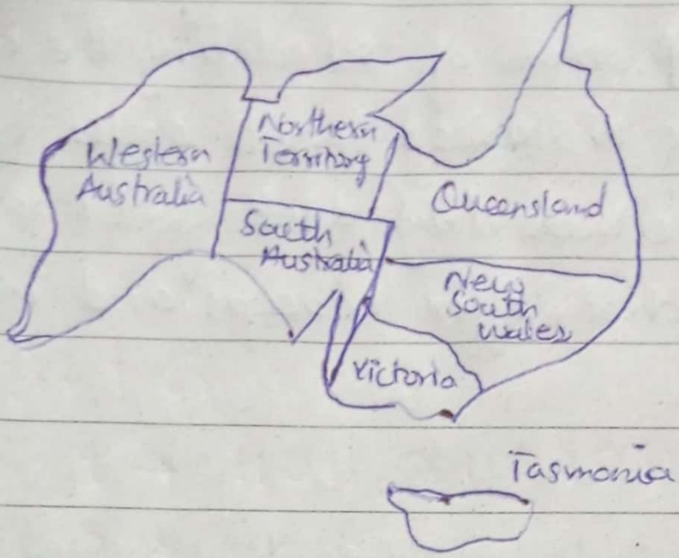
Date ______ Day ______

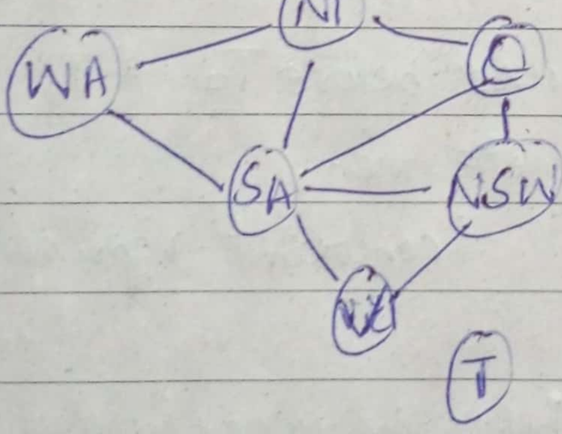
"CONSTRAINT SATISFACTION PROBLEM"



→ Graph Coloring Problem

→ we need to reduce this to CSP

graph



Variables : WA, NT, Q, NSW, V, SA, T

Domains : D_i = {Red, Blue, Green}

Constraints : Adjacent regions must have diff colors

eg: WA ≠ NT or

saare correct pairs likh diye (WA, NT) k liye

(WA, NT) ∈ {(red, green), (red, blue), (~~blue~~ green, red), (blue, red), (green, blue), (blue, green)}

Date Day

→ Job Scheduling Problem

Q: Consider the Problem of scheduling the assembly of car. There are 15 tasks: install axles (front & back), affix all four wheels (right & left, front & back), tighten nuts for each wheel, affix hubcaps and inspect final assembly.

· Variables = { $Axle_F$, $Axle_B$, $Wheel_{RF}$, $Wheel_{RB}$, $Wheel_{LF}$, $Wheel_{LB}$, $Nuts_{RF}$, $Nuts_{RB}$, $Nuts_{LF}$, $Nuts_{LB}$, Cap_{RF}, Cap_{RB}, Cap_{LF}, Cap_{LB}, Inspect }

~~→ Now we repr~~

• Whenever a task T_1 must ~~occurs~~ before T_2 & T_1 takes time duration d_1 to complete then,

$T_1 + d_1 \leq T_2$ (constraint representation)

i) Axles have to be place before the wheels are put on & it takes 10 min to install an axel, then

$Axle_F + 10 \leq Wheel_{RF}$; $Axle_B + 10 \leq Wheel_{RB}$

$Axle_F + 10 \leq Wheel_{LF}$; $Axle_B + 10 \leq Wheel_{LB}$

↳ presedence constraint

Date Day.

ii) For each wheel, we must affix the wheel (1min), then tighten the nuts (2min) & finally attach the hubcab (1 min)

$Wheel_{RF} + 1 \leq Nuts_{RF}$; $Nuts_{RF} + 2 \leq Cap_{RF}$

$Wheel_{RB} + 1 \leq Nuts_{RB}$; $Nuts_{RB} + 2 \leq Cap_{RB}$

$\vdots \qquad \vdots \qquad \vdots$

iii) We have 4 workers to install wheels but they have to share one tool that helps put the axle.

$(Axle_F + 10 \leq Axle_B)$ or $(Axle_B + 10 \leq Axle_F)$

↳ disjunctive constraint

iv) Inspection comes last & takes 3 min.

$X + d_x \leq Inspect$

↳ all tasks ↳ and their durations

v) Whole assembly to be done in 30 min

Domain = { 0, 1, 2 ... ~~30~~ 27 } (30 is liye nhi diya q k 3 min to inspection mein lagne hi hai)

↳ for tasks other than inspection

Page No.

Date   Day

- Variations in Variables

→ Variables having.

→ Discrete, finite domains ⇒ Map coloring, Scheduling with time limits, 8-Queens problem where domain for each queen is {1 ... 8}

→ Discrete, infinite domains ⇒ set of integers or strings No time limits in job scheduling problem.

→ Continuous domains ⇒ Scheduling of experiments on the Hubble Telescope requires very precise timings of observation.

- Variations in Constraints

→ Unary constraint ⇒ Single variable is involved in mentioning the constraint like (South Wales ≠ green)

→ Binary constraint ⇒ Two variables are involved (WA ≠ NT)

→ Higher Order Constraints ⇒ Constraint having ~~arbitrary~~ more than 2 no. of variables ⟨⟨X, Y, Z⟩, X < Y < Z or X > Y > Z⟩

→ Preference (soft) constraint ⇒ Constraint that is not necessary to satisfy. But it is good if satisfied

→ Global Constraint ⇒ Constraint having arbitary no. of variables
Alldiff (X, Y, Z), means X, Y, Z all must have diff values.
In crypparithmetic all ~~a~~ variable must have diff values.
~~Aeti~~ Alldiff (F, T, U, W, R, O)

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

→

A Binary CSP is one with only unary & binary constraints and can be repres-ed as a constraint graph

→ Constraint Propagation:
Using the constraints to reduce the no. of legal values for a variable.

ki domain values

• Node Consistency: Agr. har graph ki node par unary constraint satisfy hosha hai to it means that ~~node~~ variable is node-consistent.
eg: (SA ≠ green) so we now domain for SA is {red, blue}
A graph is node consistent if every variable in the graph is node consistent.

• Arc Consistent: A variable in a CSP is arc-consistent if every value in its domain satisfies binary constraint.

nd nodes

A = red A = blue A = green B = red B = blue B = green ... E = red

har node par single variable assignment krenge

Date ______ Day ______

→ Agr hum simple DFS (backtracking) se CSP ko solve krenge than it would ~~be~~ make $n! d^n$ leaf nodes, for 'n' no. of variables.

→ Isko thora kam krne we use single variable assignment. And this DFS for CSP with single variable assignment is called backtracking search

→ Nodes is liye kam krte bcz CSP is cumulative agr B ko phele red hojaye or phir A ko uske baad green Ya A ko phele green ho or B ko baad mein red. Us se farq nhi parta.

→ Backtracking search ko or improve krne k liye we can by checking

- Which variable should be assigned next? (variable order) → MRV, degree heuristic
- In what order domains values should be tried? (value order) → LCV
- can we detect failure early? → Forward checking, Arc consistency

Degree Heuristic → MCV (Most constraining variable)

Date ___________ Day ___________

- Minimum Remaining Values (MRV) (Variable Ordering)

→ Choose variable with fewest legal moves.

↳ Agr WA ko red assign krdia to now check k sabse kam domain values kis k paas bachi hai like

NT → blue, green , SA → blue, green (baaqi sab k paas saare option hai)

Ab agr tie hojaye to degree heuristic → Most constraining variable dekhenge.

↳ select variable involved in largest no. of constraints on other unassigned variables

↳ Choose one which has higher degree.

deg (NT) = ~~3~~2 , deg (SA) = ~~5~~4 → unassigned par deg nikalni hai

So assign SA → blue .

Ab NT k paas sabse kam option hai i.e 1

so assign NT → green

- Least Constraining Value : (Value Ordering)

↳ esi value choose krni hai jo baaqi unassinged variable ~~ko~~ k liye zyada se zyada options leave kre

↳ Agr WA ~~ko~~ → red & then NT → green

now ab humein agr Q ko color dena hai to hum red/blue de sakte hai. Agr red dia to SA k paas 1 options remaining agr blue dia 2 SA k paas 0 option remenjo. So we give Q → red.

Date ________ Day ________

- Forward Checking

↳ Whenever we assign a value we need to figure out now that which values are now not possible

↳ Keep track of remaining legal values for unassigned variables.

↳ Terminate search when any variable has no legal values.

	WA	NT	Q	NSW	V	SA	T
	R B G	~~R~~ B G	R B G	R B G	R B G	~~R~~ B G	R B G
1	R	~~R~~ B ~~G~~	R B G	R B G	R B G	~~R~~ B G	R B G
2	R	~~R~~ B ~~G~~	G	R B ~~G~~	R B G	~~R~~ B ~~G~~	R B G
3	R	~~R~~ B ~~G~~	G	R ~~B~~ ~~G~~	B	~~R~~ ~~B~~ ~~G~~	R B G

iske saare colors khatam so now we detect failure from this state.

Agr is step mein arc consistency check krte to failure detect hojata. bcz NT & SA both have blue only

- Arc Consistency

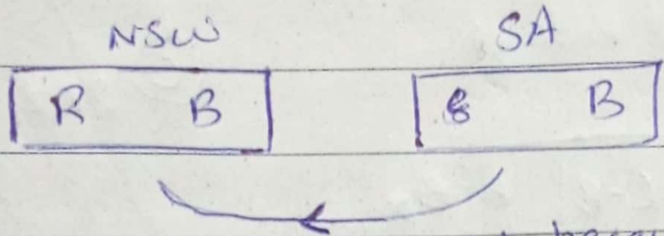
→ Agr hum constraints b/w two unassigned variables bhi check krlein har value assign krne k baad to or jldi failure detect hojayega.

If we check arc consistency of graph then we can detect failures easily.

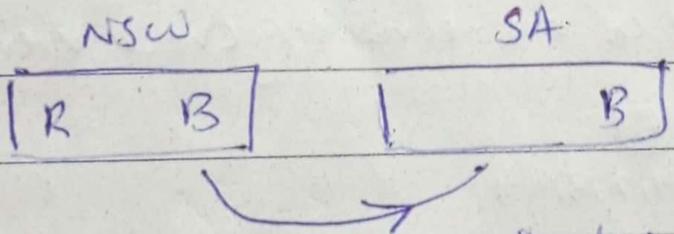
Date ________ Day ________

X → Y (an arc X To Y) is consistent iff
if for every value of x in X there is some
value of y in Y.

mean agr ek variable mein blue assign krdia to doosre
mein green, red hona zaroori hai.



it is arc consist because ~~SA mein jo blue hai~~
uske liye NSW mein red choos kr sakte hai



Not arc consistent because NSW mein Red
k liye to SA mein blue hai lekn NSW
mein jo blue hai uske liye SA mein koi color
nhi bcz since they are neighbors so same
value cant be assign.
Agr x k liye Y nhi hai to 'x' ko remove
krdenge.

e → each arc takes $O(d^2)$ to check [q k har element ko doosre element se compare krte rhe.]

• AC-3 Algorithm

$A \xrightarrow{A<B} B$ $\{\cancel{1},2,3,\cancel{4}\}$

A: $\{1,2,3,\cancel{4}\}$

$B \xrightarrow{B<C} C$ $\{\cancel{1},\cancel{2},3,4\}$

→ Sabse phele tamamne arcs ko Queue mein insert krdenge

Queue | ~~A-B~~ | ~~B-A~~ | ~~B-C~~ | ~~C-B~~ | ~~A-B~~ ← inserted again

(i) A-B ko pop krenge or arc-consistency check krenge

$A \xrightarrow{A<B} B$

A: $\{1,2,3,\cancel{4}\}$ B: $\{1,2,3,4\}$

4 k liye B mein ko value nhi

agr kisi variable ki domain shrink horhi to uska arc agr kisi doosre variable k saath ho or wo Queue mein nhi hai to Queue mein insert krdenge.

(2) B-A $B \xrightarrow{B>A} A$

B: $\{\cancel{1},2,3,4\}$ A: $\{1,2,3\}$

B shrink hoa or B ka C k saath arc hai lekn wo queue mein mojood hai.

(3) B-C $B \xrightarrow{B<C} C$

B: $\{2,3,\cancel{4}\}$ C: $\{1,2,3,4\}$

{B ki value shrink hoi or B ka A k saath constraint hai or A-B Queue mein nhi to wapis insert krdenge. B-A nhi insert krenge

→ Agr arc consistency se domain shrink hogyi hai to forward checking krne ki need nhi

Date ______ Day ______

(iv) C - B

$\underset{\{\cancel{1},\cancel{2},3,4\}}{C} \xrightarrow{C>B} \underset{\{2,3\}}{B}$

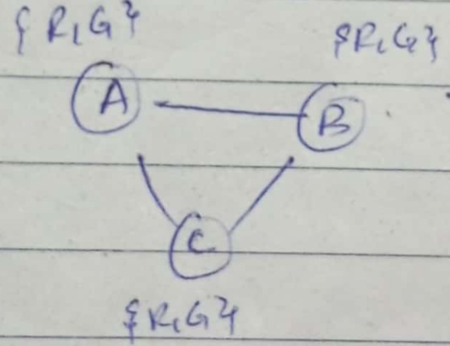
(v) A - B

$\underset{\{1,2,\cancel{3}\}}{A} \xrightarrow{A<B} \underset{\{2,3\}}{B}$

Now Queue is empty.

A → {1, 2}
B → {2, 3}
C → {3, 4}

- Path Consistency:



→ arc consistency will not detect failure here.

→ A & B ko esi values deni hai ko C bhi satisfy ho sake

(A - C) && (C - B)

- k consistency

if k=4 means 3 variables ko esi value deni hai k 4th wala bhi satisfy ho paye.