INSTITUTO SUPERIOR TÉCNICO

Search and Planning

2022/2023 Academic Year — 1^{st} Period

1^{st} Exam — November 14, 2022 — Duration: 2h

- This is a closed book exam.
- Ensure that your name and number are written on all pages.

EXAM SOLUTION

I. Modeling as a CSP (2 + 2 = 4/20)

1) An antisymmetric matrix is a square matrix that satisfies the identity $A^T = -A$. For example, matrix A is antisymmetric:

$$A = \begin{pmatrix} 0 & 2 & 3 \\ -2 & 0 & -1 \\ -3 & 1 & 0 \end{pmatrix}$$

Propose a formulation of the problem of deciding whether a matrix 3×3 is antisymmetric as a constraint network. Consider that the values in the matrix range from -9 to 9. Identify variables, domains, and constraints. Ensure that all variables are node consistency.

Solution:

Variables: $x_{ij},\ 1 \leq i, j \leq 3$, corresponding to the contents of the matrix.

$$A = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix}$$

Domains:

$$D(x_{ij}) = \{-9, ..., 9\}, \quad \forall_{1 \leq i,j \leq 3}$$

Constraints:

$$x_{ji} = -x_{ij}, \quad \forall_{1 \leq i \leq j \leq 3}$$

Variables x_{11}, x_{22}, x_{33} are used in unary constraints:

$$x_{11} = -x_{11}$$

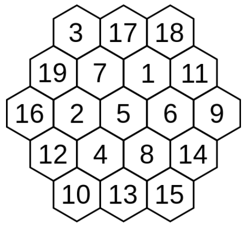
$$x_{22} = -x_{22}$$

$$x_{33} = -x_{33}$$

To ensure node consistency, the domains should be updated to:

$$D(x_{11}) = D(x_{22}) = [D(x_{33}) = \{0\}$$

2) A *magic hexagon* of order n is an arrangement of numbers in a centered hexagonal pattern with n cells on each edge, in such a way that the numbers in each row, in all three directions, sum to the same magic constant M. A *normal* magic hexagon contains the consecutive integers from 1 to $3n^2 - 3n + 1$. It turns out that normal magic hexagons exist only for $n = 1$ (which is trivial, as it is composed of only 1 cell) and $n = 3$. Moreover, the solution of order 3 is essentially unique and is shown in the figure.



Propose a formulation of the problem of deciding whether a magic hexagon with $n = 3$ is a normal magic hexagon. Identify variables, domains, and constraints.

Solution:
Note that the numbers range from 1 to 19 and $M = 38$.
19 Variables:

```
A,B,C
D,E,F,G
H,I,J,K,L
M,N,O,P
Q,R,S
```

Domains: the same for all variables [1..19]
Constraints: AllDifferent(A,B,C,...,S)

```
A+B+C = D+E+F+G = ... = Q+R+S = 38,
A+D+H = B+E+I+M = ... = L+P+S = 38,
C+G+L = B+F+K+P = ... = H+M+Q = 38.
```

II. Inference in CSP (1 + 1.5 + 1.5 = 4/20)

1) Consider a four-variable X_0, X_1, X_2, X_3 constraint network with the respective domains $D_0 = \{1,2,3,4,7\}, D_1 = \{2,4,6,8\}, D_2 = \{3,4,5,7,8,9\}, D_3 = \{6,7,8\}$. The constraints are: $X_0 = X_2$, $X_3 > X_0$ and $X_0 > X_1$.

Apply AC-3 to the network according to the algorithm that is given below.

```
AC-3(ℛ)
Input: A network of constraints ℛ = (X, D, C).
Output: ℛ', which is the largest arc-consistent network equivalent to ℛ.
for every pair {x_i, x_j} that participates in a constraint R_ij ∈ ℛ
    queue ← queue ∪ {(x_i, x_j), (x_j, x_i)}
endfor
while queue ≠ {}
    select and delete (x_i, x_j) from queue
    REVISE((x_i), x_j)
    if REVISE((x_i), x_j) causes a change in D_i
        then queue ← queue ∪ {(x_k, x_i), k ≠ i, k ≠ j}
    endif
endwhile
```

Solution:

Edge	New Domain	Edges to Reconsider
('X0', 'X1')	X0=[3, 4, 7]	–
('X1', 'X0')	X1=[2, 4, 6]	–
('X0', 'X2')	X0=[3, 4, 7]	–
('X2', 'X0')	X2=[3, 4, 7]	–
('X0', 'X3')	X0=[3, 4, 7]	–
('X3', 'X0')	X3=[6, 7, 8]	–

The algorithm terminates with the domains $D_0 = D_2 = \{3,4,7\}, D_1 = \{2,4,6\}$ and $D_3 = \{6,7,8\}$.

2) Consider a four-variable X_1, X_2, X_3, X_4 constraint network with the respective domains $D_1 = \{1,2,3,4,7\}, D_2 = \{2,4,6,8\}, D_3 = \{3,4,5,7,8,9\}, D_4 = \{6,7,8\}$. The constraints are: $X_1 = X_3$ and $X_4 > X_1 + X_2$. Apply the PC-2 algorithm.

```
PC-2(ℛ)
Input: A network ℛ = (X, D, C).
Output: ℛ' a path-consistent network equivalent to ℛ.
Q ← {(i, k, j) | 1 ≤ i < j ≤ n, 1 ≤ k ≤ n, k ≠ i, k ≠ j}
while Q is not empty
      select and delete a 3-tuple (i, k, j) from Q
      R_ij ← R_ij ∩ π_ij (R_ik ⋈ D_k ⋈ R_kj) /* (REVISE-3((i, j), k))
      if R_ij changed then
      Q ← Q ∪ {(l, i, j) (l, j, i) | 1 ≤ l ≤ n, l ≠ i, l ≠ j}
endwhile
```

Solution:

Note: Since $X_4 > X_1 + X_2$ is a ternary constraint, PC-2 ignores it. However, we can induce that $X_4 > X_1$ and $X_4 > X_2$. Consider these two constraints instead when applying PC-2.

Initial Constraints:

R_{12}:{all domain values combinations}
R_{13}:{(3,3),(4,4),(7,7)}
R_{14}:{((1,6),(1,7),(1,8),(2,6),(2,7), (2,8),(3,6),(3,7),(3,8),(4,6),(4,7),(4,8),(7,8)}
R_{23}:{all domain values combinations}
R_{24}:{(2,6),(2,7),(2,8),(4,6),(4,7),(4,8),(6,7),(6,8)}
R_{34}:{all domain values combinations}

Initial Queue Q with triplets {(2,1,3),(2,1,4),(3,1,4),(1,2,3),(1,2,4),(3,2,4),(1,3,2),(1,3,4), (2,3,4),(1,4,2),(1,4,3),(2,4,3)} with 12 elements

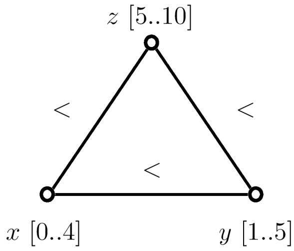
	Triplet	Relation update	Additions to Q
1	(2,1,3)	R_{23}-=$\{(*,5),(*,8),(*,9)\}$	(1,2,3),(4,2,3),(1,3,2),(4,3,2)
2	(2,1,4)	–	–
3	(3,1,4)	R_{34}-=$\{(5,*),(8,*),(9,*),(7,6),(7,7)\}$	(1,3,4),(2,3,4),(1,4,3),(2,4,3)
4	(1,2,3)	–	–
5	(1,2,4)	–	–
6	(3,2,4)	–	–
7	(1,3,2)	R_{12}-=$\{(1,*),(2,*)\}$	(3,1,2), (4,1,2),(3,2,1),(4,2,1)
8	(1,3,4)	R_{14}-=$\{(1,*),(2,*)\}$	(2,1,4), (3,1,4),(2,4,1),(3,4,1)
9	(2,3,4)	–	–
10	(1,4,2)	R_{12}-=$\{(*,8)\}$	(3,1,2),(4,1,2), (3,2,1),(4,2,1)
11	(1,4,3)	–	–
12	(2,4,3)	R_{23}-=$\{(*,5),(*,8),(*,9),(8,*)\}$	(1,2,3),(4,2,3),(1,3,2),(4,3,2)
13	(4,2,3)	–	–
14	(4,3,2)	–	–
15	(3,1,2)	–	–
16	(4,1,2)	–	–
17	(3,2,1)	–	–
18	(4,2,1)	–	–
19	(2,1,4)	–	–
20	(3,1,4)	–	–
21	(2,4,1)	–	–
22	(3,4,1)	–	–
23	(1,2,3)	–	–
24	(1,3,2)	–	–

Note: Since Rev-3((x,y),z) = Rev-3((y,x),z) there is no need to add (y,z,x) if (x,z,y) is already in Q (the opposite also holds)

Final Constraints:
R_{12}:{(3,2),(3,4),(3,6),(4,2),(4,4),(4,6),(7,2),(7,4),(7,6)}
R_{13}:{(3,3),(4,4),(7,7)}
R_{14}:{(3,6),(3,7),(3,8),(4,6),(4,7),(4,8),(7,8)}
R_{23}:{(2,3),(2,4),(2,7),(4,3),(4,4),(4,7),(6,3),(6,4),(6,7)}
R_{24}:{(2,6),(2,7),(2,8),(4,6),(4,7),(4,8),(6,7),(6,8)}
R_{34}:{(3,6),(3,7),(3,8),(4,6),(4,7),(4,8),(7,8)}

3) Consider the following network illustrated in the figure.

- Variables: x, y, z
- Domains: $D_x = [0..4], D_y = [1..5], D_z = [5..10]$
- Constraints: $x < y$, $y < z$, $x < z$.



a. Explain why it is directional path-consistent considering ordering x, y $\prec$ z.

b. Explain why it is NOT directional path-consistent considering ordering x, z $\prec$ y.

c. Show a partial assignment to the variables such that backtracking is required with ordering x, z $\prec$ y.

Solution:

a. Considering the ordering x,y $\prec$ z, for every pair of values that x and y can have with respect to the constraint x < y, there is a value for z that respects the constraints y < z and x < z.

b. Considering the ordering x,z $\prec$ y, there are pairs of values for x and z that break the constraints x < y and y < z.

c. x=4, z=5

III. Search in CSP (2.5 + 1.5 = 4/20)

1) Consider the graph with 8 nodes $A_1, A_2, A_3, A_4, H, T, F_1, F_2$. A_i is connected to A_{i+1} for all i, each A_i is connected to H, H is connected to T, and T is connected to each F_i. Find a 3-coloring of this graph by hand using the following strategy: backtracking with conflict-directed backjumping, the variable order $A_1, H, A_4, F_1, A_2, F_2, A_3, T$, and the value order R, G, B.

Solution:

a. A_1 = R.

b. H = R conflicts with A_1.

c. H = G.

d. A_4 = R.

e. F_1 = R.

f. A_2 = R conflicts with A_1, A_2 = G conflicts with H, so A_2 = B.

g. F2 =R.

h. A_3 = R conflicts with A_4, A3 = G conflicts with H, A3 = B conflicts with A2, so backtrack. Conflict set is $\{A_2, H, A_4\}$, so jump to A2. Add $\{H, A_4\}$ to A_2's conflict set.

i. A_2 has no more values, so backtrack. Conflict set is $\{A_1, H, A_4\}$ so jump back to A_4. Add $\{A_1, H\}$ to A_4's conflict set.

j. A_4 = G conflicts with H, so A_4 =B.

k. F_1 = R

l. A_2 = R conflicts with A_1, A_2 = G conflicts with H, so A_2 = B.

m. F_2 = R

n. A_3 = R.

o. T = R conflicts with F_1 and F_2, T = G conflicts with H, so T = B.

p. Success! (A_1 = R, H = G, A_4 = B, F_1 = R, A_2 = B, F_2 = R, A_3 = R, T = B)

2) Consider a simple version of WALKSAT as illustrated in the figure. Illustrate the use of WALKSAT until a solution is found, with the occurrence of at least two violated constraints, for the CNF formula $\varphi = \{(A \vee B), (\neg B \vee C \vee D), (\neg C), (\neg A \vee \neg D)\}$.

procedure WALKSAT

Input: A network $\mathcal{R} = (X, D, C)$, number of flips MAX_FLIPS, MAX_TRIES, probability p.

Output: "True," and a solution, if the problem is consistent, "false," and an inconsistent best assignment, otherwise.

1. **for** i = 1 to MAX_TRIES **do**
2. **start** with a random initial assignment $\bar{a}$.
3. Compare best assignment with $\bar{a}$ and retain the best.
4. **for** i = 1 to MAX_FLIPS
   - **if** $\bar{a}$ is a solution, **return** true and $\bar{a}$.
   - **else,**
     i. **pick** a violated constraint C, randomly
     ii. **choose** with probability p a variable-value pair $\langle x, a' \rangle$ for $x \in scope(C)$, or, with probability $1 - p$, choose a variable-value pair $\langle x, a' \rangle$ that minimizes the number of new constraints that break when the value of x is changed to a' (minus 1 if the current constraint is satisfied).
     iii. Change x's value to a'.
5. **endfor**
6. **return** false and the best current assignment.

Solution:

A possible execution of WALKSAT with at least two violated constraints could be the following:

Random assignment: {(A,True),(B,False),(C,True),(D,True)}
Pick violated constraint: $(\neg C)$
Flip assignment: C = False
Pick violated constraint: $(\neg A \vee \neg D)$
Flip assignment: D = False
Assignment {(A,True),(B,False),(C,False),(D,False)} is a SOLUTION!

IV. Plan Space Planning (2 + 2 = 4/20)

1) Consider a 2-digit binary counter that starts at 00, so that we want to get to 11. More precisely, $s_0 = \{d_2 = 0, d_1 = 0\}$ and $g = \{d_2 = 1, d_1 = 1\}$. There are two action templates, *incr-x0-to-x1* and *incr-01-to-10*, being *incr-x0-to-x1* defined as follows:

```
incr-x0-to-x1
pre:  d1=0
eff:  d1=1
```

a. Define action `incr-01-to-10`.

b. List the sequence of actions that are part of the plan for solving this problem.

Solution:

a.
```
incr-01-to-10
pre:  d2=0, d1=1
eff:  d2=1, d1=0
```

b. `incr-x0-to-x1, incr-01-to-10, incr-x0-to-x1`

2) Consider a simple planning problem in which there are three locations L={home, bakery, florist} and two products P = {bread, flowers}, that is, B = L ∪ P. There are no rigid relations. There are two action templates,

```
go(l,m)                          buy(p,l)
pre:  at(l)                      pre:  at(l), sells(l,p)
eff:  at(m), ¬ at(l)             eff:  have(p)
```

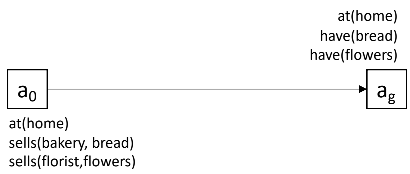
where l,m ∈ L and p ∈ P. The initial state s_0 and the goal g are

```
s0 ={at(home), sells(bakery,bread), sells(florist, flowers)};
g = {at(home), have(bread), have(flowers)}.
```

In the context of Plan-Space Search (PSP):

a. Write down the initial partial plan containing dummy actions a_0 and a_g that represent s_0 and g, respectively.

Solution:



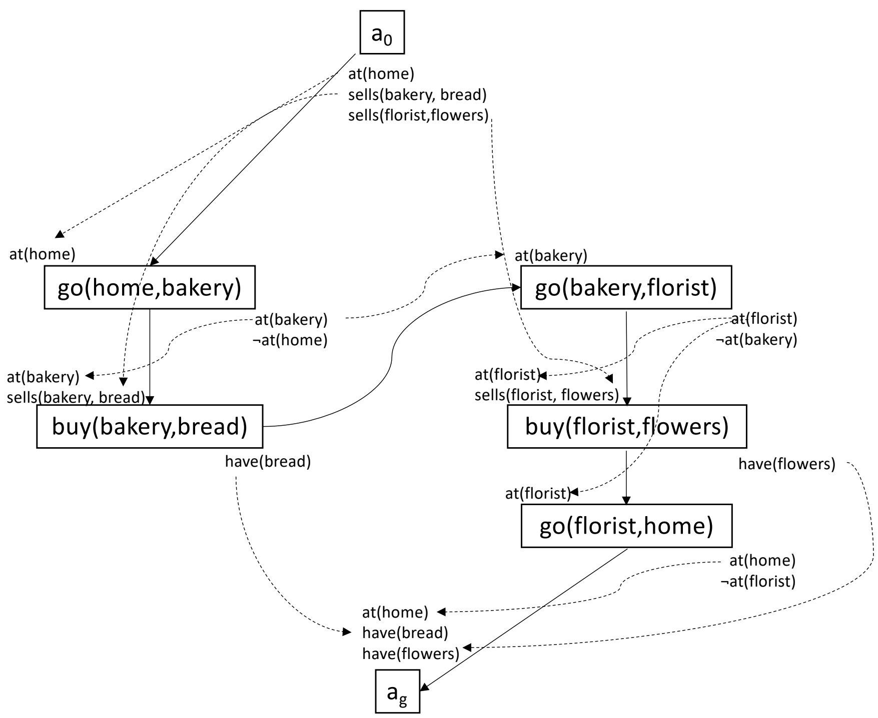
b. Identify the flaws in the initial plan.

Solution:

There are two flaws corresponding to open goals: have(bread) and have(flowers).

c. Apply PSP to the initial partial plan.

Solution:



V. Temporal Planning (2 + 2 = 4/20)

1) Consider a timeline $(\mathcal{T}, \mathcal{C})$. **Justify** the sentence "If $\mathcal{C}$ is consistent then $(\mathcal{T}, \mathcal{C})$ may or may not be consistent". **Give an example** for each one of the two situations.

Solution:

An instance of $(\mathcal{T}, \mathcal{C})$ is consistent if it satisfies all the constraints in C and does not specify two different values for a state variable at the same time. A timeline $(\mathcal{T}, \mathcal{C})$ is consistent if its set of consistent instances is not empty.

Consider two examples to illustrate both cases. For the two examples, consider $\mathcal{C} = (t_1 < t_3 < t_4 < t_2)$.

First, consider $\mathcal{T} = \{[t_1, t_3] loc(r_1) : (l_1, l_2), [t_4, t_2] loc(r_1) = l_2\}$, where r_1, l_1, l_2 are constants. There is at least one instance that is consistent, e.g. with $t_1 = 0, t_3 = 1, t_4 = 2, t_2 = 3$. Hence, $(\mathcal{T}, \mathcal{C})$ is consistent as well.

Now, consider $\mathcal{T} = \{[t_1, t_2] loc(r_1) = l_1, [t_3, t_4] loc(r_1) = l_2\}$. $\mathcal{T}$ is inconsistent because $loc(r_1)$ has two values during $[t_3, t_4]$, no matter the grounding of $(\mathcal{T}, \mathcal{C})$. Hence, $(\mathcal{T}, \mathcal{C})$ is inconsistent.

2) Run algorithm Path-Consistency (PC) on the following temporal network. Briefly **explain** the meaning of the output.

> $\mathsf{PC}(\mathcal{V}, \mathcal{E})$
>   for $k = 1, \ldots, n$ do
>   for each pair i, j such that $1 \leq i < j \leq n, i \neq k$, and $j \neq k$ do
> $r_{ij} \leftarrow r_{ij} \cap [r_{ik} \bullet r_{kj}]$
>   if $r_{ij} = \varnothing$ then return *inconsistent*

$$\begin{tikzcd}
t_1 \arrow[r, "{[1,5]}"] \arrow[d, "{[2,6]}"'] & t_2 \arrow[d, "{[4,7]}"] \arrow[dl, "{[5,8]}"] \\
t_4 & t_3 \arrow[l, "{[4,6]}"]
\end{tikzcd}$$

Solution:

```
k: 1 i: 2 j: 3 || Rij /\ [Rik * Rkj]: [4, 7] || Rij: [4, 7] || Rik * Rkj: [-inf, inf] || Rik: [-5, -1] || Rkj: [-inf, inf]
k: 1 i: 2 j: 4 || Rij /\ [Rik * Rkj]: [5, 5] || Rij: [5, 8] || Rik * Rkj: [-3, 5] || Rik: [-5, -1] || Rkj: [2, 6]
k: 1 i: 3 j: 4 || Rij /\ [Rik * Rkj]: [4, 6] || Rij: [4, 6] || Rik * Rkj: [-inf, inf] || Rik: [-inf, inf] || Rkj: [2, 6]
k: 2 i: 1 j: 3 || Rij /\ [Rik * Rkj]: [5, 12] || Rij: [-inf, inf] || Rik * Rkj: [5, 12] || Rik: [1, 5] || Rkj: [4, 7]
k: 2 i: 1 j: 4 || Rij /\ [Rik * Rkj]: [6, 6] || Rij: [2, 6] || Rik * Rkj: [6, 10] || Rik: [1, 5] || Rkj: [5, 5]
k: 2 i: 3 j: 4 || Rij /\ [Rik * Rkj]: [4, 1] || Rij: [4, 6] || Rik * Rkj: [-2, 1] || Rik: [-7, -4] || Rkj: [5, 5]
INCONSISTENT! Rij /\ [Rik * Rkj] = \{\}
```

The output means that the temporal network is inconsistent, i.e. there is no consistent assignment to t_1, t_2, t_3, t_4.