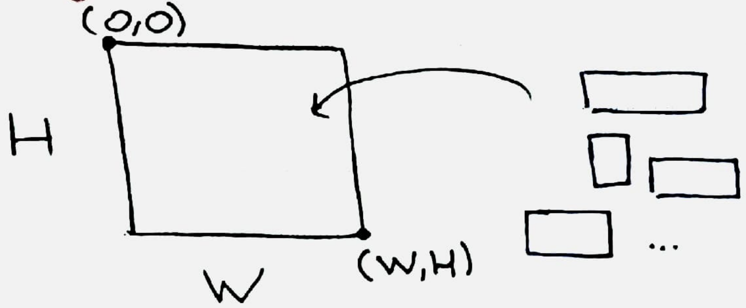
MI Sheet 5

6.4. Give a Precise Formulation (domain, Variables, Constraints) to each of the Following CSPs

1. Rectilinear Floor Planning

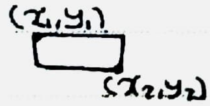
i.e., given a set of small rectangles Put them in a larger one (Floor of room) with no overlap



1. Variables

→ Associate with each rectangle i a variable $(x_1^i, y_1^i, x_2^i, y_2^i)$

Where (x_1, y_1) is the top-left corner loc. and (x_2, y_2) is the bottom-right corner loc. (in the room)



. This is exactly sufficient as its all that's needed to Place a rectangle somewhere in the room horizontally or vertically.

2. domains

$x_1^i, x_2^i, y_1^i, y_2^i \in \mathbb{R}$ for any rectangle i

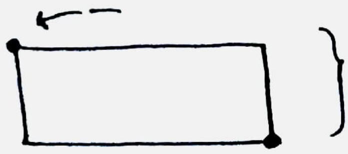
3. Constraints

$$C_1^i : (x_2^i - x_1^i = w^i \wedge y_2^i - y_1^i = h^i)$$
$$\vee (x_2^i - x_1^i = h^i \wedge y_2^i - y_1^i = w^i)$$

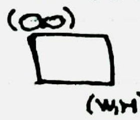
For any rectangle i with width, height $= w^i, h^i$ resp.

$$C_2^i : (x_1^i \geq 0 \wedge y_1^i \geq 0 \wedge x_2^i \leq W \wedge y_2^i \leq H)$$

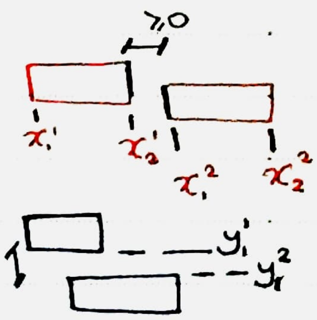
For any...



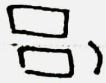
Should be within the room



$$C_3^{i,j} : (x_2^i \leq x_1^j \vee x_2^j \leq x_1^i)$$
$$\vee (y_2^i \leq y_1^j \vee y_2^j \leq y_1^i)$$



→ guaranteeing the seperation from either direction is enough. (e.g.

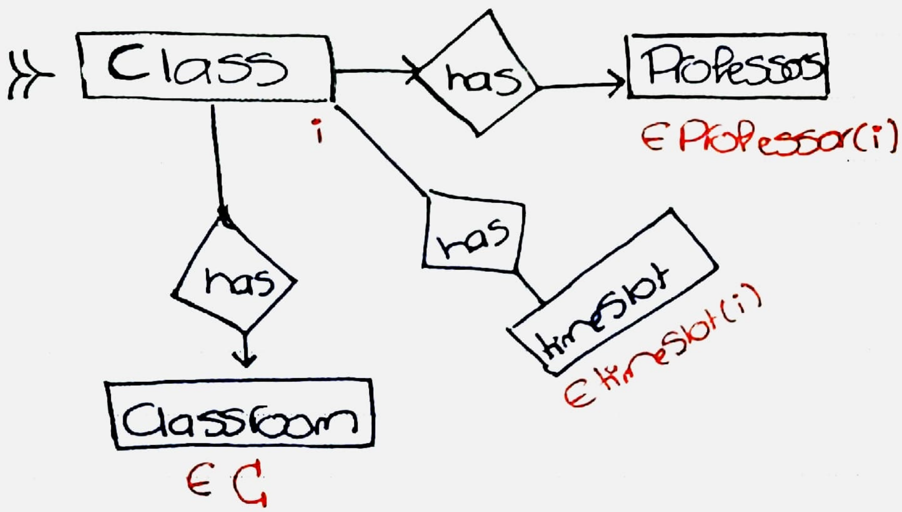


)

For any pair of rectangles i, j

b. Class Scheduling

→ Fixed no. of Professors and Classrooms
  - Call the sets P and C
→ List of Classes to be offered (L)
→ List of Possible time slots for each class
  - let timeslot(i) return them for class i
→ Each Professor has a set of Classes to teach
  - let Professor(i) return Profs for class i



- hence, to schedule a class we need to set all 3

⇒ Variables

$$V_i = (Room_i, Timeslot_i, Proffesor_i) \ \forall i \in L$$

⇒ Domain

$$V_i \in \{(r_i, t_i, P_i) \mid r_i \in C, t_i \in timeslot(i), P_i \in Prof(i)\}$$

⇒ Constraints

$C_{i,j}: (t_i = t_j) \longrightarrow (r_i \neq r_j \wedge P_i \neq P_j)$

. For any two classes i, j
if they take place at same time slots

the classrooms & the Professors assigned
should be diff.

C. Given a network cities connected by roads
Choose an order to visit all cities without
repeating any.

. let C be the (set) list of cities $\{C_1, C_2, \dots, C_n\}$
. let R be the set of roads $\{(C_i, C_j) \mid$ There's a road $\underset{C_i \rightarrow C_j}{}\}$

→ we have n cities

Variable

$V = (t_1, t_2, \dots, t_n)$ // an ordering of the cities

domain

$V \in \{(t_1, t_2, \dots, t_n) \mid t_1 \in C, t_2 \in C, \dots, t_n \in C\}$.

Constraints

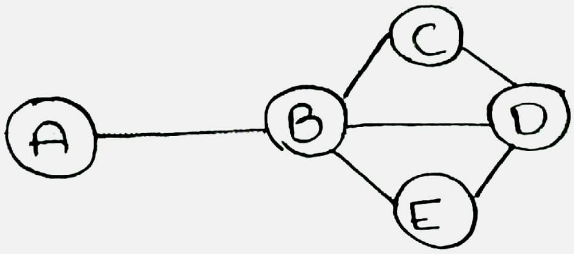
→ any two consecutive cities in V are connected by a road

$C_1 : (t_k = C_i \wedge t_{k+1} = C_j) \rightarrow ((C_i, C_j) \in R)$

. for some $C_i, C_j \in C$

must hold for $1 \leqslant k \leqslant n-1$

C_2 : Alldiff($t_1, t_2, \ldots, t_n$)

2. 

Possible Colors = {1,2,3}

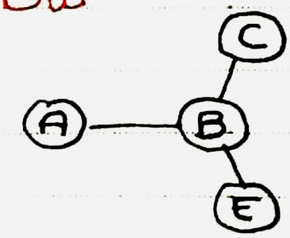
- Solve the CSP
  - → Use Backtracking with Forward check
  - → Use highest degree & ~~least Constraining value heuristic~~ (minimum rem. values) ☺ ouch

→ Initial Domains

A	B	C	D	E
1,2,3	1,2,3	1,2,3	1,2,3	1,2,3

- All variables have the same no. of values but B has the highest degree

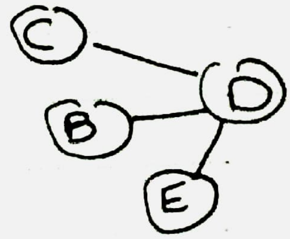
→ Assign B=1 & Forward check



A	B	C	D	E
2,3	1	2,3	2,3	2,3

- removed values inconsistent with B=1

- Didn't fail (no empty domain)
- All remaining variables have the same no. of values but D has highest degree



⇒ Assign D = 2 & Forward check
(Consistent with B=1)

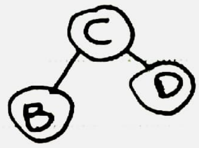
A	B	C	D	E
2, 3	1	3	2	3

highest

• C, E each has one value left and degree 2 (so can choose either)
least

⇒ Assign C = 3 & Forward check
(Consistent with B=1, D=2)

A	B	C	D	E
2, 3	1	3	2	3

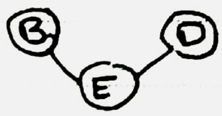


• Everything is consistent (nothing removed)

• E has less values

⇒ Assign E = 3 & Forward check
Consistent with...

A	B	C	D	E
2, 3	1	3	2	3



• only A is left

• Everything is consistent

⇒ Assign A = 2 & Forward check
Consistent with...

A	B	C	D	E
2	1	3	2	3

• Everything is consistent!

→ We've reached a complete & consistent assigned (solution) ⇒ $\{A=2, B=1, C=3, D=2, E=3\}$

- We never needed to backtrack, forward check & consistency checks were always succesful.
  - → Suppose either of them failed
    - Check the next value
    - → Suppose all values failed
      - backtrack to the last value assigned & check next value.

③ 3

6.5)

$$
\begin{array}{r}
+\ TWO \\
TWO \\
\hline
FOUR
\end{array}
$$

- Solve the cryptoarethmetic problem
- Use Backtracking with forward checking
- Use Minimum Rem. Values & least constraining values.

→ Variables & domains

$F, T \in \{1,2,3,4,5,6,7,8,9\}$
$W, O, U, R \in \{0,1,2,3,4,5,6,7,8,9\}$

$C_1, C_2, C_3 \in \{0,1\}$

- Recall

$$
\begin{array}{r}
\overset{C_3}{1}\,1\,\overset{C_1}{0} \\
234 \\
+\ 895 \\
\hline
1129
\end{array}
$$

→ Constraints

$Alldiff(F, T, W, O, U, R)$

$F = C_3$

variable 'O'

$2T + C_2 = O + 10C_3$

$2W + C_1 = U + 10C_2$

$2O = R + 10C_1$

new carry, old carry

$$\begin{array}{rl} C_3 & C_2 \\ + & T\ldots \\ & T\ldots \\ \hline & O \end{array}$$

• Minimum Remaining Values ⇒ C_1, C_2, C_3 have only two elements
⟶ choose C_3 arbitrarily

$C_3 = 0$	$C_3 = 1$
• $F = C_3$ → remove all values in dom. of F not consistent with $C_3 = 0$ $F \in \{1, 2, \ldots, 9\}$ becomes $F \in \{\}$ Failure → Check next value	• $F = C_3$ Likewise. $F \in \{1\}$ • $10C_3 + O = 2T + C_2$ next Page

$C_3 = 1$

→ have two Constraints with C_3

. After forward checking with the first, $F \in \{1\}$

→ 2nd Constraint: $10C_3 + O = 2T + C_2$

- Can we use it to reduce any domains

→ Consider $C_2 = 0$

- $10 + O = 2T$

letter, zero

- $T = 5 + \frac{O}{2}$, Since $O \geq 0$

then $T \geq 5$

- i.e., Can reduce its domain to $T \in \{5, 6, \ldots, 9\}$

→ Consider $C_2 = 1$

- $T = 4.5 + \frac{O}{2}$, $T \geq 4.5$

. no further reduction

. So far assigned $C_3 = 1$

. Next Variable with minimum rem. Values is F

\# $F = 1$ $\quad$ $\underbrace{\text{only Value}}_{\text{domain}}$ to assign

. Consistent with $C_3 = 1$ (assignment so far)

Constraints involving F

$F = C_3$ ✓

Alldiff (F, T, W, O, U, R)

- Now $W, O, U, R \in \{0, 2, 3, \ldots, 9\}$ (removed 1)

- So far assigned $\{C_3 = 1, F = 1\}$

- C_1, C_2 have minimum rem. values

  → Choose C_2 arbitrarily

$C_2 = 0$

- Forwardcheck $2T + C_2 = O + 10C_3$

$2T = O + 10$. Doesn't help reduce T's domain ($T = \frac{O}{2} + 5$)

$O = 2T - 10$

. Since $T \in \{5, 6, 7, 8, 9\}$ we can reduce O to $O \in \{0, 2, 4, 6, 8\}$

- Forward check $2W + C_1 = U + 10C_2$

$2W + C_1 = U$

$W = \frac{U - C_1}{2}$. we know $U \in \{0, 2, 3, .., 9\}$

$C_1 = 0$ $W \in \{0, 2, 3, 4\}$ ∪ $C_1 = 1$ $\in \{\}$ → $W \in \{0, 2, 3, 4\}$

$C_2 = 1$

- Forward Check $2T + C_2 = O + 10C_3$

$\rightarrow O = 2T - 9$

- Since $T \in \{5, 6, 7, 8, 9\}$

$\rightarrow O \in \{3, 5, 7, 9\}$

$2 \times 6 - 9$ $\quad 2 \times 7 - 9$ ...

$\rightarrow T = \frac{O}{2} + 4.5$

- Since $O \in \{0, 2, 3, \ldots, 9\}$

$\rightarrow T \in \{6, 7, 8, 9\}$

$\frac{3}{2} + 4.5 \quad \frac{5}{2} + 4.5$ ..

- Forward check $2W + C_1 = U + 10C_2$

$\rightarrow 2W + C_1 = U + 10$

- $W = 5 + \frac{U - C_1}{2}$ $\qquad U \in \{0, 2, 3, \ldots, 9\}$

$C_1 = 0$ $\qquad C_1 = 1$ $\qquad // \ 5 + \frac{U}{2} - \frac{1}{2}$

$W \in \{\}$ $\qquad W \in \{5, 6, 7, 8, 9\} \rightarrow W \in \{5, 6, 7, \ldots, 9\}$

- $4 + 4 + 5 = 13$ Possible values to set

- For $C_2 = 0$, had $5 + 5 + 4 = 14$

$\rightarrow$ Choose $C_2 = 0$

rest domains are common

- So far assigned $\{C_3=1, F=1, C_2=0\}$

→ Domains so far

$C_1 \in \{0,1\}$

$W \in \{0,2,3,4\}$
$O \in \{0,2,4,6,8\}$ } new by $C_2=0$
$T \in \{5,6,7,8,9\}$
$U, R \in \{0,2,3,4,\dots,9\}$

- Next minimum rem. value variable is C_1

$C_1=0$

- Forward check constraint $10C_2+U=2W+C_1$

→ $U=2W$
- $U \in \{0,2,3,4,\dots,9\} \xrightarrow[W=U/2]{} W \in \{0,2,3,4\}$

• All satisfy

- $W \in \{0,2,3,4\} \rightarrow U \in \{0,4,6,8\}$

• reduced

- Forward check $10C_1+R=2O$ ← letter

→ $R=2O$
$R \in \{0,2,3,4,\dots,9\} \xrightarrow[\frac{R}{2}]{} O \in \{0,2,4\}$

$O \in \{0,2,4,6,8\} \xrightarrow[2O]{} R \in \{0,4,8\}$

- 14 remaining values to set

$C_1 = 1$

- Forward Check $10C_2 + U = 2W + C_1$

  $\rightarrow U = 2W + 1$

  - leads to $U \in \{5, 7, 9\}$, $W \in \{2, 3, 4\}$

- Forward Check $10C_1 + R = 20$

  $\rightarrow 10 + R = 20$ ← letter

  - leads to $O \in \{6, 8\}$, $R \in \{2, 6\}$

  - 10 remaining values

$\Rightarrow$ Clearly has less values

- Assignment is now $\{C_3 = 1, F = 1, C_2 = 0, C_1 = 0\}$

Domains

$W \in \{0\ 2, 3, 4\}$
$U \in \{0, 4, 6, 8\}$
$O \in \{0, 2, 4\}$
$R \in \{0, 4, 8\}$
$T \in \{5, 6, 7, 8, 9\}$

$\rightarrow$ Can choose O or R for next variable
- let's go for O

$O = 0$ ← Two

- Forward Check $10C_1 + R = 20$

  → $R = 0$

  - $R \in \{0, 2, 4\}$ redues to $R \in \{0\}$
  - but R can't be 0 if O is 0 (alldiff)
  - $R \in \{\}$ (as if we forward checked alldiff)

  Failure

$O = 2$

- Forward Check $10C_1 + R = 20$

  → $R = 4$

  - $R \in \{4\}$ only left

- Forward check $O + 10C_3 = 2T + C_2$

  → $2 + 10 = 2T$

  - $T \in \{6\}$

- Forward check alldiff(F, T, W, O, U, R)

F	T	W	O	U	R
1	$\{6\}$	$\{0,3,4\}$	2	$\{0,4,6,8\}$	$\{4\}$

- remove 2 as O took it
- only remove from {}

TA

- 9 values

#O=4

- By Similarly Performing Forward checking

  $R \in \{8\}$
  $T \in \{7\}$
  $W \in \{0,2,3\}$
  $U \in \{0,6,8\}$

  · 8 Values → Choose $O=2$

- Assignment is now $\{C_3 = 1, F=1, C_2=0, C_1=0, O=2\}$

Domains
$T \in \{6\}$
$W \in \{0,3,4\}$
$U \in \{0,4,6,8\}$
$R \in \{4\}$

⇒ Assign T (minimum rem. val.)

#T=6

- Only Constraint relating T and unassigned variables is alldiff(F, T, W, O, U, R)

$$\mathrm{alldiff}(\underset{1}{F},\ \underset{6}{T},\ \underset{\{0,3,4\}}{W},\ \underset{2}{O},\ \underset{\{0,4,\cancel{6},8\}}{U},\ \underset{\{4\}}{R})$$

⇒ Assign R
#R=4

- Forward Checking alldiff(F, T, W, O, U, R)

$F \to 1$, $T \to 6$, $W \to \{0,3,4\}$, $O \to 2$, $U \to \{0,4,8\}$, $R \to 4$

→ Assignment $\{C_3=1, F=1, C_2=0, C_1=0, O=2, T=6, R=4\}$

Domains

$W \in \{0,3\}$

$U \in \{0,8\}$

⇒ Assign W

$W=0$

- $10C_2 + U = 2W + C_1$

→ $U = 2W = 0$

⟶ $U \in \{0\}$

- Alldiff (F, T, W, O, U, R)

→ $U \in \{\}$ $W \to 0$

Failure

$W=3$

- $10C_2 + U = 2W + C_1$

→ $U = 6$

$U \in \{\}$

Failure

- Backtrack to $R=4$ → no other value
- Backtrack to $T=6$ → no other value

- backtrack to $O=4$
  → Can take $O=2$ (last value in O's dom.)

Now Assignment $= \{C_3=1, F=1, C_2=0, C_1=0, O=\underline{4}\}$

Domains — 1/2 Pages ago

$T \in \{7\}$
$R \in \{8\}$
$W \in \{0, 2, 3\}$
$U \in \{0, 6, 8\}$

- Next assign $T=7$
  - Alldiff leads to (last constr. in T)
    $W \in \{0,2,3\}$, $U \in \{0,6,8\}$, $R \in \{8\}$ (no change)
- Next assign $R=8$
  - Alldiff leads to
    $W \in \{0,2,3\}$, $U \in \{0,6,\cancel{8}\}$
- Next assign U

$U=0$
- $10C_2 + U = 2W + C_1$
  $W=0$
- $W \in \{0\}$ left
- Alldiff
  $W \in \{\}$ (U=0 already) $\boxed{\text{fail}}$

$U=6$
- $10C_2 + U = 2W + C_1$
  $W=3$
- $W \in \{3\}$ left
- Alldiff ✓

Now assignment = $\{C_3=1, F=1, C_2=0, C_1=0, O=4,$
$T=7, R=8, U=6\}$

Domains

$W \in \{3\}$

→ Assign $W=3$

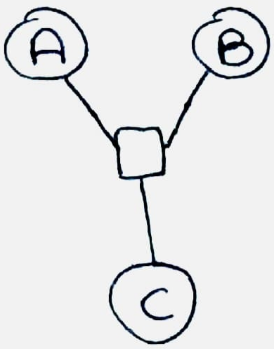
- No constraints that relate W to unassigned variables (forward checking)
- Done, we now have a complete & consistent assignment

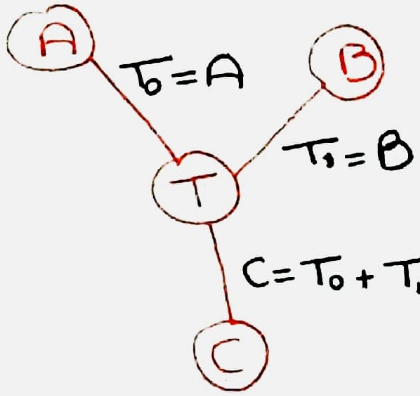
$$
\begin{array}{r}
734 \\
+\ 734 \\
\hline
1468
\end{array}
$$

6.6. Show that a ternary constraint such as $A+B=C$ can be turned into 3 binary constraints using an auxiliary variable

$$A+B=C \longrightarrow T=(T_0,T_1)$$ one variable

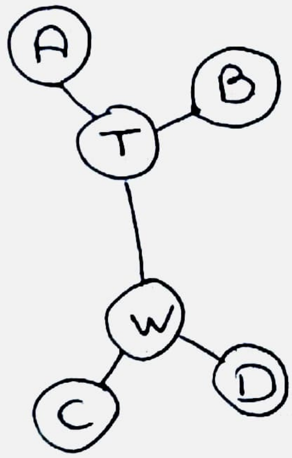
$$T_0=A$$
$$T_1=B$$
$$T_0+T_1=C$$





- If we had more variables

$$\underbrace{A+B}_{T=(A,B)} = \underbrace{C+D}_{W=(C,D)}$$

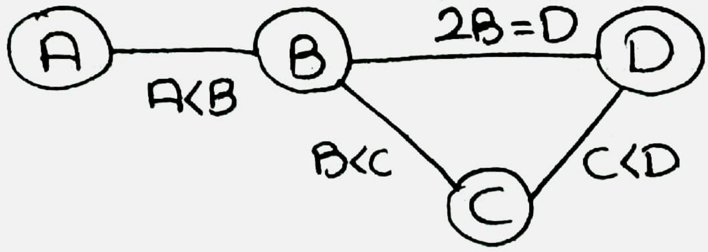


→ So we can, in effect reduce any n-ary constraint ($n>2$) into binary.

- How about Unary Constraints
  → Can be eliminated by reducing domains
  e.g.

$$x\in[1,10] \xrightarrow[C:\ x\geq 5]{} x\in[5,10]$$

- So we can always reduce Problem to only binary Constr.

- 

- A<B<C<D
- 2B=D
- Possible Values {1, 2, 3, 4, 5}

1. Consider Arcs into A

B ⟶ A
~~1~~, 2, 3, 4, 5   1, 2, 3, 4, 5

- need a value in A that's smaller

2. Consider Arcs into B

A ⟶ B
1, 2, 3, 4, ~~5~~   2, 3, 4, 5

- for each A, need larger B
→ No need to redo B→A

D ⟶ B
~~1~~, ~~2~~, ~~3~~, 4, ~~5~~   2, 3, 4, 5

- for each value in D, need half of it in B

C ⟶ B
~~1~~, ~~2~~, 3, 4, 5   2, 3, 4, 5

- for each value in C need a smaller value in B

- Consider arcs into C

B ⟶ C
2, 3, 4, ~~5~~   3, 4, 5

- for each value in B need a lager one in C   (redo B)

D ⟶ C
4   3, 4, 5

- for each value in D need a smaller one in C ✓

- Consider arcs into D

B ⟶ D
2, [illegible]   4

C ⟶ D
3, ~~4~~, ~~5~~   4

- For each B, need a larger D
(redo B)
- (redo C)

- Consider arcs into B

$\underset{1,2,\cancel{3},4}{A} \longrightarrow \underset{2,3}{B}$ • No need to redo A ($B \to A$)

$\underset{4}{D} \longrightarrow \underset{2,3}{B}$ ✓

$\underset{3}{C} \longrightarrow \underset{2,3}{B}$ ✓

- Consider arcs into C

$\underset{2,\cancel{3}}{B} \longrightarrow \underset{3}{C}$ redo but (Con execud $C \to B$)

$\underset{4}{D} \rightarrow \underset{3}{C}$ ✓

- Consider arcs into B

$\underset{1,\cancel{2}}{A} \longrightarrow \underset{2}{B}$ • No need to redo A

$\underset{4}{D} \longrightarrow \underset{2}{B}$ ✓

$\underset{3}{C} \longrightarrow \underset{2}{B}$ ✓

- Done (Nothing more to redo)

$\rightarrow$ Reduction in Domains

A	B	C	D	
{1}	{2}	{3}	{4}	} only one value in domain