20L0325 BCS 6E

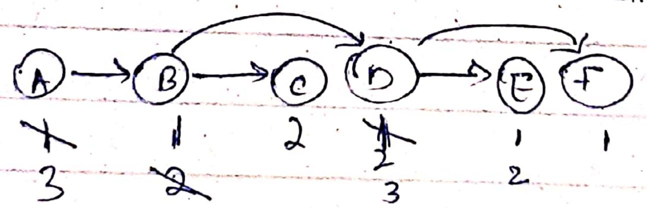
AI ASSIGNMENT 2

Q1:

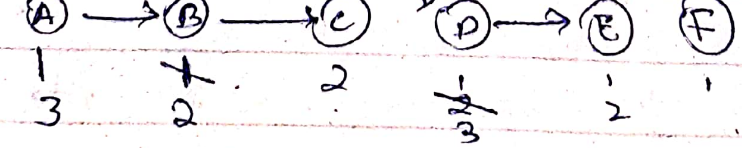
→ Checking arc ~~erstincy~~ consistency from the leaf nodes means that domain values are reduced from bottom to top so by the time the root node is reached, all assignments are guaranteed to be valid since all previous arcs have already been made consistent ~~dursi~~ before forward assignment take place.

→ However, starting the check from root means that a greater number of domains will need to be checked that might result in more backtracking since failure won't be detected until leaves are reached (No guarantee of valid assignment since failure not detected)

From leaves:



From root:



↳ Fails at C and can also fail at F (both leaf nodes)

Q2:

(1) C R O A C K
(3) H A C K
(2) E R R O R

C ~~0~~ 1 ~~2 3 4 5 6 7 8 9~~
R 0 ~~1 2~~ 3 4 5 6 7 8 9
A 0 ~~1 2~~ 3 4 5 6 7 8 9
O 0 ~~1 2~~ 3 4 5 6 7 8 9
K 0 ~~1 2~~ 3 4 5 6 7 8 9
H 0 ~~1 2~~ 3 4 5 6 7 8 9
R 0 ~~1 2~~ 3 4 5 6 7 8 9
E 0 ~~1~~ 2 3 4 5 6 7 8 9

MRV

→ Removing 0 from domain of C (as carry can't be zero)
→ Adding carry above C gives E = 2
C = 1

1 + H + R = R ; ~~H = 3, R = 0~~

→ R = 0, not possible as 1 + R + H ≠ R
→ R = 3
→ H = 0, not possible (1+3+0 ≠ 3)
→ H = 4, not possible (1+3+4 ≠ 3)
→ H = 5, not possible (1+3+5 ≠ 3)
⋮ H = 6 - H = 8 not possible
→ H = 9 ✓ (1+3+9 = 13) possible

C R A O K H E
1 3 9 2

→ A + A = 1R i.e. A + A = 13

→ A = 0 not possible (0 + 0 ≠ 13)
→ A = 4 not possible (4+4 ≠ 13)
→ A = 5 not possible (5+5 ≠ 13)

Scanned with CamScanner

→ A < 6 not possible ($~~A~~$ $6+6 \neq 13$)

⋮ A = 7 − 9 not possible

→ Add carry to constraint i.e. $1+A+A = 13$

→ A = 0. not possible ($1+0+0 \neq 13$)

→ ~~A~~ A = 1 − 5 not possible

→ A = 6 ✓ possible ($1+6+6 = 13$)

C	R	A	D	K	H	E
1	3	6			9	2

→ $C+C = 0$ i.e. $1+1 = 0$

D = 2 not possible (already assigned to E)

~~C R A D K H E~~

~~1 3~~ → Backtrack to $C+1 = E$ (first constraint)

C R H A

→ C = 2

→ $E = 2+1 = 3$ ⇒ C = 2, E = 3

→ R

→ R = 0, not possible ($1+0+H \neq 0$)

→ R = 1

→ H = 0 not possible ($1+1+0 \neq 1$)

→ H = 4 not possible ($1+1+4 \neq 1$)

⋮

→ H = 9 ✓ ($1+1+9 = 11$)

C E R H
2 3 1 9

→ A+A = R i.e. A+A = 11

→ A = 0 not possible (0+0 ≠ 11)

A = 1 not possible (1+1 ≠ 2)

⋮

~~A = 2~~ A = 4 − 8 not possible

→ Add a carry to constraint (1+A+A = 11)

~~A = 0 ✓ (1+0+0 = 1)~~

→ A = 0 not possible

→ A = 1 not possible

→ A = 2 not possible

→ A = 3 not possible

→ A = 4 not possible

→ A = 5 ✓ possible (1+5+5 = 11)

C E R H A
2 3 1 9 5

→ 2+2 = O

O = 1 ✗

⋮

O = 4 ✓ =>

C E R H A O
2 3 1 9 5 4

→ K+K = R i.e., K+K = 1

Not possible so backtrack to C

C ERHAO

→ $C = 3$

→ $E = 1+3 = 4$

C E
3 4

→ R = 0

→ A = 1

→ H = 0 not possible $(1+1+0 \neq 11)$

→ H = 2 not possible $(1+1+2 \neq 11)$

⋮ H = 5 - 8 not possible

→ H = 9 ✓ $(1+1+9 = 11)$

C E R H
3 4 1 9

→ A+A+1 = 1R i.e A+A+1 = 11

→ A = 0 not possible $(0+0+1 \neq 11)$

⋮

→ A = 5 possible $(1+5+5 = 11)$

C E R H A
3 4 1 9 5

→ C+C = O i.e., 3+3 = O

O = ~~0~~ ✗

⋮

O = 6 ✓

~~C E R H A O~~
~~3 4 1 9 5 6~~

→ K+K = R i.e, K+K = 1
Not possible. so backtrack to C

Scanned with CamScanner

$C = 4$

$E = 1+4 = 5$

C E
4 5

$\rightarrow R = 0$ ✗

$\rightarrow R = 1$ ✗ (Not possible since $K+K \neq 1$)

~~$\rightarrow H = 0$ not possible ($1+1+0 \neq$~~

$\rightarrow R = 2$.

$\longrightarrow H = 0$ not possible since $(1+2+0 \neq 12)$

$\longrightarrow H = 1$ not possible since $(1+2+1 \neq 12)$

$\longrightarrow H = 3$ " " $(1+2+3 \neq 12)$

$\rightarrow$ ⋮

$\longrightarrow H = 6-8$ " "

$\longrightarrow H = 9$ ✓ $(1+9+2 = 12)$

C E R H
4 5 2 9

$\longrightarrow A+A = 1R$ i.e. $A+A = 12$

$A = 0$ ✗

⋮

$A = 6$ ✓ $(6+6 = 12)$

C E R H A
4 5 2 9 6

$\longrightarrow \underline{C+C = O}$ i.e. $\underline{4+4 = O}$:

$A = 0$ ✗ ... $A = 8$ ✓ $(4+4 = 8)$

C E R H A O
4 5 2 9 6 8

$\rightarrow \underline{K+K = R}$ i.e. $\underline{K+K = 2}$:

$\longrightarrow K = 0$ ✗ $\longrightarrow K = 1$ ✓ $(1+1 = 2)$

$\boxed{\begin{matrix} C & E & R & H & A & O & K \\ 4 & 5 & 2 & 9 & 6 & 8 & 1 \end{matrix}}$

Scanned with CamScanner

Q3:

- One possible representation involves creating 25 variables.

  Each variable would be an attribute of a house number e.g. "House-1-Color" would represent The color of House 1.
  There are 5 houses, each having five different attributes i.e. Color, Nationality, Candy, Drink and Pet. Thus $5 \times 5 = 25$ variables

Domains

HouseXColor : { red, yellow, green, ivory, blue }
HouseXNationality : { Englishman, Spaniard, Norwegian, Japanese, Ukranian }
HouseXCandy : { Smarties, Snickers, kitkat, MilkyWay, Hershey }
HouseXDrink : { Milk, Water, Orange Juice, tea, coffee }
HouseXPet : { Zebra, Dog, Fox, Horse, Snail }

where X represents house no. from 1 to 5

Scanned with CamScanner

Constraints (General)

- Binary and unary constraints will be used i.e.
  - House 3 Drink = "milk" (Unary)
  - House 3 Drink ≠ House 2 Drink (For all drinks, pets, candies, colors, national
  - House 1 Nationality = "Norwegian"
  - House X Color = "Ivory" → House (X+1) Color = "green" (X = 1 to 4)
  - House X Candy = "Hershey" → (House (X+1) = "Fox" or House (X-1) = "Fox") (X = 2 to 4)
  - ~~House X Color = "Yellow" →~~
  - House X Candy = "kitkat" → (House X color = "yellow" or (House (X-1) Pet = "Horse" or House (X+1) Pet = "Horse")) {X = 2 to 4}
  - House X Nationality = "Englishman" → ~~House~~ House X Color = "red"
  - House X Nationality = "spaniard" → House X Pet = "dog"
  - House X Nationality = "Norwegian" → House (X-1)/(X+1) Color = "Blue"
  - House X Candy = "Smarties" → House X Pet = "snail"
  - House X Candy = "Snickers" → House X Drink = "Orange Juice"

Scanned with CamScanner

- HouseX Nationality = "Ukraine" → House X Drink = "tea"
- HouseX Nationality = "Japanese" → House X Candy = "Milky Way"

- House X Color = "Green" → House X Drink = "Coffee"

↳ Large search space means more constraints and much worse time complexity
5^{25} possible assignments and a maximum of $O(25^2) = 625$ constraints

- An alternative representation involves 5 variables for each house.

The domains for each house is as follows:
House X = {Color, Nationality, Candy, Pet, Drink}
where X = 1 to 5

→ 5 variables, 5 domain size, so 5^5 assignments

→ 5 variables so a maximum of $5^2 = 25$ possible constraints

↳ Lesser search space (25 variables vs 5 variables) implies lesser constraints (25 vs 625) which results in better time complexity (5^5 vs 5^{25}) so we should use 5 variable representation

Scanned with CamScanner

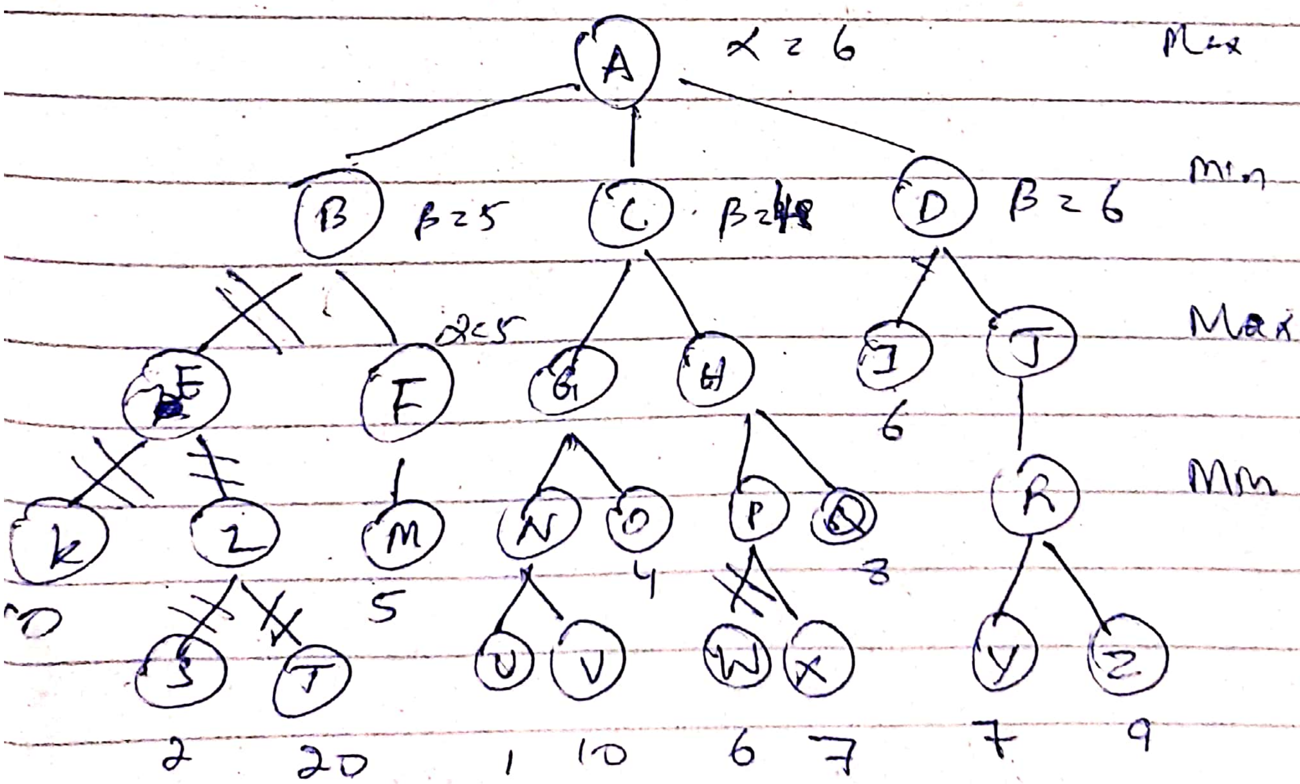
Q4:

a) 6

b) A → D

c) D → I

d) E, K, L, S, T, W



Q5.

(a) The papers explore resource allocation between licenced users and unlicensed (cognitive) users in radio networks, and how the latter dynamically utilizes available ~~pat~~ channels through proposed non-cooperative game theory techniques. The papers also discuss challenges faced when applying game theory to dynamically allocate resources in a network while protecting access of LUs and maximising QoS of CRs.

Scanned with CamScanner

Q5:

a)

b) Non-cooperative game theory is used where set of players are CRN nodes, their set of actions are coding rate, flow control, power level etc. Each node aims to maximise its utility function i.e. allocation of resources and nodes don't have access to each others' strategies. The utility function being maximised is SINR which denotes efficiency of the spectrum to reach Nash equilibrium state

c) Overlay allows CRs to use only unoccupied bands left by LUs and doesn't tolerate any interference caused by CRs to LU thus ~~doesn't~~ focuses more on QoS for LUs while Underlay allows LUs and CRs to co-exist and is tolerant of CR interference as it focuses on both LU protection and CR QoS at the same time

d) All utility functions aim to ~~increase~~ reach Nash equilibrium state. For this, three utility functions are proposed i.e.

1) Quality based utility in which ~~all~~ the throughput of all links is added together i.e. Quality based.

Scanned with CamScanner

2) Minimum interference utility function in which nodes choose the channel which causes minimum interference to other nodes

3) Combined utility function where both the quality (power) and interference of throughput links are weighted, and the weights determine whether high throughput, low interference, or a tradeoff is preferred.

Q6:

a) The first paper proposes alpha-beta pruning algorithm for games with more than two players to improve performance in terms of depth of search and time complexity. (space & complexity)

The second paper discusses how alpha beta pruning algorithms can be parallelized across multiple computers ~~to further improve~~ and investigates its efficiency for multiplayer games

b) OpenMP allocates each processor or thread a certain part of the game tree to solve in parallel, so multiple threads simultaneously solve ~~the~~ parts of the game tree. Sequential approach tries to solve complete tree in single thread which is slower

Scanned with CamScanner

c) CUDA based approach assigns tree traversal to the GPU which is then able to ~~perform~~ ~~traversal~~ solve the game tree in parallel by using its cores to solve a different subtree at the same time;

Sequential approach uses CPU single thread to perform the search and is therefore much slower.

Scanned with CamScanner