Exam Candidate Number: Y3868453

Constraint Programming

# Section 1: Forward Checking

### Description of Code

In this python code, I created two class files using Object-Oriented Programming to program the CSP. The first file is called CSP.py which calls the instantiation function whilst FCSolver.py holds the necessary function used for forward checking to solve the n-queens problem. I used two main functions for the FCSolver.py one called ‘FCBranchLeft’ inputs a queen (variable) and value (domain) and checks if the equality can be pruned. The ‘FCBranchRight’ is used to check the opposite for when the queen and value are not equal. The ‘restoreValue’ in the CSP.py is then used to put the value back into the domain and then sort it so that each domain is pruned in order.

I used a list instantiated in the CSP.py by the ‘set\_constraint’ procedure to create the constraints between two different queens, the ‘revise’ method then works in FCSolver.py to check which domains still work referring to the list of constraints. On each branch attempt, I used the python library copy to take a screenshot of all the domains to redo each branch.

Output trace of 6 Queens

Left Branch now Q0=0

Revised domains:

Q0: [0, 1, 2, 3, 4, 5]

Q1: [2, 3, 4, 5]

Q2: [1, 3, 4, 5]

Q3: [1, 2, 4, 5]

Q4: [1, 2, 3, 5]

Q5: [1, 2, 3, 4]

Left Branch now Q1=2

Revised domains:

Q0: [0, 1, 2, 3, 4, 5]

Q1: [2, 3, 4, 5]

Q2: [1, 4, 5]

Q3: [1, 5]

Q4: [1, 3]

Q5: [1, 3, 4]

Left Branch now Q2=1

Revised domains:

Q0: [0, 1, 2, 3, 4, 5]

Q1: [2, 3, 4, 5]

Q2: [1, 4, 5]

Q3: [5]

Q4: [3]

Q5: [3, 4]

Left Branch now Q3=5

Revised domains:

Q0: [0, 1, 2, 3, 4, 5]

Q1: [2, 3, 4, 5]

Q2: [1, 4, 5]

Q3: [5]

Q4: [3]

Q5: [4]

Left Branch now Q4=3

Domain Wipeout

Right Branch now Q4!=3

Right Branch now Q3!=5

Right Branch now Q2!=1

Revised domains:

Q0: [0, 1, 2, 3, 4, 5]

Q1: [2, 3, 4, 5]

Q2: [4, 5]

Q3: [1]

Q4: [1, 3]

Q5: [1, 3, 4]

Left Branch now Q2=4

Revised domains:

Q0: [0, 1, 2, 3, 4, 5]

Q1: [2, 3, 4, 5]

Q2: [4, 5]

Q3: [1]

Q4: [1, 3]

Q5: [1, 3]

Left Branch now Q3=1

Domain Wipeout

Right Branch now Q3!=1

Right Branch now Q2!=4

Revised domains:

Q0: [0, 1, 2, 3, 4, 5]

Q1: [2, 3, 4, 5]

Q2: [5]

Q3: [1]

Q4: [1]

Q5: [1, 3, 4]

Left Branch now Q2=5

Revised domains:

Q0: [0, 1, 2, 3, 4, 5]

Q1: [2, 3, 4, 5]

Q2: [5]

Q3: [1]

Q4: [1]

Q5: [1, 3, 4]

Left Branch now Q3=1

Domain Wipeout

Right Branch now Q3!=1

Right Branch now Q2!=5

Right Branch now Q1!=2

Revised domains:

Q0: [0, 1, 2, 3, 4, 5]

Q1: [3, 4, 5]

Q2: [1, 3, 5]

Q3: [1, 2, 4, 5]

Q4: [1, 2, 3, 5]

Q5: [1, 2, 3, 4]

Left Branch now Q1=3

Revised domains:

Q0: [0, 1, 2, 3, 4, 5]

Q1: [3, 4, 5]

Q2: [1, 5]

Q3: [1, 2, 4]

Q4: [1, 2, 5]

Q5: [1, 2, 4]

Left Branch now Q2=1

Revised domains:

Q0: [0, 1, 2, 3, 4, 5]

Q1: [3, 4, 5]

Q2: [1, 5]

Q3: [2]

Q4: [2]

Q5: [2, 4]

Left Branch now Q3=2

Domain Wipeout

Right Branch now Q3!=2

Right Branch now Q2!=1

Revised domains:

Q0: [0, 1, 2, 3, 4, 5]

Q1: [3, 4, 5]

Q2: [5]

Q3: [1, 2]

Q4: [1, 2]

Q5: [1, 4]

Left Branch now Q2=5

Revised domains:

Q0: [0, 1, 2, 3, 4, 5]

Q1: [3, 4, 5]

Q2: [5]

Q3: [1, 2]

Q4: [1, 2]

Q5: [1, 4]

Left Branch now Q3=1

Domain Wipeout

Right Branch now Q3!=1

Domain Wipeout

Right Branch now Q2!=5

Right Branch now Q1!=3

Revised domains:

Q0: [0, 1, 2, 3, 4, 5]

Q1: [4, 5]

Q2: [1, 3]

Q3: [1, 2, 4, 5]

Q4: [1, 2, 3, 5]

Q5: [1, 2, 3, 4]

Left Branch now Q1=4

Revised domains:

Q0: [0, 1, 2, 3, 4, 5]

Q1: [4, 5]

Q2: [1, 3]

Q3: [1, 2, 5]

Q4: [1, 2, 3, 5]

Q5: [1, 2, 3]

Left Branch now Q2=1

Revised domains:

Q0: [0, 1, 2, 3, 4, 5]

Q1: [4, 5]

Q2: [1, 3]

Q3: [5]

Q4: [2]

Q5: [2, 3]

Left Branch now Q3=5

Revised domains:

Q0: [0, 1, 2, 3, 4, 5]

Q1: [4, 5]

Q2: [1, 3]

Q3: [5]

Q4: [2]

Q5: [2]

Left Branch now Q4=2

Domain Wipeout

Right Branch now Q4!=2

Right Branch now Q3!=5

Right Branch now Q2!=1

Revised domains:

Q0: [0, 1, 2, 3, 4, 5]

Q1: [4, 5]

Q2: [3]

Q3: [1, 5]

Q4: [2]

Q5: [1, 2]

Left Branch now Q2=3

Revised domains:

Q0: [0, 1, 2, 3, 4, 5]

Q1: [4, 5]

Q2: [3]

Q3: [1, 5]

Q4: [2]

Q5: [1, 2]

Left Branch now Q3=1

Revised domains:

Q0: [0, 1, 2, 3, 4, 5]

Q1: [4, 5]

Q2: [3]

Q3: [1, 5]

Q4: [2]

Q5: [2]

Left Branch now Q4=2

Domain Wipeout

Right Branch now Q4!=2

Right Branch now Q3!=1

Revised domains:

Q0: [0, 1, 2, 3, 4, 5]

Q1: [4, 5]

Q2: [3]

Q3: [5]

Q4: [2]

Q5: [1, 2]

Left Branch now Q3=5

Revised domains:

Q0: [0, 1, 2, 3, 4, 5]

Q1: [4, 5]

Q2: [3]

Q3: [5]

Q4: [2]

Q5: [1, 2]

Left Branch now Q4=2

Domain Wipeout

Right Branch now Q4!=2

Right Branch now Q3!=5

Right Branch now Q2!=3

Right Branch now Q1!=4

Revised domains:

Q0: [0, 1, 2, 3, 4, 5]

Q1: [5]

Q2: [1, 3]

Q3: [1, 2, 4]

Q4: [1, 3]

Q5: [2, 3, 4]

Left Branch now Q1=5

Revised domains:

Q0: [0, 1, 2, 3, 4, 5]

Q1: [5]

Q2: [1, 3]

Q3: [1, 2, 4]

Q4: [1, 3]

Q5: [2, 3, 4]

Left Branch now Q2=1

Domain Wipeout

Right Branch now Q2!=1

Domain Wipeout

Right Branch now Q1!=5

Right Branch now Q0!=0

Revised domains:

Q0: [1, 2, 3, 4, 5]

Q1: [0, 1, 2, 3, 4, 5]

Q2: [0, 1, 2, 3, 4, 5]

Q3: [0, 1, 2, 3, 4, 5]

Q4: [0, 1, 2, 3, 4, 5]

Q5: [0, 1, 2, 3, 4, 5]

Left Branch now Q0=1

Revised domains:

Q0: [1, 2, 3, 4, 5]

Q1: [0, 3, 4, 5]

Q2: [0, 2, 4, 5]

Q3: [0, 2, 3, 5]

Q4: [0, 2, 3, 4]

Q5: [0, 2, 3, 4, 5]

Left Branch now Q1=0

Revised domains:

Q0: [1, 2, 3, 4, 5]

Q1: [0, 3, 4, 5]

Q2: [2, 4, 5]

Q3: [2, 3, 5]

Q4: [2, 3, 4]

Q5: [2, 3, 4, 5]

Left Branch now Q2=2

Revised domains:

Q0: [1, 2, 3, 4, 5]

Q1: [0, 3, 4, 5]

Q2: [2, 4, 5]

Q3: [5]

Q4: [3]

Q5: [3, 4]

Left Branch now Q3=5

Revised domains:

Q0: [1, 2, 3, 4, 5]

Q1: [0, 3, 4, 5]

Q2: [2, 4, 5]

Q3: [5]

Q4: [3]

Q5: [4]

Left Branch now Q4=3

Domain Wipeout

Right Branch now Q4!=3

Right Branch now Q3!=5

Right Branch now Q2!=2

Revised domains:

Q0: [1, 2, 3, 4, 5]

Q1: [0, 3, 4, 5]

Q2: [4, 5]

Q3: [2, 3]

Q4: [2, 3, 4]

Q5: [2, 3, 4, 5]

Left Branch now Q2=4

Revised domains:

Q0: [1, 2, 3, 4, 5]

Q1: [0, 3, 4, 5]

Q2: [4, 5]

Q3: [2, 3]

Q4: [2, 3]

Q5: [2, 3, 5]

Left Branch now Q3=2

Domain Wipeout

Right Branch now Q3!=2

Domain Wipeout

Right Branch now Q2!=4

Revised domains:

Q0: [1, 2, 3, 4, 5]

Q1: [0, 3, 4, 5]

Q2: [5]

Q3: [2, 3]

Q4: [2, 4]

Q5: [3, 4]

Left Branch now Q2=5

Revised domains:

Q0: [1, 2, 3, 4, 5]

Q1: [0, 3, 4, 5]

Q2: [5]

Q3: [2, 3]

Q4: [2, 4]

Q5: [3, 4]

Left Branch now Q3=2

Revised domains:

Q0: [1, 2, 3, 4, 5]

Q1: [0, 3, 4, 5]

Q2: [5]

Q3: [2, 3]

Q4: [4]

Q5: [3, 4]

Left Branch now Q4=4

Domain Wipeout

Right Branch now Q4!=4

Right Branch now Q3!=2

Domain Wipeout

Right Branch now Q2!=5

Right Branch now Q1!=0

Revised domains:

Q0: [1, 2, 3, 4, 5]

Q1: [3, 4, 5]

Q2: [0, 2, 5]

Q3: [0, 2, 3, 5]

Q4: [0, 2, 3, 4]

Q5: [0, 2, 3, 4, 5]

Left Branch now Q1=3

Revised domains:

Q0: [1, 2, 3, 4, 5]

Q1: [3, 4, 5]

Q2: [0, 2, 5]

Q3: [0, 2]

Q4: [0, 2, 4]

Q5: [0, 2, 4, 5]

Left Branch now Q2=0

Revised domains:

Q0: [1, 2, 3, 4, 5]

Q1: [3, 4, 5]

Q2: [0, 2, 5]

Q3: [2]

Q4: [2, 4]

Q5: [2, 4, 5]

Left Branch now Q3=2

Revised domains:

Q0: [1, 2, 3, 4, 5]

Q1: [3, 4, 5]

Q2: [0, 2, 5]

Q3: [2]

Q4: [4]

Q5: [5]

Left Branch now Q4=4

Domain Wipeout

Right Branch now Q4!=4

Right Branch now Q3!=2

Right Branch now Q2!=0

Revised domains:

Q0: [1, 2, 3, 4, 5]

Q1: [3, 4, 5]

Q2: [2, 5]

Q3: [0, 2]

Q4: [0, 2, 4]

Q5: [0, 4]

Left Branch now Q2=2

Revised domains:

Q0: [1, 2, 3, 4, 5]

Q1: [3, 4, 5]

Q2: [2, 5]

Q3: [0]

Q4: [0]

Q5: [0, 4]

Left Branch now Q3=0

Domain Wipeout

Right Branch now Q3!=0

Right Branch now Q2!=2

Revised domains:

Q0: [1, 2, 3, 4, 5]

Q1: [3, 4, 5]

Q2: [5]

Q3: [0, 2]

Q4: [0, 2, 4]

Q5: [0, 4]

Left Branch now Q2=5

Revised domains:

Q0: [1, 2, 3, 4, 5]

Q1: [3, 4, 5]

Q2: [5]

Q3: [0, 2]

Q4: [0, 2, 4]

Q5: [0, 4]

Left Branch now Q3=0

Revised domains:

Q0: [1, 2, 3, 4, 5]

Q1: [3, 4, 5]

Q2: [5]

Q3: [0, 2]

Q4: [2, 4]

Q5: [4]

Left Branch now Q4=2

Revised domains:

Q0: [1, 2, 3, 4, 5]

Q1: [3, 4, 5]

Q2: [5]

Q3: [0, 2]

Q4: [2, 4]

Q5: [4]

Left Branch now Q5=4

Revised domains:

Q0: [1, 2, 3, 4, 5]

Q1: [3, 4, 5]

Q2: [5]

Q3: [0, 2]

Q4: [2, 4]

Q5: [4]

Our Solution Assignments : {0: 1, 1: 3, 2: 5, 3: 0, 4: 2, 5: 4}

# Section 2: Modelling the Task Allocation Problem

For this model, I used an occurrence relation where the columns represent the task, and the rows represent the person. The matrix allocation is used as a Boolean where the number 1 represents the person been allocated the task and 0 represent the person that has not been allocated the task. This is model is represented in the ‘TAP.eprime’ file.

**Evaluation of Instances**

Evaluation of Instance1.param

This Instance is given in the exam as the first example. From the Allocation table, we have received an allocation where all constraints are met with each task allocated to one person and for each person, the duration of the task allocated to them does not exceed the deadline. The objective function also met its best optimal solution at 0 which shows the same solution as the exam example.

Evaluation of Instance2.param

This is Appendix A Instance in the exam. From this Allocation table, found that all constraints are met with the total objective function coming to 36. All people have their allocated needs met for them, so this is the optimal solution.

Evaluation of Instance3.param

For this Instance, I tested what happens when all the 3 people choose the same Bids, this also came up to the optimal solution which was 4 as this is the square of two people without their preferred choice.

Evaluation of Instance4.param

For this Instance, I max out all the tasks to have the same amount of time to take as the deadline to see if the model will only allocate one task per individual. This worked with the optimal strategy for the bids.

Evaluation of Instance5.param

For this last instance, I made one of the tasks with a time of 51 where the deadline is 50. This created no solution file.

**Most Challenging Task**

The most challenging task you can create with 12 tasks and 6 people is that each person bids on 2 tasks however the task that they bid for add up to more than the deadline. This causes the model to allocate to people who have did not bid for that task.

**Different Viewpoint**

For a different viewpoint, the ‘TAPExplicit.eprime’ file represented the allocation matrix as a single row where the index is the task, and the element of that index is the person who has been allocated that task.