.2 \\ = & \alpha * 0.0108 \end{array}$$

Thus, after normalization:

$$\begin{array}{rl} P(\ covid19|\neg nausea \wedge headache \wedge fever) = & 0.7777777777777778 \\ P(\neg covid19|\neg nausea \wedge headache \wedge fever) = & 0.2222222222222222 \end{array}$$

]

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

9

(a) Describe as Pseudo-Code the Depth-Limited Search and Iterative-Deepening procedures.

[Solution:

```
function DEPTH-LIMITED-SEARCH(problem, limit) returns a solution, or failure/cutoff
  return RECURSIVE-DLS(MAKE-NODE(problem.INITIAL-STATE), problem, limit)

function RECURSIVE-DLS(node, problem, limit) returns a solution, or failure/cutoff
  if problem.GOAL-TEST(node.STATE) then return SOLUTION(node)
  else if limit = 0 then return cutoff
  else
     cutoff_occurred? ← false
     for each action in problem.ACTIONS(node.STATE) do
        child ← CHILD-NODE(problem, node, action)
        result ← RECURSIVE-DLS(child, problem, limit − 1)
        if result = cutoff then cutoff_occurred? ← true
        else if result ≠ failure then return result
     if cutoff_occurred? then return cutoff else return failure

function ITERATIVE-DEEPENING-SEARCH(problem) returns a solution, or failure
   for depth = 0 to ∞ do
     result ← DEPTH-LIMITED-SEARCH(problem, depth)
     if result ≠ cutoff then return result
```

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

(b) calling b the branching factor and s the depth of the shallowest solution,

- what is the time complexity of the Iterative-Deepening procedure? [Solution: $O(b^s)$]
- what is the memory complexity of the Iterative-Deepening procedure? [Solution: $O(bs)$]

10

(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 ← number of variables in X
  assignment ← an empty assignment
  root ← any variable in X
  X ← 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] ← 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.]

11

Use *hill climbing* for solving the maximization problem over (x, y) with the following objective function:

y								
7	0	2	3	4	4	2	1	0
6	1	1	4	5	5	3	2	1
5	2	3	5	8	6	4	2	3
4	3	5	6	9	10	6	5	1
3	3	4	7	10	9	6	4	1
2	2	2	4	6	7	5	4	3
1	1	1	3	5	4	3	1	1
0	0	1	2	3	4	2	2	0
	0	1	2	3	4	5	6	7 x

The next state is selected among the neighbors with strictly higher objective function value with *probability proportional to their objective function value*. Neighbors of (x, y) are sorted as follows:

$$(x-1, y-1), (x, y-1), (x+1, y-1), (x-1, y), (x+1, y), (x-1, y+1), (x, y+1), (x+1, y+1)$$

The choice vector is: $[1/4, 3/4]$.

The *initial state* is: $(7, 0)$.
For each step in the resolution process report (1) the current state; (2) the list of candidate next states.

[Solution:
it: 1, curr: (7, 0), candidates = [(6, 0) : 1/2, (6, 1) : 1/4, (7, 1) : 1/4], choice: 1/4 → (6, 0)
it: 2, curr: (6, 0), candidates = [(5, 1) : 1], choice: 3/4 → (5, 1)
it: 3, curr: (5, 1), candidates = [(4, 0) : 4/24, (4, 1) : 4/24, (4, 2) : 7/24, (5, 2) : 5/24, (6, 2) : 4/24], choice: 1/4 → (4, 1)
it: 4, curr: (4, 1), candidates = [(3, 1) : 5/23, (3, 2) : 6/23, (4, 2) : 7/23, (5, 2) : 5/23], choice: 3/4 → (4, 2)
it: 5, curr: (4, 2), candidates = [(3, 3) : 10/19, (4, 3) : 9/19], choice: 1/4 → (3, 3)
it: 6, curr: (3, 3), candidates = [], STOP

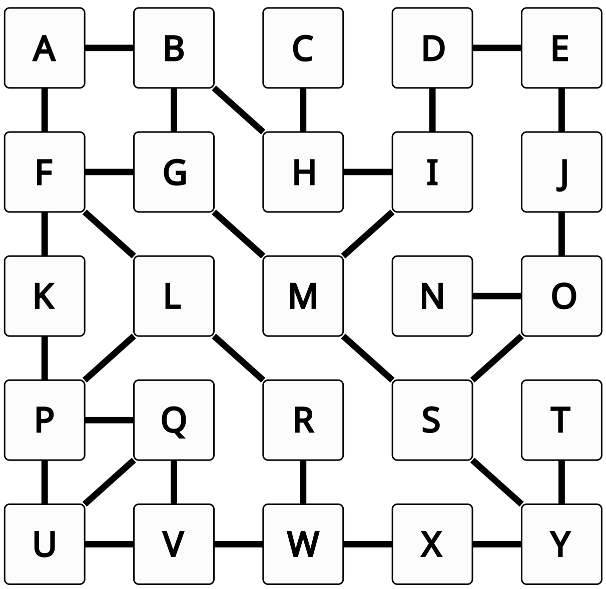
]

12

In the following state graph, apply *breadth-first search* 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.



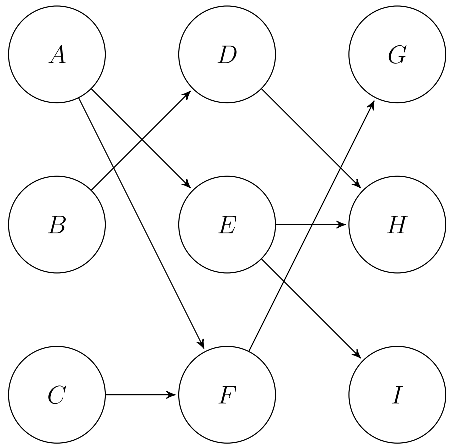
Start: **M** - Goal: **P**

[Solution:
M - [G, I, S]
G - [B, F]
I - [D, H]
S - [O, Y]
B - [A]
F - [K, L]
D - [E]
H - [C]
O - [J, N]
Y - [T, X]
A - []
K - stop
]

13

Consider the following actions with durations / dependencies:

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



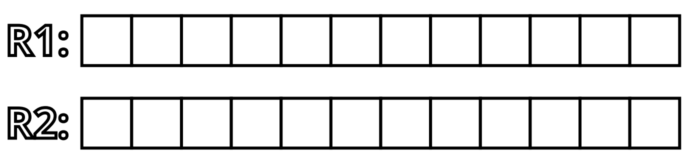
1) Compute the earliest / latest possible start time (ES/LS) for each action using the Critical Path method. [40pts.]

Assume that completing each action requires one unit of a reusable resource:

2) Report an optimal schedule given 2 reusable resources in the grid below. [40pts.]

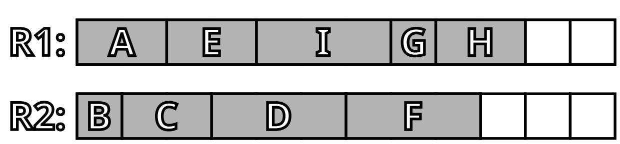
3) What would be the optimal execution time given 1 reusable resource? [10pts.]

4) What would be the optimal execution time given infinite resources? [10pts.]



[Solution:

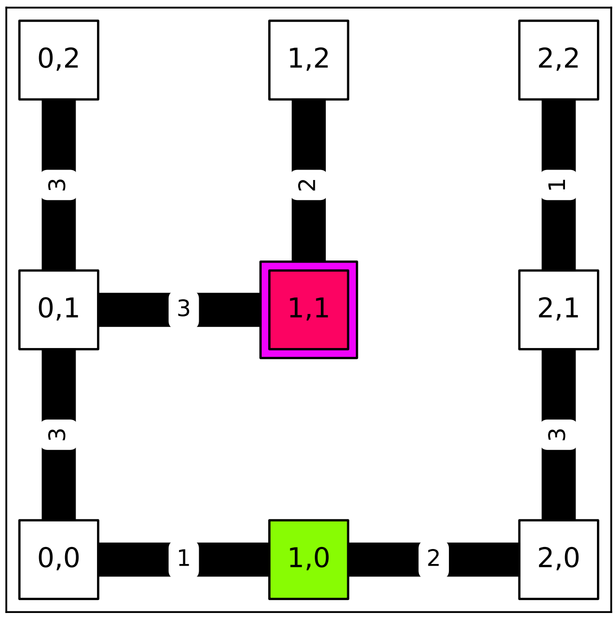
a	ES	LS
A	0	0
B	0	1
C	0	1
D	1	2
E	2	2
F	2	3
G	5	6
H	4	5
I	4	4



3) 19.
4) 7.]

14

In the following state graph, apply *LRTA** 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 $[up, right, down, left]$.



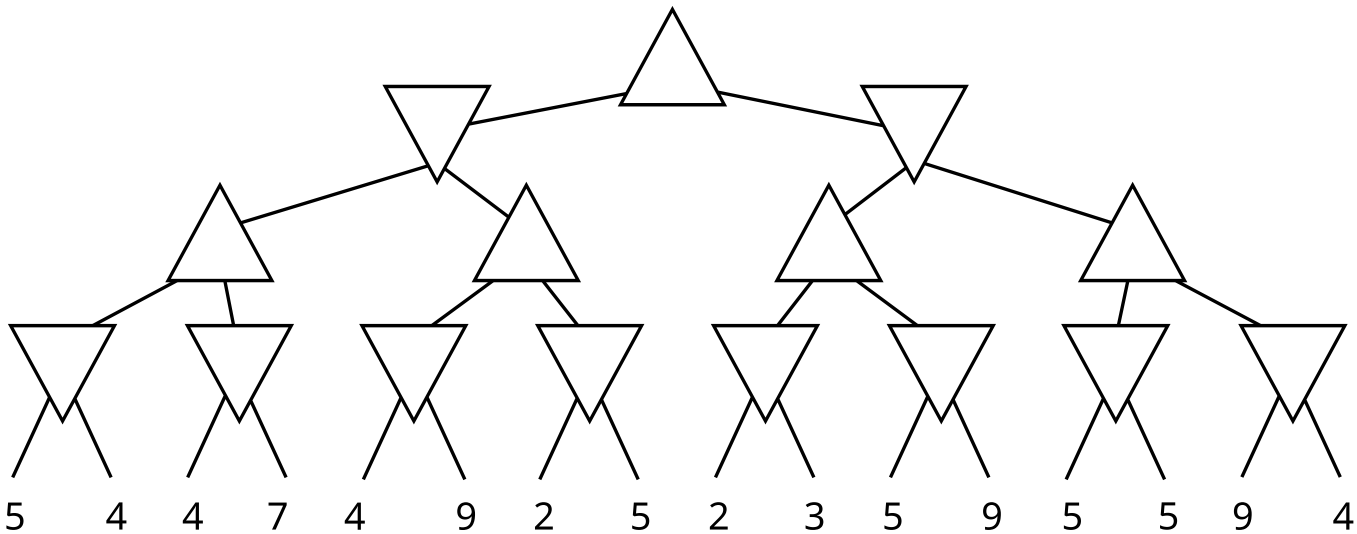
Start: $(1,1)$ - Goal: $(1,0)$

[Solution:

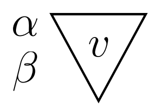
$(1,1)$, $(1,2)$, $(1,1)$, $(0,1)$, $(0,2)$, $(0,1)$, $(1,1)$, $(1,2)$, $(1,1)$, $(0,1)$, $(0,0)$, $(0,1)$, $(0,0)$, $(1,0)$
]

15

Use *Minimax with α, β-pruning* to propagate the utilities from the leaves to the root of the tree.

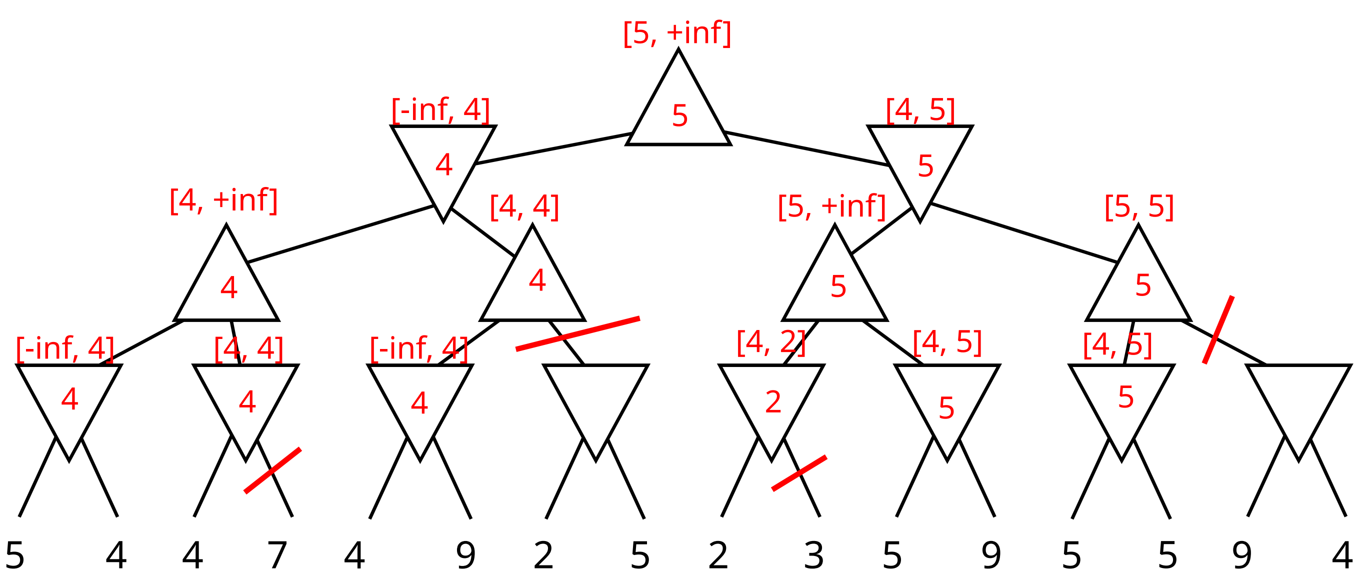


For each node report the values of α, β, as well as its returned value v **when the recursive calls ends**. Use the following format:



Additionally, clearly mark in the tree the pruned branches.

[Solution:



]