Bilge Kaan Güneyli
2020400051

CMPE 480 HOMEWORK 2 REPORT

GENERAL DESCRIPTION

In the project, I used a *Node* object which hold parents to create the search graph. These objects also hold two dictionaries called *problem* and *assignment*. *Assignment* holds the variables and their assigned values, *problem* has 3 keys: *variables*, *domains* and *constraints*. These hold the relevant values, variables is a list and the other two are dictionaries.

Implementation of the project grounds on the lecture slides, the functions are named according to the slides too. Only the arc consistency algorithm is not taken directly because it was hard to adjust the algorithm to my case. Therefore, I created my version which only makes the propagation when only one variable is not assigned in a constraint. This relies on the fact that, only with one equation (since we are not combining constraints in CP) we cannot solve for multiple variables.

EXAMPLE RUNS

1.

Problem: 3 - P2 (3 colored Australian map)

Content of problem.txt:

variables:
['WA', 'NT', 'SA', 'Q', 'NSW', 'V', 'T']
domains:
{'WA': ['c1', 'c2', 'c3'], 'NT': ['c1', 'c2', 'c3'], 'SA': ['c1', 'c2', 'c3'], 'Q': ['c1', 'c2', 'c3'], 'NSW': ['c1', 'c2', 'c3'], 'V': ['c1', 'c2', 'c3'], 'T': ['c1', 'c2', 'c3']}
constraints:

```
'WA-NT': lambda WA, NT: WA != NT,

'WA-SA': lambda WA, SA: WA != SA,

'NT-SA': lambda NT, SA: NT != SA,

'NT-Q': lambda NT, Q: NT != Q,

'SA-Q': lambda SA, Q: SA != Q,

'SA-NSW': lambda SA, NSW: SA != NSW,

'SA-V': lambda SA, V: SA != V,

'Q-NSW': lambda Q, NSW: Q != NSW,

'NSW-V': lambda NSW, V: NSW != V,
```

Output: *(tried each of MRV, DH LCV CP one by one and also altogether, results were same)*
Number of expanded nodes: 8
Solution:
{'T': 'c1', 'V': 'c1', 'NSW': 'c2', 'SA': 'c3', 'Q': 'c1', 'NT': 'c2', 'WA': ‘c1'}

Bilge Kaan Güneyli
2020400051

2.
Problem: 8 - P1 (8-queens)

Content of problem.txt:
variables:
['Q1', 'Q2', 'Q3', 'Q4', 'Q5', 'Q6', 'Q7', 'Q8']
domains:
{'Q1': [1, 2, 3, 4, 5, 6, 7, 8], 'Q2': [1, 2, 3, 4, 5, 6, 7, 8], 'Q3': [1, 2, 3, 4, 5, 6, 7, 8], 'Q4': [1, 2, 3, 4, 5, 6, 7, 8], 'Q5': [1, 2, 3, 4, 5, 6, 7, 8], 'Q6': [1, 2, 3, 4, 5, 6, 7, 8], 'Q7': [1, 2, 3, 4, 5, 6, 7, 8], 'Q8': [1, 2, 3, 4, 5, 6, 7, 8]}
constraints:
'Q1-Q2': lambda Q1, Q2: Q1 != Q2 and abs(Q1 - Q2) != 1

'Q1-Q3': lambda Q1, Q3: Q1 != Q3 and abs(Q1 - Q3) != 2

'Q1-Q4': lambda Q1, Q4: Q1 != Q4 and abs(Q1 - Q4) != 3

'Q1-Q5': lambda Q1, Q5: Q1 != Q5 and abs(Q1 - Q5) != 4

'Q1-Q6': lambda Q1, Q6: Q1 != Q6 and abs(Q1 - Q6) != 5

'Q1-Q7': lambda Q1, Q7: Q1 != Q7 and abs(Q1 - Q7) != 6

'Q1-Q8': lambda Q1, Q8: Q1 != Q8 and abs(Q1 - Q8) != 7

'Q2-Q3': lambda Q2, Q3: Q2 != Q3 and abs(Q2 - Q3) != 1

'Q2-Q4': lambda Q2, Q4: Q2 != Q4 and abs(Q2 - Q4) != 2

'Q2-Q5': lambda Q2, Q5: Q2 != Q5 and abs(Q2 - Q5) != 3

'Q2-Q6': lambda Q2, Q6: Q2 != Q6 and abs(Q2 - Q6) != 4

'Q2-Q7': lambda Q2, Q7: Q2 != Q7 and abs(Q2 - Q7) != 5

'Q2-Q8': lambda Q2, Q8: Q2 != Q8 and abs(Q2 - Q8) != 6

'Q3-Q4': lambda Q3, Q4: Q3 != Q4 and abs(Q3 - Q4) != 1

'Q3-Q5': lambda Q3, Q5: Q3 != Q5 and abs(Q3 - Q5) != 2

'Q3-Q6': lambda Q3, Q6: Q3 != Q6 and abs(Q3 - Q6) != 3

'Q3-Q7': lambda Q3, Q7: Q3 != Q7 and abs(Q3 - Q7) != 4

'Q3-Q8': lambda Q3, Q8: Q3 != Q8 and abs(Q3 - Q8) != 5

'Q4-Q5': lambda Q4, Q5: Q4 != Q5 and abs(Q4 - Q5) != 1

Bilge Kaan Güneyli
2020400051

'Q4-Q6': lambda Q4, Q6: Q4 != Q6 and abs(Q4 - Q6) != 2

'Q4-Q7': lambda Q4, Q7: Q4 != Q7 and abs(Q4 - Q7) != 3

'Q4-Q8': lambda Q4, Q8: Q4 != Q8 and abs(Q4 - Q8) != 4

'Q5-Q6': lambda Q5, Q6: Q5 != Q6 and abs(Q5 - Q6) != 1

'Q5-Q7': lambda Q5, Q7: Q5 != Q7 and abs(Q5 - Q7) != 2

'Q5-Q8': lambda Q5, Q8: Q5 != Q8 and abs(Q5 - Q8) != 3

'Q6-Q7': lambda Q6, Q7: Q6 != Q7 and abs(Q6 - Q7) != 1

'Q6-Q8': lambda Q6, Q8: Q6 != Q8 and abs(Q6 - Q8) != 2

'Q7-Q8': lambda Q7, Q8: Q7 != Q8 and abs(Q7 - Q8) != 1

Output:
‘ ‘:
Number of expanded nodes: 114
Solution:
{'Q1': 1, 'Q2': 5, 'Q3': 8, 'Q4': 6, 'Q5': 3, 'Q6': 7, 'Q7': 2, 'Q8': 4}

MRV:
Number of expanded nodes: 76
Solution:
{'Q1': 1, 'Q2': 5, 'Q3': 8, 'Q4': 6, 'Q7': 2, 'Q5': 3, 'Q6': 7, 'Q8': 4}

LCV:
Number of expanded nodes: 129
Solution:
{'Q1': 1, 'Q2': 5, 'Q3': 8, 'Q4': 6, 'Q5': 3, 'Q6': 7, 'Q7': 2, 'Q8': 4}

MRV-CP:
Number of expanded nodes: 76
Solution:
{'Q1': 1, 'Q2': 5, 'Q3': 8, 'Q4': 6, 'Q7': 2, 'Q5': 3, 'Q6': 7, 'Q8': 4}

MRV-DH-LCV-CP:
Number of expanded nodes: 40
Solution:
{'Q1': 1, 'Q2': 6, 'Q3': 8, 'Q4': 3, 'Q5': 7, 'Q6': 4, 'Q7': 2, 'Q8': 5}

Bilge Kaan Güneyli
2020400051

3.
Problem: 0 - P3 (TO + TO = FOR)

Content of problem.txt:
variables:
['T', 'O', 'F', 'R']
domains:
{'T': [1, 2, 3, 4, 5, 6, 7, 8, 9], 'O': [0, 1, 2, 3, 4, 5, 6, 7, 8, 9], 'F': [0, 1], 'R': [0, 1, 2, 3, 4, 5, 6, 7, 8, 9], 'C1': [0, 1]}
constraints:

```
'T-O': lambda T, O: T != O,

'T-F': lambda T, F: T != F,

'T-R': lambda T, R: T != R,

'O-F': lambda O, F: O != F,

'O-R': lambda O, R: O != R,

'F-R': lambda F, R: F != R,

'T-O-F-R': lambda T, O, F, R: 2 * (10*T + O) == 100*F + 10*O + R,

'O-R*': lambda O, R: 2 * O == R or 2 * O == R + 10,

'T-F-O': lambda T, F, O: 2 * T == 10*F + O or 2 * T == 10*F + O - 1
```

Output:
‘ ‘:
Number of expanded nodes: 7
Solution:
{'F': 0, 'O': 2, 'R': 4, 'T': 1}

4.
Problem: 1 - P3 (TWO + TWO = FOWR)

Content of problem.txt:
variables:
['T', 'W', 'O', 'F', 'R']
domains:
{'T': [1, 2, 3, 4, 5, 6, 7, 8, 9], 'W': [0, 1, 2, 3, 4, 5, 6, 7, 8, 9], 'O': [0, 1, 2, 3, 4, 5, 6, 7, 8, 9], 'F': [0, 1], 'R': [0, 1, 2, 3, 4, 5, 6, 7, 8, 9], 'X1': [0, 1], 'X2': [0, 1]}
constraints:

```
'T-W': lambda T, W: T != W,

'T-O': lambda T, O: T != O,
```

```
            'T-F': lambda T, F: T != F,

            'T-R': lambda T, R: T != R,

            'W-O': lambda W, O: W != O,

            'W-F': lambda W, F: W != F,

            'W-R': lambda W, R: W != R,

            'O-F': lambda O, F: O != F,

            'O-R': lambda O, R: O != R,

            'F-R': lambda F, R: F != R,

              'T-W-O-F-R': lambda T, W, O, F, R: 2 * (100*T + 10*W + O) == 1000*F + 100*O + 10*W + R,

            'O-R*': lambda O, R: 2 * O == R or 2 * O == R + 10,

             'W': lambda W: 2 * W == W or 2 * W == W + 10 or 2 * W == W + 9 or 2 * W == W - 1,

            'T-O-F': lambda T, O, F: 2 * T == 10*F + O or 2 * T == 10*F + O - 1,
```

Output:
‘ ‘:
Number of expanded nodes: 21
Solution:
{'F': 0, 'O': 7, 'R': 4, 'T': 3, 'W': 9}

DH:
Number of expanded nodes: 21
Solution:
{'F': 0, 'O': 7, 'R': 4, 'T': 3, 'W': 9}

CP:
Number of expanded nodes: 21
Solution:
{'F': 0, 'O': 7, 'R': 4, 'T': 3, 'W': 9}

DH-LCV:
Number of expanded nodes: 19
Solution:
{'F': 0, 'O': 7, 'R': 4, 'T': 3, 'W': 9}

Bilge Kaan Güneyli
2020400051

MRV-DH-CP:
Number of expanded nodes: 16
Solution:
{'F': 0, 'W': 9, 'O': 7, 'R': 4, 'T': 3}

MRV-DH-LCV-CP:
Number of expanded nodes: 14
Solution:
{'F': 0, 'W': 9, 'O': 7, 'R': 4, 'T': 3}

5.
Problem: 3 - P3 (TYXWO + TYXWO = FOWXYR)

Content of problem.txt:
variables:
['T', 'Y', 'X', 'W', 'O', 'F', 'R']
domains:
{'T': [1, 2, 3, 4, 5, 6, 7, 8, 9], 'Y': [0, 1, 2, 3, 4, 5, 6, 7, 8, 9], 'X': [0, 1, 2, 3, 4, 5, 6, 7, 8, 9], 'W': [0, 1, 2, 3, 4, 5, 6, 7, 8, 9], 'O': [0, 1, 2, 3, 4, 5, 6, 7, 8, 9], 'F': [0, 1], 'R': [0, 1, 2, 3, 4, 5, 6, 7, 8, 9], 'C1': [0, 1], 'C2': [0, 1], 'C3': [0, 1], 'C4': [0, 1]}
constraints:
'T-Y': lambda T, Y: T != Y,

'T-X': lambda T, X: T != X,

'T-W': lambda T, W: T != W,

'T-O': lambda T, O: T != O,

'T-F': lambda T, F: T != F,

'T-R': lambda T, R: T != R,

'Y-X': lambda Y, X: Y != X,

'Y-W': lambda Y, W: Y != W,

'Y-O': lambda Y, O: Y != O,

'Y-F': lambda Y, F: Y != F,

'Y-R': lambda Y, R: Y != R,

'X-W': lambda X, W: X != W,

'X-O': lambda X, O: X != O,

'X-F': lambda X, F: X != F,

```
'X-R': lambda X, R: X != R,

'W-O': lambda W, O: W != O,

'W-F': lambda W, F: W != F,

'W-R': lambda W, R: W != R,

'O-F': lambda O, F: O != F,

'O-R': lambda O, R: O != R,

'F-R': lambda F, R: F != R,

    'T-Y-X-W-O-F-R': lambda T, Y, X, W, O, F, R: 2 * (10000*T + 1000*Y + 100*X + 10*W + O) == 100000*F + 10000*O + 1000*Y + 100*X + 10*W + R,

'O-R*': lambda O, R: 2 * O == R or 2 * O == R + 10,

'W-Y*': lambda W, Y: 2 * W == Y or 2 * W == Y + 10 or 2 * W == Y - 1 or 2 * W == Y + 9,

'X': lambda X: 2 * X == X or 2 * X == X + 10 or 2 * X == X - 1 or 2 * X == X + 9,

'Y-W*': lambda Y, W: 2 * Y == W or 2 * Y == W + 10 or 2 * Y == W - 1 or 2 * Y == W + 9,

'T-O-F': lambda T, O, F: 2 * T == 10*F + O or 2 * T == 10*F + O - 1,
```

Output:
‘ ‘:
Number of expanded nodes: 208
Solution:
None

MRV:
Number of expanded nodes: 144
Solution:
None

MRV-DH-LCV-CP:
Number of expanded nodes: 144
Solution:
None