School of Data and Computer Science, Sun Yat-sen University

T02 CSP and KR

17341137 Zhenpeng Song

October 16, 2019

Contents

1 Q1

1.1 (a)

- Variables:

  V11, V12, ..., V21, V22, ..., V33.

- Domains:

  - Dom[V1i, Vi1] = {First letter of a certain word.}
  - Dom[V2i, Vi2] = {Second letter of a certain word.}
  - Dom[V3i, Vi3] = {Third letter of a certain word.}

- Constraints:

  - Make-A-Word(V1i,V2i,V3i)
  - Make-A-Word(Vi1,Vi2,Vi3)

1.2 (b)

- Variables:

  V1, V2, ..., Vk

- Domains:

  Dom[Vi] = {One vertex of the graph.}

- Constraints:

  Not-Adjacent(Vi, Vj)(i!=j)

1.3 (c)

- Variables:

  V1, V2, ..., V8(For each different letters)

- Domains:

  Dom[Vi] = {1-9}

- Constraints:

  Vi != Vj, i!=j

2 Q2

2.1 (a)

Please refer to the appendices.

2.2 (b)

Please refer to the appendices.

3 Q3

To resolve $(\exists x \forall y P(x,y) \vee \exists x \forall y Q(x,y)) \mid = \ \exists x \forall y (P(x,y) \vee Q(x,y))$
Do conversion and form clauses:

1. $\exists x \forall y P(x,y) \vee \exists r \forall z Q(r,z)$
2. $P(a,y) \vee Q(b,z)$ (1)
3. $\neg(\exists x \forall y (P(x,y) \vee Q(x,y))$
4. $\forall x \exists y (\neg P(x,y) \wedge \neg Q(x,y))$
5. $\forall x \left(\neg P\left(x, g(x)\right) \wedge \neg Q(x, g(x))\right)$
6. $\neg P(x, g(x))$ (2)
7. $\neg Q(x, g(x))$ (3)

$R[1,2](x = a, y = g(a)) \ = \ Q(b,z)$ (4)
$R[3,4](x = b, z = g(b)) \ = \ ()$ (*Done.*)

4 Q4

4.1 (a)

Using prolog:

- h(X,Y) :- man(X), man(Y), dis(X,Y), \ +X=Y.
- nh(X,Y) :- man(X), man(Y), \ +dis(X,Y), \ +X=Y.
- p(X) :- man(X), at(X).
- a(X) :- man(X), \ +at(X).
- m(X,Y) :- h(X,Y), p(X), \ +a(X), \ +nh(X,Y).
- man('a').
- man('b').
- man('c').
- man('v').

IF A is NOT True: dis('b','v'). at('a'). at('b').
IF B is NOT True: dis('c','v'). at('b'). dis('b','v'). at('a').
IF C is NOT True: dis(c,v).

```
1 ?- m(X, 'v'),write(X).
b
X = b
```

Figure 1: As a result, ONLY when B is NOT True, we found murderer B.

4.2 (b)

Using prolog in (a),
IF A is INNOCENT: h('c','v'). at('b'). h('b','v'). MAYBE at('c').
IF B is INNOCENT: h('b','v'). MAYBE at('a'). h('a','v'). MAYBE at('c'). h('c','v').
IF C is INNOCENT: h('b','v'). at('a'). at('b'). MAYBE h('a','v').

As a result, for each suspect that is innocent, we can infer that there are possibilities that the other two suspects committed the murder.

5 Appendix

Event	Node	Unassigned Var	Curdom	Value	DWO	Appendix
A << 1	1	A	1 2 3 4	1	0	A assigned
A>D	2	D	1 2 3 4	lf 1 2 3 4 F	0	
	3	D	Empty	\	1	DWO OCCUR
Restore	4	D	1 2 3 4	\	0	Back from FCcheck
A << 2	5	A	1 2 3 4	2	0	A assigned
A>D	6	D	1 2 3 4	lf 2 3 4 F	0	
	7	D	1		0	
C!=A	8	C	1 2 3 4	lf 2 F	0	
	9	C	1 3 4		0	
B>=A	10	B	1 2 3 4	lf 1 F	0	
	11	B	2 3 4		0	
CurDom:	A:{1,2,3,4} B:{2,3,4}		C:{1,3,4} D:{1} E:{1,2,3,4}			MRV:D
D << 1	13	D	1	1	0	A,D assigned
D>E	14	E	1 2 3 4	lf 1 2 3 4 F	0	
	15	E	Empty	\	1	DWO OCCUR
Restore	16	E	1 2 3 4	\	0	Back from FCcheck
Restore	17	D	1 2 3 4	\	0	
	18	C	1 2 3 4	\	0	
	19	B	1 2 3 4	\	0	Back from D
A << 3	20	A	1 2 3 4	3	0	A assigned
A>D	21	D	1 2 3 4	lf 3 4 F	0	
	22	D	1 2	\	0	
C!=A	23	C	1 2 3 4	lf 3 F	0	
	24	C	1 2 4	\	0	
B>=A	25	B	1 2 3 4	lf 1 2 F	0	
	26	B	3 4	\		
CurDom:	A:{1,2,3,4} B:{3,4}		C:{1,2,4} D:{1,2} E:{1,2,3,4}			MRV:B
B << 3	28	B	3 4	3	0	A,B assigned
B!=C	29	C	1 2 4	NO F	0	
CurDom:	A:{1,2,3,4} B:{3,4}		C:{1,2,4} D:{1,2} E:{1,2,3,4}			MRV:D
D << 1	31	D	1 2	1	0	A,B,D assigned
D>E	32	E	1 2 3 4	lf 1 2 3 4 F	0	
	33	E	Empty	\	1	DWO OCCUR
Restore	34	E	1 2 3 4	\	0	Back from FCcheck
D << 2	35	D	1 2	2	0	A,B,D assigned
D>E	36	E	1 2 3 4	lf 2 3 4 F	0	
	37	E	1	\	0	
C!=D	38	C	1 2 4	lf 2 F	0	
	39	C	1 4	\	0	
C!=D+1	40	C	1 4	NO F	0	
CurDom:	A:{1,2,3,4} B:{3,4}		C:{1,4} D:{1,2} E:{1}			MRV:E
E << 1	42	E	1	1	0	A,B,D,E assigned
C>E	43	C	1 4	lf 1 F	0	
	44	C	4	\	0	
Only C Left.						
C << 4	46	C	4	4	0	All assigned
As a result, the time slots look like:						
E	D	A / B	C			Time slots

Figure 2: (a)

Event	Node	Unassigned Var	Curdom	Value	DWO	Appendix
A << 1	1	A	1 2 3 4	1	0	
Prune	2	A	1	1	0	
Push into GAC Queue: A>D; C!=A; B>+A.						
GAC_Enforcing						
CurDom:	A:{1} B:{1,2,3,4}			C:{1,2,3,4} D:{1,2,3,4} E:{1,2,3,4}		
A>D: A	6	A	1	1	0	NOT FOUND
Remove	7	A	1	1	0	
	8	A	Empty	\	1	DWO OCCUR
Restore	9	A	1 2 3 4	\	0	Back from enforce
A << 2	10	A	1 2 3 4	2	0	
Prune	11	A	2	2	0	
Push into GAC Queue: A>D; C!=A; B>=A.						
GAC_Enforcing						
CurDom:	A:{2} B:{1,2,3,4}			C:{1,2,3,4} D:{1,2,3,4} E:{1,2,3,4}		
	15	A	2	2	0	Find D = 1
A>D: D	16	D	1 2 3 4	1		Find A = 2
	17	D	1 2 3 4	2 F	0	NOT FOUND
	18	D	1 3 4	3 F	0	NOT FOUND
	19	D	1 4	4 F	0	NOT FOUND
Remove	20	D	1	\	0	
C!=A:C	21	C	1 2 3 4	1	0	Find A = 2
	22	C	1 2 3 4	2 F	0	NOT FOUND
	23	C	1 3 4	3	0	Find A = 2
	24	C	1 3 4	4	0	Find A = 2
C!=A: A	25	A	2	2	0	Find C = 1/3/4
B>=A: B	26	B	1 2 3 4	1 F	0	NOT FOUND
	27	B	2 3 4	2	0	Find A = 2
	28	B	2 3 4	3	0	Find A = 2
	29	B	2 3 4	4	0	Find A = 2
B>=A: A	30	A	2	2	0	Find B = 2/3/4
GAC Queue is empty.						
CurDom:	A:{2} B:{2,3,4}			C:{1,3,4} D:{1} E:{1,2,3,4}		
D << 1	33	D	1	1	0	
Prune	34	D	1	1	0	
Push into GAC Queue: A>D; D>E; C!=D; C!=D+1.						
GAC_Enforcing						
CurDom:	A:{2} B:{2,3,4}			C:{1,3,4} D:{1} E:{1,2,3,4}		
A>D: A	38	A	2	2	0	Find D = 1
A>D: D	39	D	1	1	0	Find A = 2
D>E: D	40	D	1	1	0	NOT FOUND
Remove	41	D	1	1	0	
	42	D	Empty	\	1	DWO OCCUR
Restore	43	D	1	1	0	Back from enforce
Restore	44	A	1 2 3 4	\	0	Back from A
A << 3	45	A	1 2 3 4	3	0	
Prune	46	A	3	3	0	
Push into GAC Queue: A>D; C!=A; B>=A.						
GAC_Enforcing						

Figure 8: (b)(1)

GAC_Enforcing						
CurDom:	A:{3} B:{1,2,3,4}			C:{1,2,3,4} D:{1,2,3,4} E:{1,2,3,4}		
A>D: A	50	A	3	3	0	Find D = 1/2
A>D: D	51	D	1 2 3 4	1	0	Find A = 3
	52	D	1 2 3 4	2	0	Find A = 3
	53	D	1 2 3 4	3	0	NOT FOUND
	54	D	1 2 4	4	0	NOT FOUND
Remove	55	D	1 2	\	0	
Push D>E; C!=D; C!=D+1 into GAC Queue: C!=A; B>=A; D>E; C!=D; C!=D+1.						
C!=A: C	57	C	1 2 3 4	1	0	Find A = 3
	58	C	1 2 3 4	2	0	Find A = 3
	59	C	1 2 3 4	3	0	NOT FOUND
	60	C	1 2 4	4	0	Find A = 3
Push C>E; B!=C into GAC Queue: B>=A; D>E; C!=D; C!=D+1; C>E; B!=C.						
C!=A: A	62	A	3	3	0	Find C = 1/2/4
B>=A: B	63	B	1 2 3 4	1	0	NOT FOUND
	64	B	2 3 4	2	0	NOT FOUND
	65	B	3 4	3	0	Find A = 3
	66	B	3 4	4	0	Find A = 3
B>=A: A	67	A	3	3	0	Find B = 3/4
D>E: D	68	D	1 2	1	0	NOT FOUND
	69	D	2	2	0	Find E = 1
D>E: E	70	E	1 2 3 4	1	0	Find D = 2
	71	E	1 2 3 4	2	0	NOT FOUND
	72	E	1 3 4	3	0	NOT FOUND
	73	E	1 4	4	0	NOT FOUND
Remove	74	E	1	\	0	
C!=D: C	75	C	1 2 4	1	0	Find D = 2
	76	C	1 2 4	2	0	NOT FOUND
	77	C	1 4	4	0	Find D = 2
C!=D: D	78	D	2	2	0	Find C = 1/4
C!=D+1: C	79	C	1 4	1	0	Find D = 2
	80	C	1 4	4	0	Find D = 2
C!=D+1: D	81	D	2	2	0	Find C = 1/4
C>E: C	82	C	1 4	1	0	NOT FOUND
	83	C	4	4	0	Find E = 1
C>E: E	84	E	1	1	0	Find C = 4
B!=C: B	85	B	3 4	3	0	Find C = 4
	86	B	3 4	4	0	NOT FOUND
Remove	87	B	3	\	0	
B!=C: C	88	B	3	3	0	Find C = 4
GAC Queue is empty.						
CurDom:	A:{3} B:{3}			C:{4} D:{2} E:{1}		
B << 3	91	B	3	3	0	
C << 4	92	C	4	4	0	
D << 2	93	D	2	2	0	
E << 1	94	E	1	1	0	
As a result, the time slots look like:						
E		D	A B	C		Time slots

Figure 4: (b)(2)