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

Prof. Roberto Sebastiani
DISI, Università di Trento, Italy

2026.02.19

812917

[COPY WITH SOLUTIONS]

1

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 $O(b^s)$ steps to find a solution.
[Solution: false, requires $O(b^m)$ steps]

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

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

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

2

Consider propositional logic (PL); let A, B, C, D, E, F, G 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:

$$\frac{\Gamma, \quad (\ C \vee \neg D \vee \neg F), \quad (\neg A \vee \neg C \vee \neg D)}{\Gamma, \quad (\neg D \vee \neg A \vee \neg F)}$$

[Solution: true]

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

$$\frac{\Gamma, \quad (\ F \vee \neg C \vee \ \ A), \quad (\neg F \vee \neg A \vee \ \ D)}{\Gamma, \quad (\neg C \vee \ \ D)}$$

[Solution: False]

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

$$\frac{\Gamma, \quad (\ C), \quad (\ F \vee \neg C \vee \ \ A)}{\Gamma, \quad (\ C), \quad (\ F \vee \ \ A)}$$

[Solution: true]

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

$$\frac{\Gamma, \quad (\ F \vee \ \ A), \quad (\ F \vee \neg C \vee \ \ A)}{\Gamma, \quad (\ F \vee \neg C \vee \ \ A)}$$

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

(b) The FOL formula $(2 > 4)$ is unsatisfiable.
[Solution: false (in FOL ">", "2", "4" have no fixed interpretation)]

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

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

4

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, Mop, Dib, Ate, Bek, Etu, Gop}

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

$\mathcal{T}$			$\mathcal{A}$
Perun		$\langle primitive\ concept \rangle$	Mop : Woana; Etu : Woana; Gop : Woana; Lok : Woana;
Feale		$\langle primitive\ concept \rangle$	
Malle		$\langle primitive\ concept \rangle$	Jud : Mnana; Ate : Mnana; Bek : Mnana; Dib : Mnana;
Dotot		$\langle primitive\ concept \rangle$	
Egier		$\langle primitive\ concept \rangle$	Ate : Dotot; Mop : Dotot
Woana	$\equiv$	Perun $\sqcap$ Feale	
Mnana	$\equiv$	Perun $\sqcap$ Malle	Etu : Egier; Gop : Egier; Lok : Egier
Moeth	$\equiv$	Woana $\sqcap \exists$haCidod.Perun	
Faeth	$\equiv$	Mnana $\sqcap \exists$haCidod.Perun	$\langle$Jud, Mop$\rangle$: haCidod; $\langle$Jud, Ate$\rangle$: haCidod;
Panet	$\equiv$	Perun $\sqcap$ haCidod.Perun	$\langle$Jud, Etu$\rangle$: haCidod;
Gamae	$\equiv$	Moeth $\sqcap$ haCidod.Panet	
Gafae	$\equiv$	Faeth $\sqcap$ haCidod.Panet	$\langle$Mop, Gop$\rangle$: haCidod; $\langle$Mop, Bek$\rangle$: haCidod;
Gapan	$\equiv$	Panet $\sqcap$ haCidod.Panet	$\langle$Mop, Dib$\rangle$: haCidod; $\langle$Mop, Lok$\rangle$: haCidod;

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

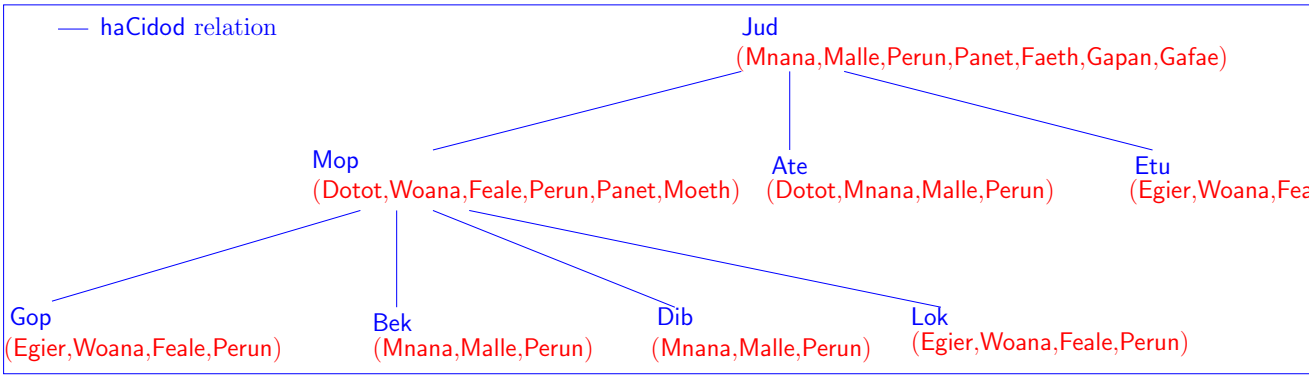
(a) Jud : Faeth $\sqcap \exists$haCidod.Panet [Solution: true]

(b) Jud : Gapan $\sqcap \forall$haCidod.Moeth [Solution: false]

(c) Mop : Moeth $\sqcap (\geq 3)$haCidod.Egier [Solution: false]

(d) Jud : Faeth $\sqcap (\geq 2)$haCidod.Dotot [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 $\mathbf{P}(A, B, C) = \mathbf{P}(A)\mathbf{P}(B)\mathbf{P}(C)$
[Solution: false: $\mathbf{P}(A, B, C) = \mathbf{P}(A, B|C)\mathbf{P}(C) = \mathbf{P}(A|C)\mathbf{P}(B|C)\mathbf{P}(C) \neq \mathbf{P}(A)\mathbf{P}(B)\mathbf{P}(C)$]

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

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

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

6

(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 d the depth of the shallowest solution,

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

7

(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.]

8

Given the following Sudoku scenario:

1	2	3	4	5	6	7	8	9	
1		4							A
2				6					B
3						8	7		C
4								7	D
5								8	E
6								9	F
7	4	1							G
	5	2		9					H
		3							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
$D_{H1} := \{8\}$,
$D_{I1} := \{9\}$,
$D_{I2} := \{6\}$,
$D_{C2} := \{9\}$

(b) No. Decent Sudoku players can infer also $A2 = B2 = \{7, 8\}$, so that, by path-consistency (not arc-consistency!), we have $B3 := \{5\}$, $C3 := \{6\}$, but this cannot be performed by AC-3.

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

]

9

Consider the following CNF formula in PL:

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

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

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

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 $E, \neg N$ would cause other clauses to become unsatisfied. Thus WalkSAT flips B, obtaining the assignment

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

which satisfies all clauses.]

10

An experienced doctor has to cope with an epidemic of covid19, where 30% of people of the area have been infected. She considers the following possible symptoms:

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

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]

$P(fever\|\ \ covid19)$	$= 0.8$
$P(fever\|\neg covid19)$	$= 0.1$
$P(nausea\|\ \ covid19)$	$= 0.4$
$P(nausea\|\neg covid19)$	$= 0.2$
$P(headache\|\ \ covid19)$	$= 0.6$
$P(headache\|\neg covid19)$	$= 0.3$

She is informed that one patient has nausea and headache but not fever. 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:

$$\begin{aligned}
& P(\ \ covid19|\neg fever \wedge nausea \wedge headache) \\
= \ & \alpha * P(\ \ covid19) * P(\neg fever|\ \ covid19) * P(\ \ nausea|\ \ covid19) * P(\ \ headache|\ \ covid19) \\
= \ & \alpha * 0.3 * (1 - 0.8) * 0.4 * 0.6 \\
= \ & \alpha * 0.0144 \\
& P(\neg covid19|\neg fever \wedge nausea \wedge headache) \\
= \ & \alpha * P(\neg covid19) * P(\neg fever|\neg covid19) * P(\ \ nausea|\neg covid19) * P(\ \ headache|\neg covid19) \\
= \ & \alpha * (1 - 0.3) * (1 - 0.1) * 0.2 * 0.3 \\
= \ & \alpha * 0.0378
\end{aligned}$$

Thus, after normalization:

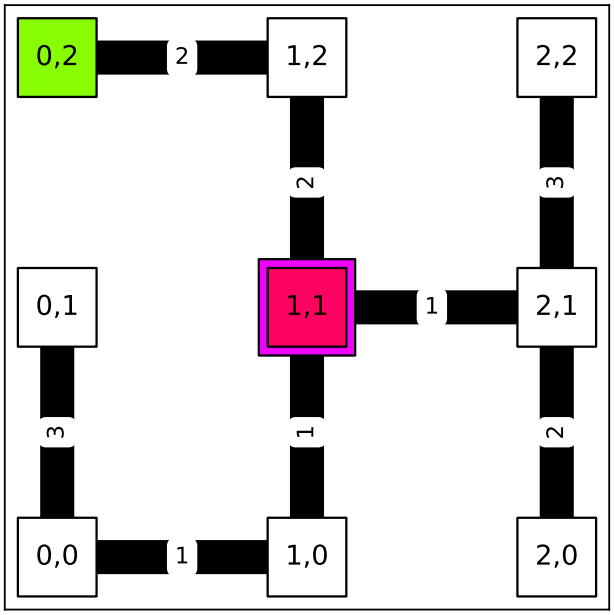
$$\begin{aligned}
P(\ \ covid19|\neg fever \wedge nausea \wedge headache) = \ & 0.2758620689655172 \\
P(\neg covid19|\neg fever \wedge nausea \wedge headache) = \ & 0.7241379310344828
\end{aligned}$$

]

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

11

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: $(0,2)$

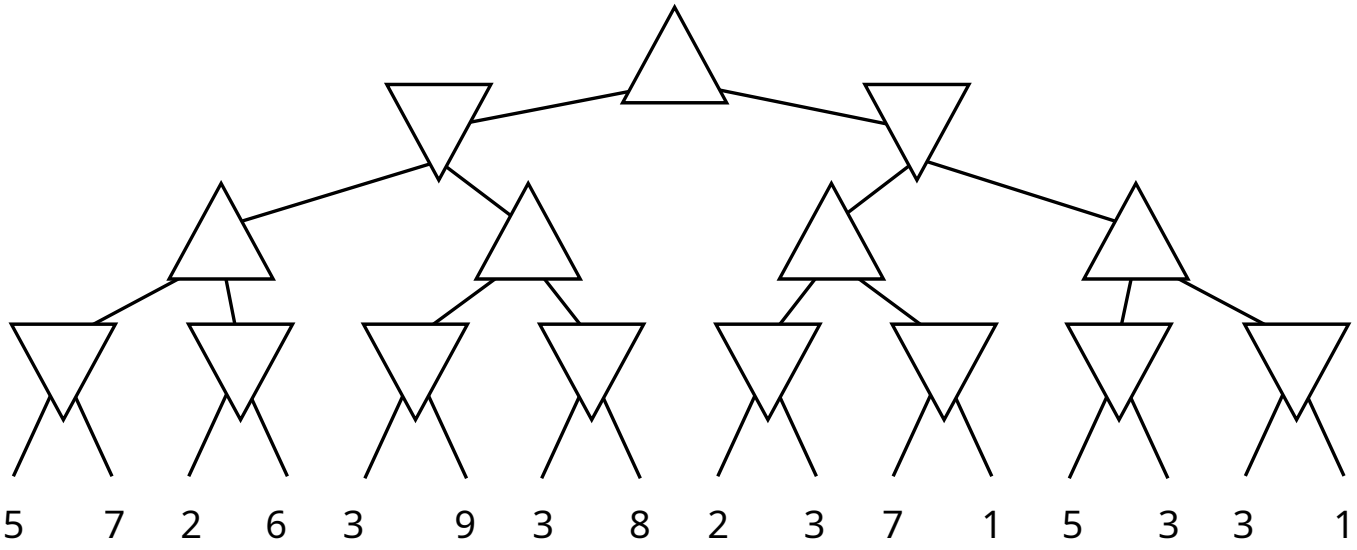
[Solution:

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

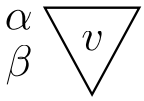
]

12

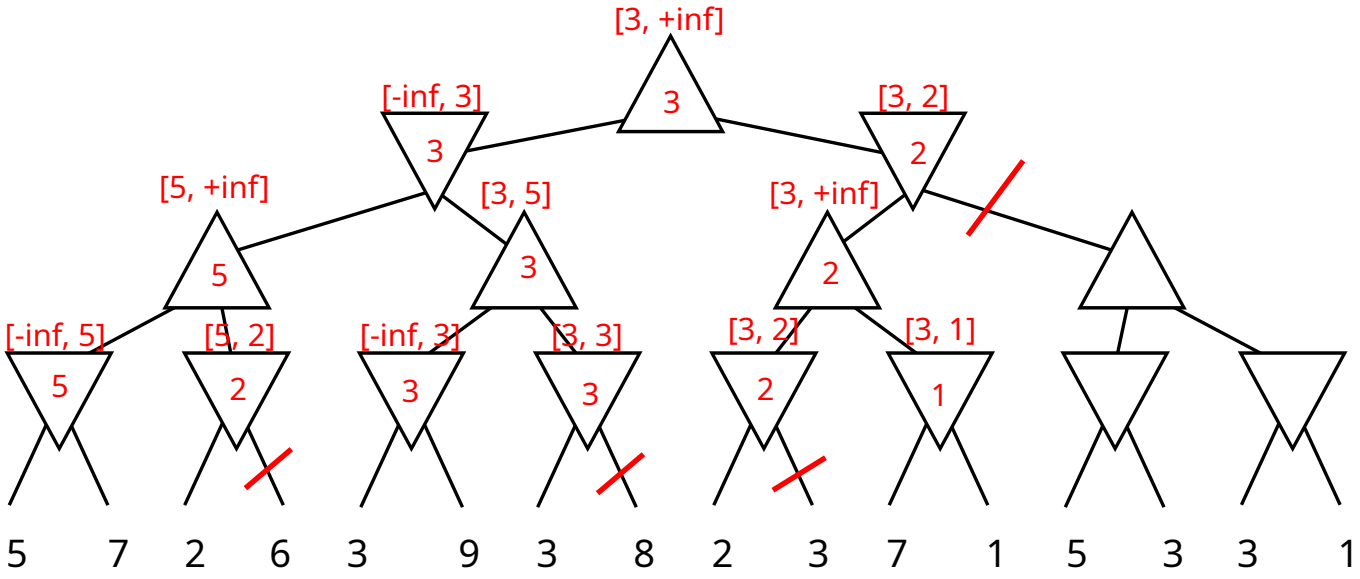
Use *Minimax with α, β-pruning* to propagate the utilities from the leaves to the root of the tree.



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:



]

13

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

[Solution:
it: 1, curr: (0, 0), candidates = [(1, 0) : 1/3, (0, 1) : 1/3, (1, 1) : 1/3], choice: 1/4 → (1, 0)
it: 2, curr: (1, 0), candidates = [(2, 0) : 2/5, (2, 1) : 3/5], choice: 3/4 → (2, 1)
it: 3, curr: (2, 1), candidates = [(3, 1) : 5/15, (2, 2) : 4/15, (3, 2) : 6/15], choice: 1/4 → (3, 1)
it: 4, curr: (3, 1), candidates = [(3, 2) : 6/13, (4, 2) : 7/13], 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

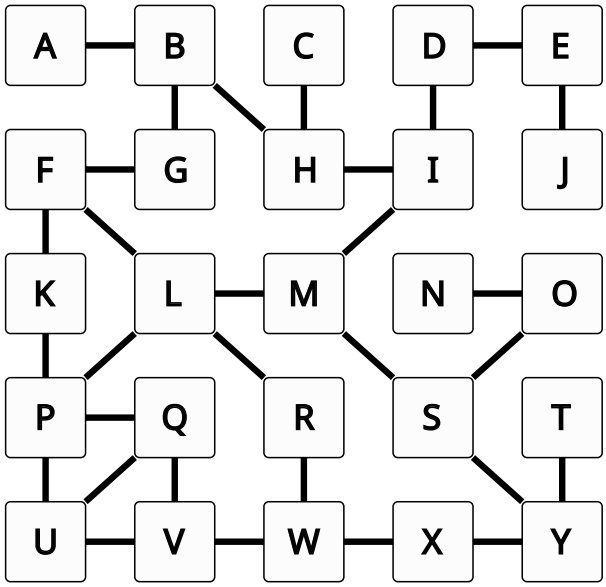
]

14

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.



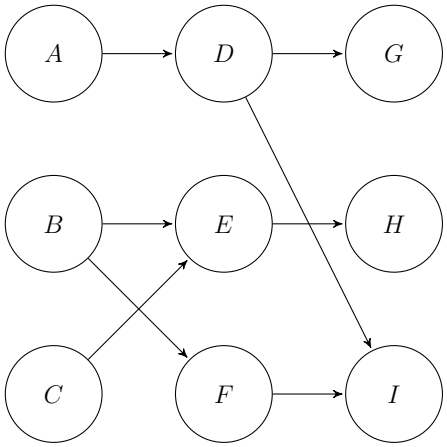
Start: **M** - Goal: **J**

[Solution:
M - [I, L, S]
I - [D, H]
L - [F, P, R]
S - [O, Y]
D - [E]
H - [B, C]
F - [G, K]
P - [Q, U]
R - [W]
O - [N]
Y - [T, X]
E - stop
]

15

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:

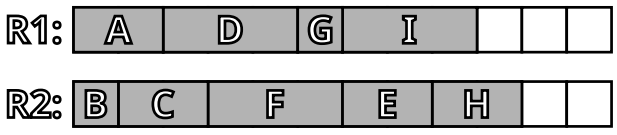
2) Report an optimal schedule given 2 reusable resources in the grid below. [40pts.]

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	2
D	2	2
E	2	4
F	1	2
G	5	7
H	4	6
I	5	5



3) 19.
4) 8.]