Hanwen Wang 260778557

1.

a)

Variables: {K1, K2, Q1, Q2}

Domain (same for all variables): {1, 2, 3, 4}

Assume the horizontal axis is i and vertical axis is j.

Constraints: Q1j = 3, Q2j = 4, K1j = 1, K2j = 2,

K1i ≠ K2i ± 2, K1i ≠ Q1i ± 1, K2i ≠ Q2i ± 1, K2i ≠ Q1i ± 2,

|Q1i-Q2i| ≠ |Q1j-Q2j|, Q1i ≠ Q2i, |Qi – Ki| ≠ |Qj - Kj|, Qi ≠ Ki.

图片包含 墙壁

描述已自动生成

b)

K1 = {~~1~~, ~~2~~, ~~3~~, ~~4~~}

K2 = {~~1~~, ~~2~~, ~~3~~, ~~4~~}

Q1 = {~~1~~, ~~2~~, ~~3~~, ~~4~~}

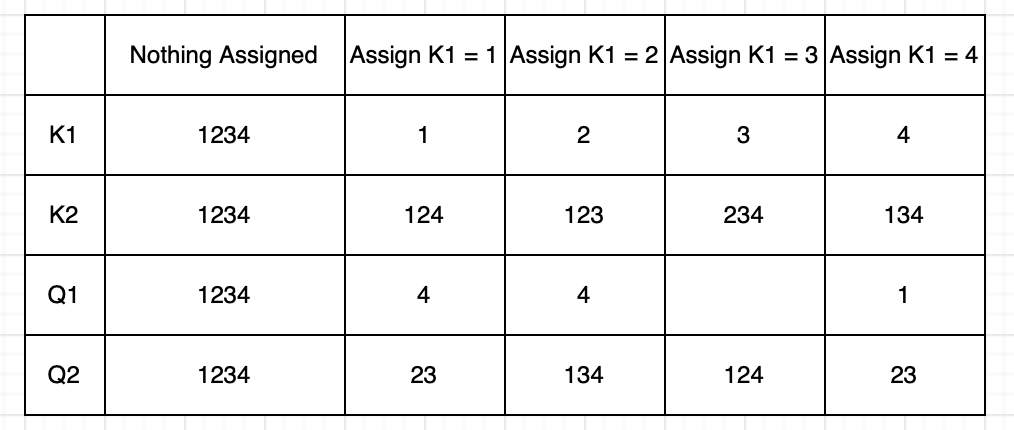
Q2 = {~~1~~, ~~2~~, ~~3~~, ~~4~~}

Queue: ~~K1i ≠ K2i ± 2~~, ~~K2i ± 2 ≠ K1i~~, ~~K1i ≠ Q1i ± 1~~, ~~Q1i ± 1 ≠ K1i~~, ~~K2i ≠ Q2i ± 1~~, ~~Q2i ± 1 ≠ K2i~~, ~~K2i ≠ Q1i ± 2~~, ~~Q1i ± 2 ≠ K2i~~, ~~|Q1i-Q2i| ≠ |Q1j-Q2j|~~, ~~|Q1j-Q2j| ≠ |Q1i-Q2i|~~, ~~Q1i ≠ Q2i~~, ~~Q2i ≠ Q1i~~, ~~|Qi – Ki| ≠ |Qj - Kj|~~, ~~|Qj - Kj| ≠ |Qi – Ki|~~, ~~Qi ≠ Ki~~, ~~Ki ≠ Qi~~, ~~Q1i ± 1 ≠ K1i~~, ~~Q2i ± 1 ≠ K2i~~, ~~Q1i ± 2 ≠ K2i~~, ~~Q2i ≠ Q1i~~.

For the backtracking search, since there is no value left for each variable, a tree cannot be built using this method, as a result, no solution can be found.

c)

One step forward checking:



Backtracking search:

图片包含 地图, 文字

描述已自动生成

2.

a)

图片包含 文字, 地图

描述已自动生成

b)

In this situation, I can play **rock** or **scissors** and my friend can play **paper** or **scissors**. If I play rock, my friend plays scissors, then no matter what the result is in the third round, I win this match. However, if he plays paper, then I lose this round and we will tie using scissors at the third round and finally we tie this match. On the contrary, if I play scissors at the second round, my friend plays paper, then I win this match. Else if he plays scissors, then we tied at the second round and he will win me at the third round because I can only play rock and he can only play paper. So we will tie this match. In conclusion, I cannot guarantee a win of this game now.

图片包含 文字

描述已自动生成

The above graph is a game tree starting from second round. The connection with arrow is the direction of flow after applying minimax algorithm. This terminal result supports the idea that I cannot guarantee a win overall.

3)

a)

i)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | A ∧ B | (A ∧ B) ∨ C |
| T | T | T | T | T |
| T | T | F | T | T |
| T | F | T | F | T |
| T | F | F | F | F |
| F | T | T | F | T |
| F | T | F | F | F |
| F | F | T | F | T |
| F | F | F | F | F |

5 models

ii)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | A => B | A ∧ (A => B) | A ∧ (A => B) ∧ ¬ B |
| T | T | T | T | F |
| T | F | F | F | F |
| F | T | T | F | F |
| F | F | T | F | F |

0 models

iii)

We assume the statement is ¬f1 ∨ f2. In order to ensure this logic to be true, f1 should be false and f2 should be true. So for f1, exclude all values are true, all other combinations will lead to a false. As a result, we only need to consider that exempt condition. When f1 is true, B ∧ C is true. Thus the final result is still true for this situation. In conclusion, there are models.

b)

i)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | f1: A=>B | f2: A ∨ C | f3: f1∧f2 | f4: B ∨ C | f3 => f4 |
| T | T | T | T | T | T | T | T |
| T | T | F | T | T | T | T | T |
| T | F | T | F | T | F | T | T |
| T | F | F | F | T | F | F | T |
| F | T | T | T | T | T | T | T |
| F | T | F | T | F | F | T | T |
| F | F | T | T | T | T | T | T |
| F | F | F | T | F | F | F | T |

This is valid.

ii)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | f1: A ∨ B | f2: f1=>C | f3: f2=>C | f4: B=>C | f5: f3∧f4 | f2=>f5 |
| T | T | T | T | T | T | T | T | T |
| T | T | F | T | F | F | F | F | T |
| T | F | T | T | T | T | T | T | T |
| T | F | F | T | F | F | T | F | T |
| F | T | T | T | T | T | T | T | T |
| F | T | F | T | F | T | F | F | T |
| F | F | T | F | T | T | T | T | T |
| F | F | F | F | T | T | T | T | T |

This is also valid.