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

T02 CSP and KRR

16337110 匡乾, 16337111 赖若潘

2018 年 10 月 16 日

目录

1 Q1

(a) Magic square

variables:

$V1, V2, V3, V4, V5, V6, V7, V8, V9$

图 1: 变量在图上的分布

domain:

$Dom[Vi] = \{1, 2, 3, \cdots, 9\}$ $(i \in \{1, 2, 3, \cdots, 9\})$

constraints:

我们很容易可以证明，当每一行，每一列，每一对角线之和相等时，当且仅当它们的和为15。

证明如下：$(1 + 2 + 3 + \cdots + 9)/3 = 15$

1. C(Vi,Vj):$Vi \neq Vj$ $(i, j \in \{1, 2, 3, \cdots, 9\}, i \neq j)$
2. C(V1,V4,V7):$V1 + V4 + V7 = 15$
3. C(V2,V5,V8):$V2 + V5 + V8 = 15$
4. C(V3,V6,V9):$V3 + V6 + V9 = 15$
5. C(V1,V2,V3):$V1 + V2 + V3 = 15$
6. C(V4,V5,V6):$V4 + V5 + V6 = 15$
7. C(V7,V8,V9):$V7 + V8 + V9 = 15$
8. C(V1,V5,V9):$V1 + V5 + V9 = 15$
9. C(V3,V5,V7):$V3 + V5 + V7 = 15$

(b) Hamiltonian tour

variables:

$City_i (i \in \{1, 2, \cdots, n\})$　表示某个城市

$Road_{(i,j)} (i, j \in \{1, 2, \cdots, n\}, i \neq j)$　表示某两个城市之间的道路

$Route_i (i \in \{1, 2, \cdots, n\})$　表示访问的路径

domain:

$City_i = \{0, 1\}$　0表示该城市未访问，1表示该城市已访问

$Road_{(i,j)} = \{0, 1\}$　城市i,城市j之间，无道路用0表示，有道路用1表示

$Route_i = \{1, 2, \cdots, n\}$　通过访问城市的顺序来表示访问的路径

constraints:

$C(City_1, City_2, \cdots, City_n) : \sum_{i=1}^{n} City_i = n$　保证所有城市都已经访问

$C(Route_i) : Route_i \neq Route_j \ (j \in \{1, \cdots, i\})$　保证当前访问的城市之前都没有访问过

$C(Route_i, Route_{i+1}) : Road_{(Route_i, Route_{i+1})} = 1 \ (i \in \{1, \cdots, n-1\})$　保证路径上相邻的两个城市有道路

$C(Road_{(i,j)}, Road_{(j,i)}) : Road_{(i,j)} = Road_{(j,i)}$　保证道路是双向的

(c) Crypto-arithmetic puzzle

varibles:

I, N, T, L, A

domain:

$Dom[I, L, A] = \{1, 2, 3, \cdots, 9\}$　首位数字不能为零

$Dom[N, T] = \{0, 1, 2, 3, \cdots, 9\}$

constraints:

1. $C(I, N, T, L, A) : (I * 100 + N * 10 + T) * L = (A * 1110 + I)$
2. $C(I, N) : I \neq N$
3. $C(I, T) : I \neq T$
4. $C(I, L) : I \neq L$
5. $C(I, A) : I \neq A$
6. $C(N, T) : N \neq T$
7. $C(N, L) : N \neq L$
8. $C(N, A) : N \neq A$
9. $C(T, L) : T \neq L$
10. $C(T, A) : T \neq A$
11. $C(L, A) : L \neq A$

2 Q2

(a) 见图(2)

(b)
- 首先进行一次GAC算法，求出变量的值域。
  - 对于A=1，B中没有值可以满足约束$A \geq B$，A中删去1。
  - 对于A=2，B中没有值可以满足约束$A \geq B$，A中删去2。
  - 对于A=3, 4，B中有B=3可以满足约束$A \geq B$。
  - 对于B=3, 4，A中有A=4可以满足约束$A \geq B$，C中有C=2满足约束$B > C \ or \ C - B = 2$。
  - 对于B=5, 8, 9，A中没有值可以满足约束$A \geq B$，B中删去5, 8, 9。
  - 对于C=2, 3, 5, 6，B中有B=9满足$B > C \ or \ C - B = 2$，D中都有值满足约束$C \neq D$。
  - 对于C=7, 9，B中没有值满足$B > C \ or \ C - B = 2$，C中删去9。
  - 对于D中所有值，C中都有值满足约束$C \neq D$。

  所以进行GAC算法后的值域为：D_A={3, 4}，D_B={3, 4}，D_C={2, 3, 5, 6}，D_D={3, 5, 7, 9}。
- 接下来用GAC算法，找到第一个结果，见图(3)：

3 Q3

$S_1 : \forall x \forall y (P(x,y) \supset P(y,x)) \iff \forall x \forall y (\neg P(x,y) \vee P(y,x)) \iff \neg P(x,y) \vee P(y,x)$

$S_2 : \forall x \forall y \forall z ((P(x,y) \wedge P(y,z)) \supset P(x,z)) \iff \forall x \forall y \forall z (\neg (P(x,y) \wedge P(y,z)) \vee P(x,z)) \iff \forall x \forall y \forall z (\neg P(x,y) \vee \neg P(y,z) \vee P(x,z)) \iff \neg P(x,y) \vee \neg P(y,z) \vee P(x,z)$

$S_3 : \forall x \exists y P(x,y) \iff \forall x P(x, f(x)) \iff P(x, f(x))$

$Enquiry : \forall x P(x,x)$

refutation:

1. $(\neg P(x,y), P(y,x))$
2. $(\neg P(u,v), \neg P(v,w), P(u,w))$
3. $P(z, f(z))$
4. $\neg P(a,a)$
5. R[1b,2b]{w=x,u=x,v=y} $(\neg P(x,y), P(x,x))$

6. R[4,5b]{x=a} $\neg P(a, y)$

7. R[3,6]{z=a,y=f(a)} ()

So, we prove that $S_1 \vee S_2 \vee S_3 \supset \forall x P(x, x)$

4 Q4

T(X) means that X is the murderer (it was thief so I use T to represent the action)

K(X) means that X knows who is the victim

S(X) means that X saw the victim that day

Alexander:

$\neg T(A) \rightarrow (\neg T(P) \wedge \neg T(M) \wedge K(P) \wedge K(M) \wedge K(A)) \iff T(A) \vee (\neg T(P) \wedge \neg T(M) \wedge K(P) \wedge K(M) \wedge K(A)) \iff$

- $T(A) \vee \neg T(P)$
- $T(A) \vee \neg T(M)$
- $T(A) \vee K(P)$
- $T(A) \vee K(M)$
- $T(A) \vee K(A)$

Peter:

$\neg T(P) \rightarrow (\neg S(P) \wedge \neg K(P)) \iff T(P) \vee (\neg S(P) \wedge \neg K(P)) \iff$

- $T(P) \vee \neg S(P)$
- $T(P) \vee \neg K(P)$

Michael:

$\neg T(M) \rightarrow (S(A) \wedge S(P) \wedge S(M) \wedge K(P) \wedge K(M) \wedge K(A)) \iff T(M) \vee (S(A) \wedge S(P) \wedge S(M) \wedge K(P) \wedge K(M) \wedge K(A)) \iff$

- $T(M) \vee S(A)$
- $T(M) \vee S(P)$
- $T(M) \vee S(M)$

- $T(M) \lor K(P)$
- $T(M) \lor K(M)$
- $T(M) \lor K(A)$

Query: $\exists x T(x)$

So now we get the KB as below:

1. $(T(A), \neg T(P))$
2. $(T(A), \neg T(M))$
3. $(T(A), K(P))$
4. $(T(A), K(M))$
5. $(T(A), K(A))$
6. $(T(P), \neg S(P))$
7. $(T(P), \neg K(P))$
8. $(T(M), S(A))$
9. $(T(M), S(P))$
10. $(T(M), S(M))$
11. $(T(M), K(P))$
12. $(T(M), K(M))$
13. $(T(M, K(A))$
14. $(\neg T(x), ans(x))$

(a) refutation:

15. $R[1b, 6a]$ $(T(A), \neg S(P))$
16. $R[9b, 15b]$ $(T(A), T(M))$
17. $R[2b, 16b]$ $T(A)$
18. $R[14a, 17]\{x = A\}$ $ans(A)$

Therefore, we can know the murderer is A.

(b) refutation:

19. $R[7b, 11b]$ $(T(P), T(M))$

20. $R[3b, 7b]$ $(T(A), T(P))$

21. $R[1b, 19a]$ $(T(A), T(M))$

By 19, 20 and 21, we can know that there are at least two murderers.

Now, we will find who are the murderers as below:

Query: $\exists x \exists y (T(x) \wedge T(y))$

22. $(\neg T(x), \neg T(y), ans(x, y))$

23. $R[19ab, 22ab]\{x = P, y = M\}$ $ans(P, M)$

24. $R[20ab, 22ab]\{x = A, y = P\}$ $ans(A, P)$

25. $R[21ab, 22ab]\{x = A, y = M\}$ $ans(A, M)$

We get three answers that one suspect is innocent and the other two are guilty, and the answers are all possible and correct.

In conclusion, we can't assume that there was only a single thief.

2a.PNG

图 2: Q2(a): Forward Checking

9

2b.PNG

图 3: Q2(b): GAC algorithm