INSTITUTO SUPERIOR TÉCNICO

Search and Planning

2022/2023 Academic Year 1st Period

1st Exam November 6, 2023 Duration: 2h

- This is a closed book exam.
- To ensure equal conditions for all students, no answers will be given to questions asked during the exam.
- Ensure that your name and number are written on all pages.

EXAM SOLUTION

I. Modeling with CSP ((1.5+0.5)+2 = 4/20)

1) Recall a few notes about cryptarithmetic problems:

- Each letter is assigned a digit, with different letters being assigned different digits.
- A solution to a problem is constrained by the arithmetic operation.
- A number cannot start with digit 0.

Now consider the following cryptarithmetic problem instance: GO + TO = OUT.

a. Formulate it as a constraint network, identifying variables, domains, and constraints.

b. Infer the assignments to the variables that would be identified by a constraint solver.

Solution:

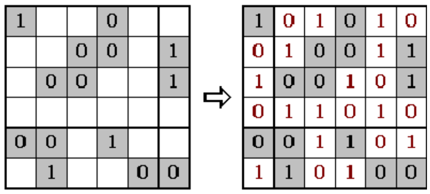
a. Variables: $\{G,O,T,U,C_1,C_2\}$. Domains: $D_{G,O,T,U} = \{0..9\}$ and $D_{C_1,C_2} = \{0,1\}$, Constraints: $G \neq 0, T \neq 0, O \neq 0, O+O = T+10\times C_1, G+T+C_1 = U+10\times C_2, O = C_2$, AllDifferent$(G,O,T,U)$.

b. Since O = 1, then O + O = 1 + 1 = 2. So, T = 2. G + 2 = 10 + U. If G = 9, then U = 1, which is not valid since O = 1. So, G = 8 and U = 0. Hence, O + U + T = 1 + 0 + 2 = 3. Final assignments: O=1, G=8, T=2, U=0.

2) Now consider the problem of binary puzzles. A binary puzzle is represented by an $n \times n$ grid, and we need to fill it with 0s or 1s somehow that three constraints must be satisfied:

i. The number of 1s and 0s must be the same in a row or column.
ii. All rows must be different from each other. And columns too.
iii. There must not be more than two consecutive 1s and 0s in a column or row.

The figure illustrates an example of a binary puzzle, where the initial grid and the solution grid are shown.



Propose a formulation of the problem as a constraint network, identifying variables, domains, and constraints.

Solution:
Variables $X_{i,j}$ with $1 \leq i, j \leq n$.

Domains $D_X = \{0, 1\}$.

Constraints (i):
$\forall_{1 \leq i \leq n} \sum_{j=1}^{n} X_{i,j} = n/2$
$\forall_{1 \leq j \leq n} \sum_{i=1}^{n} X_{i,j} = n/2$

Constraints (ii):
$\forall_{a,b}\ s.t. 1 \leq a < b \leq n : (X_{a,1}, ..., X_{a,n}) \neq (X_{b,1}, ..., X_{b,n})$
$\forall_{a,b}\ s.t. 1 \leq a < b \leq n : (X_{1,n}, ..., X_{n,a}) \neq (X_{1,b}, ..., X_{n,b})$

Constraints (iii):
$\forall_{1 \leq i \leq n} \forall_{1 \leq j \leq n-2}\ 1 \leq X_{i,j} + X_{i,j+1} + X_{i,j+2} \leq 2$
$\forall_{1 \leq i \leq n-2} \forall_{1 \leq j \leq n}\ 1 \leq X_{i,j} + X_{i+1,j} + X_{i+2,j} \leq 2$

II. Inference in CSP (2 + 2 = 4/20)

1) Suppose we have the following CSP:

- Variables: R, S, T
- Domains: $D_R = D_S = D_T = \{0, 1, 2\}$
- Constraints: $R \neq S, S < T, R = \lfloor T/2 \rfloor$ (i.e. R = floor(T/2).

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

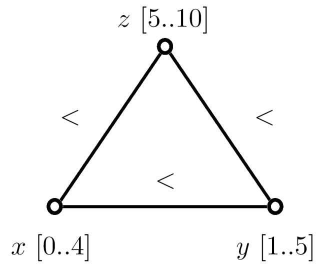
```
AC-3(ℛ)
Input: A network of constraints ℛ = (X, D, C).
Output: ℛ', which is the largest arc-consistent network equivalent to ℛ.
for every pair {x_i, x_j} that participates in a constraint R_ij ∈ ℛ
    queue ← queue ∪ {(x_i, x_j), (x_j, x_i)}
endfor
while queue ≠ {}
    select and delete (x_i, x_j)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

Initial Queue Q = {(R,S),(S,R),(T,S),(S,T),(T,R),(R,T)}

Pair	Constraint	Consistent?	Domain update	Additions to Q
(R,S)	$R \neq S$	yes	–	–
(S,R)	$R \neq S$	yes	–	–
(T,S)	$T > S$	no	D_T -= $\{0\}$	+{(R,T)} already in Q
(S,T)	$S < T$	no	D_S -= $\{2\}$	+{(R,S)}
(T,R)	$R = \lfloor T/2 \rfloor$	yes		–
(R,T)	$R = \lfloor T/2 \rfloor$	no	D_R -= $\{2\}$	+{(S,R)}
(R,S)	$R \neq S$	yes	–	–
(S,R)	$R \neq S$	yes	–	–

2) 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: Constraints: $x < y$, $y < z$, $x < z$

Show the result of applying path-consistency to the network. For your convenience, the PC-2 algorithm is given below.

PC-2($\mathcal{R}$)
Input: A network $\mathcal{R} = (X, D, C)$.
Output: $\mathcal{R}'$ a path-consistent network equivalent to $\mathcal{R}$.
$Q \leftarrow \{(i, k, j) \mid 1 \leq i < j \leq n,\ 1 \leq k \leq n,\ k \neq i,\ k \neq j\}$
while Q is not empty
    select and delete a 3-tuple (i, k, j) from Q
    $R_{ij} \leftarrow R_{ij} \cap \pi_{ij} (R_{ik} \bowtie D_k \bowtie R_{kj})$ /* (REVISE-3($(i, j), k$))
    if R_{ij} changed then
    $Q \leftarrow Q \cup \{(l, i, j)\ (l, j, i) \mid 1 \leq l \leq n,\ l \neq i,\ l \neq j\}$
endwhile

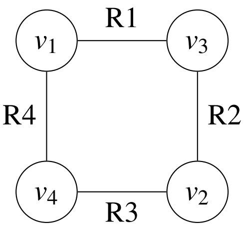
Solution:

Step	Triplet	Relation update	Additions to Q
1	(x,y,z)	$R_{xz} \leftarrow R_{xz} \setminus \{(4,5)\}$	(y,x,z), (y,z,x)
2	(x,z,y)	none	—
3	(y,x,z)	none	—
4	(y,z,x)	none	—

Explanation for removing tuple (4,5) from R_{xz}:
$R_{x,z} = \{(x,z) | x < z, x \in [0..4], y \in [5..10]\}$. But for $4 \in [0..4]$ and $5 \in [5..10]$ there is no $y \in [1..5]$ s.t. $4 < y$ and $y < 5$.

III. Search in CSP ((1.5 + 1) + 1.5 = 4/20)

1) Consider the following constraint network.



Variables: $V = \{v_1, v_2, v_3, v_4\}$
Domains: $D(v_i) = \{0,1,2,3\}$
Constraints: $R1 : v_1 \neq v_3, R2 : v_3 \neq v_2, R3 : v_2 \neq v_4, R4 : v_4 = v_1 - 1$

a. **Perform** backtracking using the *graph-based backjumping algorithm*. The variables and values must be chosen using ascending alphabetical order. Indicate the induced ancestors where applicable.

b. **Discuss** the drawback of graph-based backjumping using this constraint network.

Solution:

a.

Order: v_1, v_2, v_3, v_4

1	$v_1 = 0$	$I(v_1) = \emptyset$
2	$v_2 = 0$	$I(v_2) = \emptyset$
3	$v_3 = \cancel{0}, 1$	$I(v_3) = \{v_1, v_2\}$
4	$v_4 = \cancel{0}, \cancel{1}, \cancel{2}, \cancel{3}$	$I(v_4) = \{v_1, v_2\}$
5	$v_2 = 1$	$I(v_2) = \emptyset$
6	$v_3 = \cancel{0}, \cancel{1}, 2$	$I(v_3) = \{v_1, v_2\}$
7	$v_4 = \cancel{0}, \cancel{1}, \cancel{2}, \cancel{3}$	$I(v_4) = \{v_1, v_2\}$
8	$v_2 = 2$	$I(v_2) = \emptyset$
9	$v_3 = \cancel{0}, 1$	$I(v_3) = \{v_1, v_2\}$
10	$v_4 = \cancel{0}, \cancel{1}, \cancel{2}, \cancel{3}$	$I(v_4) = \{v_1, v_2\}$
11	$v_2 = 3$	$I(v_2) = \emptyset$
12	$v_3 = \cancel{0}, 1$	$I(v_3) = \{v_1, v_2\}$
13	$v_4 = \cancel{0}, \cancel{1}, \cancel{2}, \cancel{3}$	$I(v_4) = \{v_1, v_2\}$
14	$v_2 = \emptyset$	$I(v_2) = \{v_1\}$
15	$v_1 = 1$	$I(v_1) = \emptyset$
16	$v_2 = 0$	$I(v_2) = \emptyset$
17	$v_3 = \cancel{0}, \cancel{1}, 2$	$I(v_3) = \{v_1, v_2\}$
18	$v_4 = \cancel{0}, \cancel{1}, \cancel{2}, \cancel{3}$	$I(v_4) = \{v_1, v_2\}$
19	$v_2 = 1$	$I(v_2) = \emptyset$
20	$v_3 = 0$	$I(v_3) = \{v_1, v_2\}$
21	$v_4 = 0$	$I(v_4) = \{v_1, v_2\}$

b. The drawback is the so-called thrashing, i.e. the assignment $v_1 = 0$ has been followed by assigning different combinations of values to v_2, v_3, v_4 to end up concluding that any solution requires $v_1 \neq 0$. Conflict-directed backjumping would have avoided this situation.

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 $\varphi = \{(P \vee Q \vee \neg R), (\neg P), (\neg Q \vee R)\}$.

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.

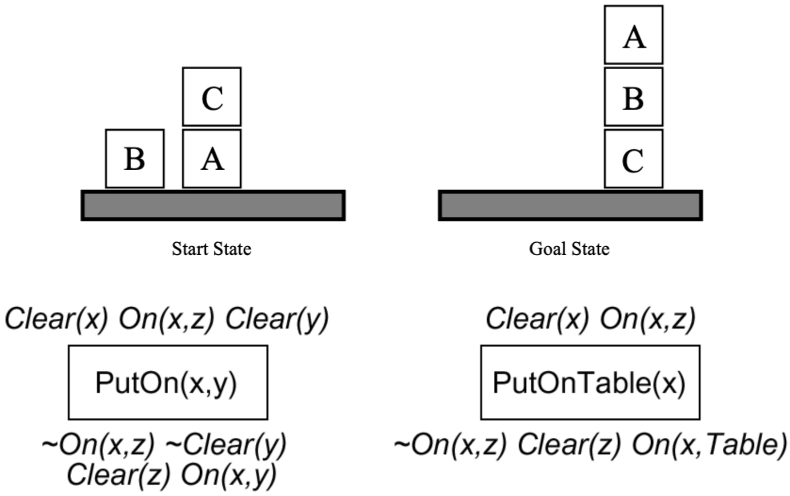
1. **for** i = 1 to MAX_TRIES **do**
2. **start** with a random initial assignment $\bar{a}$.
3. Compare best assignment with $\bar{a}$ and retain the best.
4. **for** i = 1 to MAX_FLIPS
   - **if** $\bar{a}$ is a solution, **return** true and $\bar{a}$.
   - **else,**
     - i. **pick** a violated constraint 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:

Random initial assignment for variables P, Q, R = {T,T,F}
Pick violated constraint $(\neg P)$ and get assignment {F,T,F}
Pick violated constraint $(\neg Q \vee R)$ and get solution assignment {F,T,T}

IV. Plan Space Planning (1 + 3 = 4/20)

1) Consider the following start and goal states for the planning problem illustrated below, as well as two actions.

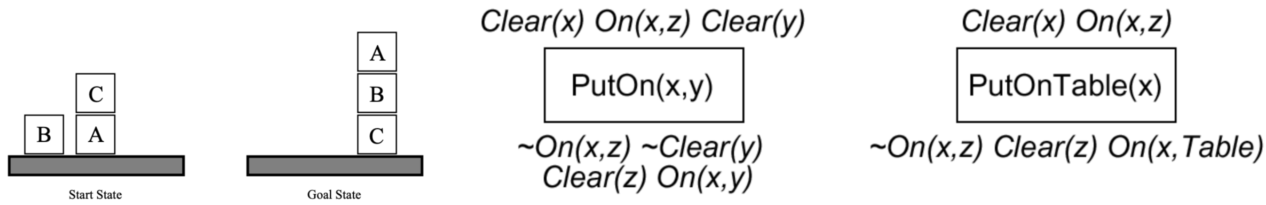


Define the initial state and the goal state as a set of predicates.

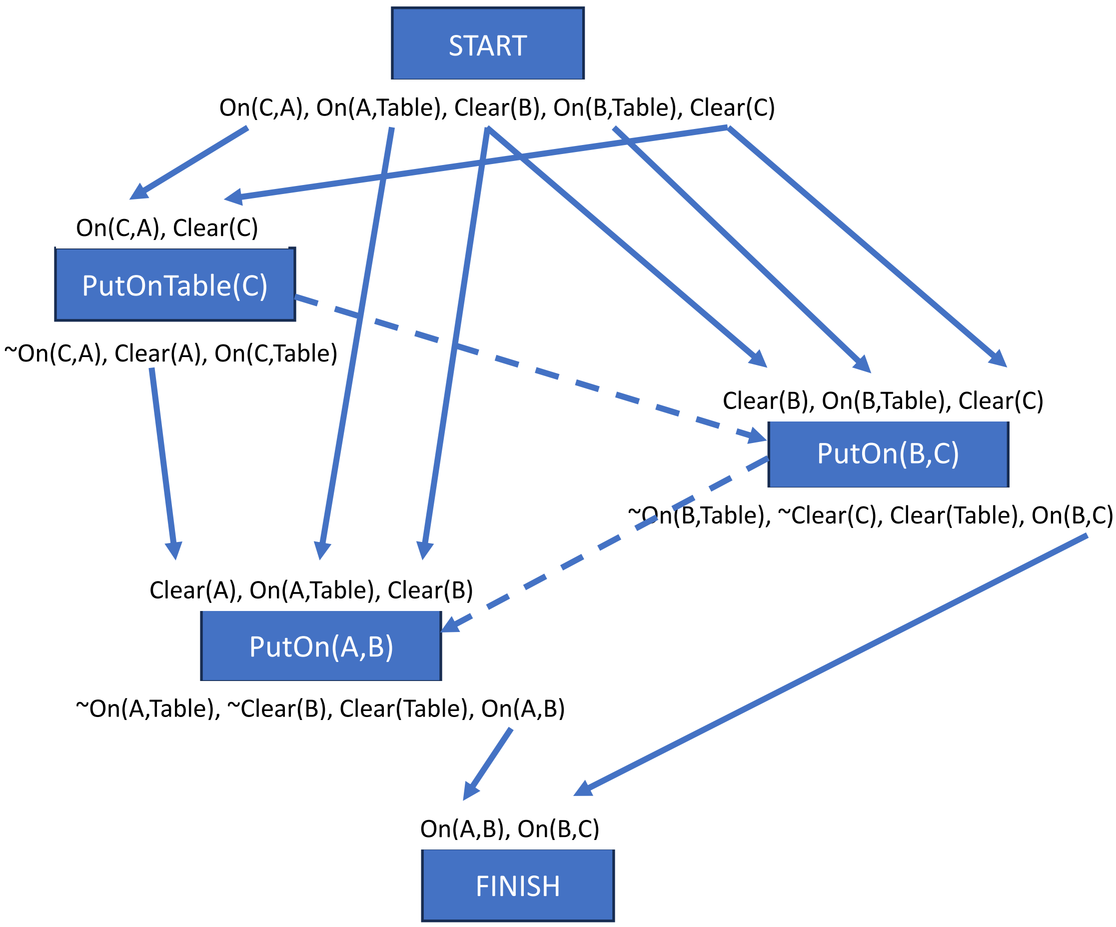
Solution:

Initial state: On(A,Table), On(B,Table), On(C,A), Clear(B), Clear(C)
Goal state: On(A,B), On(B,C)

2) **Illustrate** the plan resulting from applying Plan-Space Search (PSP) to the planning problem defined before.



Solution:



V. Temporal Planning ((1 + 1) + 2 = 4/20)

1) Consider the timeline $(\{[t_1,t_2]loc(r) = loc1, [t_3,t_4]loc(r1) = l\}, \{t_1 < t_2, t_3 < t_4\})$.

a. **Explain** why the timeline is consistent but not secure.

Solution:

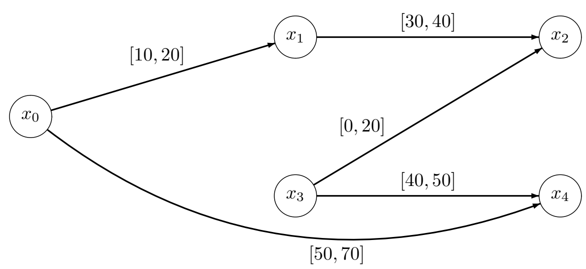
The timeline is consistent because there is a ground instance such that (i) it satisfies all the constraints in C and (ii) it does not specify two different values for a state variable at the same time. For example, assume instance r=r2 and $t_1 = 1, t_2 = 2, t_3 = 3, t_4 = 4$. However, the timeline is not secure because we can find an instance that meets the constraints in C but is not consistent. For example, assume that r=r1, l=loc2 and $t_1 = 1, t_2 = 3, t_3 = 2, t_4 = 4$. In this case, r1 is at two different locations (loc1 and loc2) between time 2 and 3.

b. **Suggest** TWO alternative additional constraints that make the timeline secure.

Solution:

$r \neq r1$ and $t_2 < t_3$

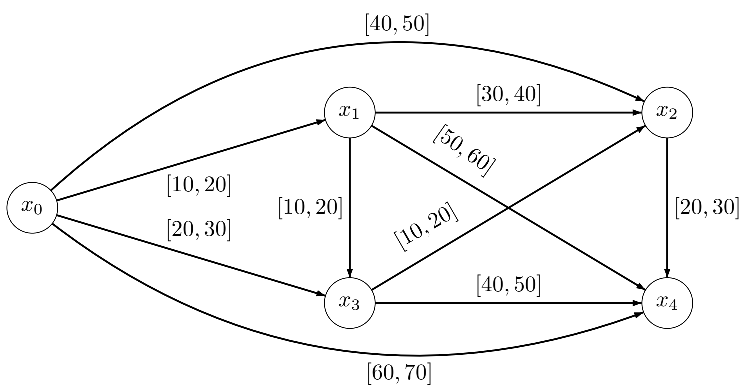
2) Consider the following temporal network.



Run algorithm Path-Consistency (PC) on the network.

```
PC(𝒱, ℰ)
  for k = 1, ..., n do
    for each pair i, j such that 1 ≤ i < j ≤ n, i ≠ k, and j ≠ k do
      r_ij ← r_ij ∩ [r_ik • r_kj]
      if r_ij = ∅ then return inconsistent
```

Solution:



k	i	j	R_{ik}	R_{kj}	$R_{ik} \cdot R_{kj}$	R_{ij}	$R_{ij} \leftarrow R_{ij} \cap (R_{ik} \cdot R_{kj})$
0	1	2	[-20,-10]	$[-\infty,+\infty]$	$[-\infty,+\infty]$	[30,40]	[30,40]
0	1	3	[-20,-10]	$[-\infty,+\infty]$	$[-\infty,+\infty]$	$[-\infty,+\infty]$	$[-\infty,+\infty]$
0	1	4	[-20,-10]	[30,70]	[30,60]	$[-\infty,+\infty]$	[30,60]
0	2	3	$[-\infty,+\infty]$	$[-\infty,+\infty]$	$[-\infty,+\infty]$	[-20,0]	[-20,0]
0	2	4	$[-\infty,+\infty]$	[50,70]	$[-\infty,+\infty]$	$[-\infty,+\infty]$	$[-\infty,+\infty]$
0	3	4	$[-\infty,+\infty]$	[50,70]	$[-\infty,+\infty]$	[40,50]	[40,50]
1	0	2	[10,20]	[30,40]	[40,60]	$[-\infty,+\infty]$	[40,60]
1	0	3	[10,20]	$[-\infty,+\infty]$	$[-\infty,+\infty]$	$[-\infty,+\infty]$	$[-\infty,+\infty]$
1	0	4	[10,20]	[30,60]	[40,80]	[50,70]	[50,70]
1	2	3	[-40,-30]	$[-\infty,+\infty]$	$[-\infty,+\infty]$	[-20,0]	[-20,0]
1	2	4	[-40,-30]	[30,60]	[-10,30]	$[-\infty,+\infty]$	[-10,30]
1	3	4	$[-\infty,+\infty]$	[30,60]	$[-\infty,+\infty]$	[40,50]	[40,50]
2	0	1	[40,60]	[-40,-30]	[0,30]	[10,20]	[10,20]
2	0	3	[40,60]	[-20,0]	[20,60]	$[-\infty,+\infty]$	[20,60]
2	0	4	[40,60]	[-10,30]	[30,90]	[50,70]	[50,70]
2	1	3	[30,40]	[-20,0]	[10,40]	$[-\infty,+\infty]$	[10,40]
2	1	4	[30,40]	[-10,30]	[20,70]	[30,60]	[30,60]
2	3	4	[0,20]	[-10,30]	[-10,50]	[40,50]	[40,50]
3	0	1	[20,60]	[-40,-10]	[-20,50]	[10,20]	[10,20]
3	0	2	[20,60]	[0,20]	[20,80]	[40,60]	[40,60]
3	0	4	[20,60]	[40,50]	[60,110]	[50,70]	[60,70]
3	1	2	[10,40]	[0,20]	[10,60]	[30,40]	[30,40]
3	1	4	[10,40]	[40,50]	[50,90]	[30,60]	[50,60]
3	2	4	[-20,0]	[40,50]	[20,50]	[-10,30]	[20,30]
4	0	1	[60,70]	[-60,-50]	[0,20]	[10,20]	[10,20]
4	0	2	[60,70]	[-30,-20]	[30,50]	[40,60]	[40,50]
4	0	3	[60,70]	[-30,-40]	[10,30]	[20,60]	[20,30]
4	1	2	[50,60]	[-30,-20]	[20,40]	[30,40]	[30,40]
4	1	3	[50,60]	[-50,-40]	[0,20]	[10,40]	[10,20]
4	2	3	[20,30]	[-50,-40]	[-30,-10]	[-20,0]	[20,-10]