GIFT UNIVERSITY

Converting Knowledge into Practical Experience

Report

Name	Sufyan Ahmad
Roll No	201980013
Subject	Artificial Intelligence
Section	Cs 415(C)
Instructor	Muhammad Shakeel

CSP INTRODUCTION

As a set of objects whose state must satisfy a number of **constraints** or limitations. CSPs represent the entities in a **problem** as a homogeneous collection of finite **constraints** over variables, which is solved by **constraint satisfaction** methods

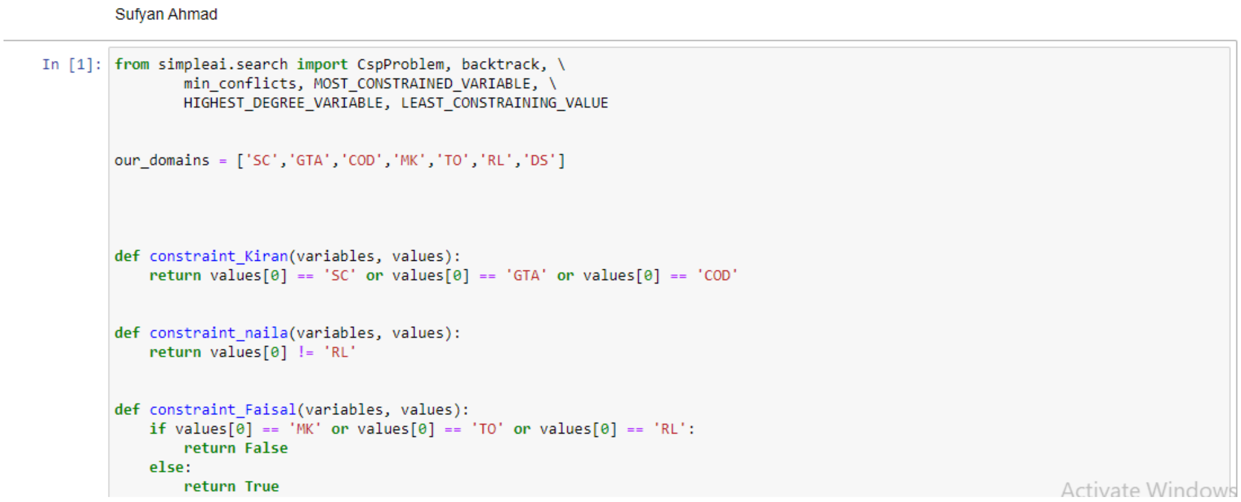
Minimum remaining values (MRV)

Choose the variable with the fewest possible values. Least-constraining value heuristic: choose a value that rules out the smallest number of values in variables connected to the current variable by constraints.

CSP Components

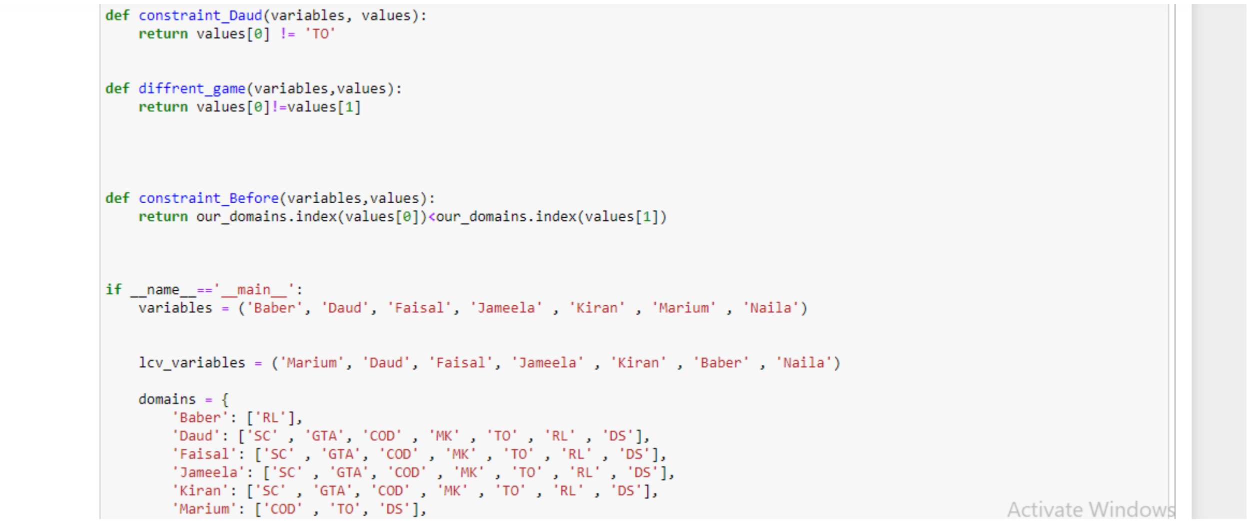
V is set of variables.
D is set of domain.
C is set of specify combination of values.

Algorithm Working:

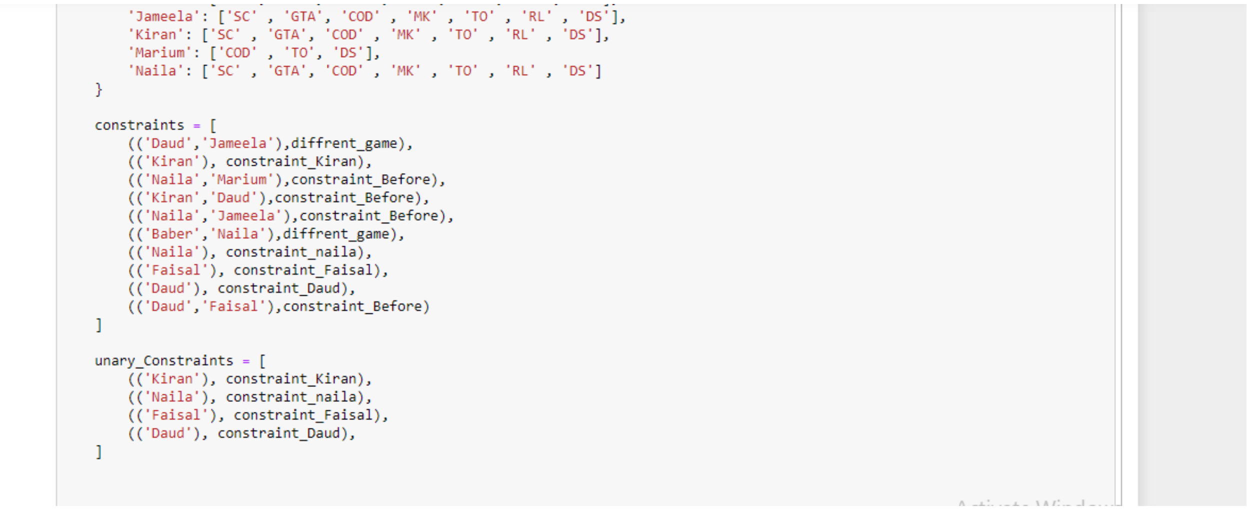


Sufyan Ahmad

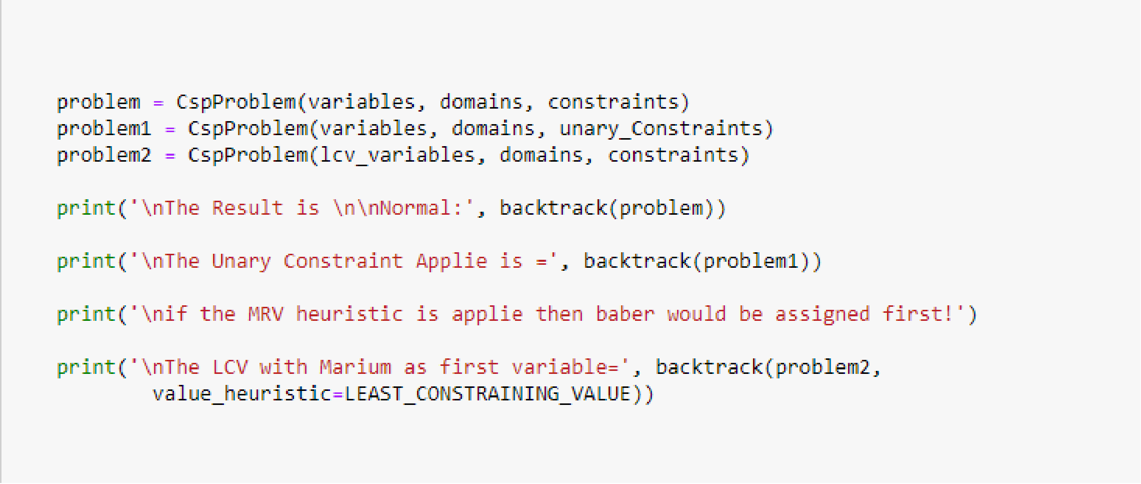
```
In [1]: from simpleai.search import CspProblem, backtrack, \
                min_conflicts, MOST_CONSTRAINED_VARIABLE, \
                HIGHEST_DEGREE_VARIABLE, LEAST_CONSTRAINING_VALUE


        our_domains = ['SC','GTA','COD','MK','TO','RL','DS']




        def constraint_Kiran(variables, values):
            return values[0] == 'SC' or values[0] == 'GTA' or values[0] == 'COD'


        def constraint_naila(variables, values):
            return values[0] != 'RL'


        def constraint_Faisal(variables, values):
            if values[0] == 'MK' or values[0] == 'TO' or values[0] == 'RL':
                return False
            else:
                return True
```



```
def constraint_Daud(variables, values):
    return values[0] != 'TO'


def diffrent_game(variables,values):
    return values[0]!=values[1]



def constraint_Before(variables,values):
    return our_domains.index(values[0])<our_domains.index(values[1])



if __name__=='__main__':
    variables = ('Baber', 'Daud', 'Faisal', 'Jameela' , 'Kiran' , 'Marium' , 'Naila')


    lcv_variables = ('Marium', 'Daud', 'Faisal', 'Jameela' , 'Kiran' , 'Baber' , 'Naila')

    domains = {
        'Baber': ['RL'],
        'Daud': ['SC' , 'GTA', 'COD' , 'MK' , 'TO' , 'RL' , 'DS'],
        'Faisal': ['SC' , 'GTA', 'COD' , 'MK' , 'TO' , 'RL' , 'DS'],
        'Jameela': ['SC' , 'GTA', 'COD' , 'MK' , 'TO' , 'RL' , 'DS'],
        'Kiran': ['SC' , 'GTA', 'COD' , 'MK' , 'TO' , 'RL' , 'DS'],
        'Marium': ['COD' , 'TO', 'DS'],
```



```
        'Jameela': ['SC' , 'GTA', 'COD' , 'MK' , 'TO' , 'RL' , 'DS'],
        'Kiran': ['SC' , 'GTA', 'COD' , 'MK' , 'TO' , 'RL' , 'DS'],
        'Marium': ['COD' , 'TO', 'DS'],
        'Naila': ['SC' , 'GTA', 'COD' , 'MK' , 'TO' , 'RL' , 'DS']
    }

    constraints = [
        (('Daud','Jameela'),diffrent_game),
        (('Kiran'), constraint_Kiran),
        (('Naila','Marium'),constraint_Before),
        (('Kiran','Daud'),constraint_Before),
        (('Naila','Jameela'),constraint_Before),
        (('Baber','Naila'),diffrent_game),
        (('Naila'), constraint_naila),
        (('Faisal'), constraint_Faisal),
        (('Daud'), constraint_Daud),
        (('Daud','Faisal'),constraint_Before)
    ]

    unary_Constraints = [
        (('Kiran'), constraint_Kiran),
        (('Naila'), constraint_naila),
        (('Faisal'), constraint_Faisal),
        (('Daud'), constraint_Daud),
    ]
```



```
    problem = CspProblem(variables, domains, constraints)
    problem1 = CspProblem(variables, domains, unary_Constraints)
    problem2 = CspProblem(lcv_variables, domains, constraints)

    print('\nThe Result is \n\nNormal:', backtrack(problem))

    print('\nThe Unary Constraint Applie is =', backtrack(problem1))

    print('\nif the MRV heuristic is applie then baber would be assigned first!')

    print('\nThe LCV with Marium as first variable=', backtrack(problem2,
            value_heuristic=LEAST_CONSTRAINING_VALUE))
```

.First of all import simpleai.search with CSPproblem and backtrack

.Then Baber , Daud , Faisal , Jameela , Kiran , Mariyam, Naila assign as a variables.

.Then pass domain in algorithm for example which SC, GTA ,CoD, MK , RL , DS is available for Baber .

.Then made method for constraint_Daud, diffrent_game, constraint_Before.

Then enter constraints in algorithm.

Q1

```
The Unary Constraint Applie is = {'Baber': 'RL', 'Daud': 'SC', 'Faisal': '
SC', 'Jameela': 'SC', 'Kiran': 'SC', 'Marium': 'COD', 'Naila': 'SC'}
```

Q2

```
if the MRV heuristic is applie then baber would be assigned first!
```

Q3

```
The LCV with Marium as first variable= {'Marium': 'COD', 'Daud': 'GTA', 'Fais
al': 'COD', 'Jameela': 'COD', 'Kiran': 'SC', 'Baber': 'RL', 'Naila': 'SC'}
```

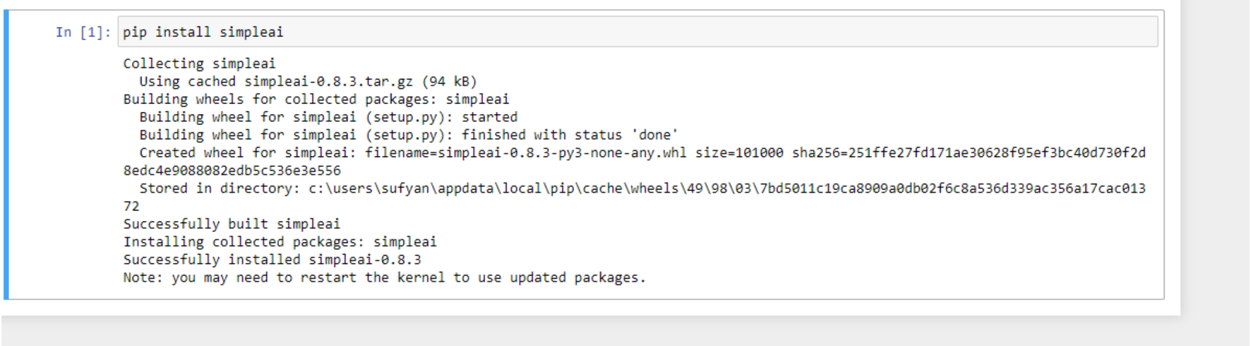
Q4

```
Kiran: SC, Daud: GTA, Faisal: CoD, Jameela: RL, Naila: MK, Baber: TO, Marium:
TO
```

Install Simpleai in Jupyter

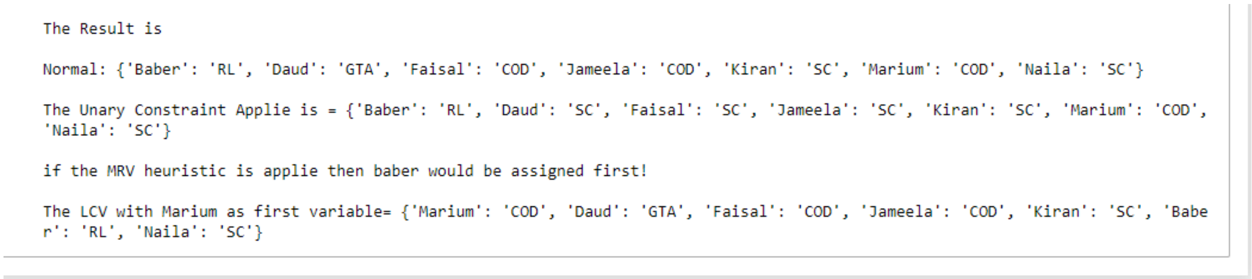
.First of all open jupyter notebook.

.Then install simpleai library with the help of enter command pip intallsimpleai then successfully install simpleai library.



```
In [1]: pip install simpleai

        Collecting simpleai
          Using cached simpleai-0.8.3.tar.gz (94 kB)
        Building wheels for collected packages: simpleai
          Building wheel for simpleai (setup.py): started
          Building wheel for simpleai (setup.py): finished with status 'done'
          Created wheel for simpleai: filename=simpleai-0.8.3-py3-none-any.whl size=101000 sha256=251ffe27fd171ae30628f95ef3bc40d730f2d
        8edc4e9088082edb5c536e3e556
          Stored in directory: c:\users\sufyan\appdata\local\pip\cache\wheels\49\98\03\7bd5011c19ca8909a0db02f6c8a536d339ac356a17cac013
        72
        Successfully built simpleai
        Installing collected packages: simpleai
        Successfully installed simpleai-0.8.3
        Note: you may need to restart the kernel to use updated packages.
```

Final Output is:



```
The Result is

Normal: {'Baber': 'RL', 'Daud': 'GTA', 'Faisal': 'COD', 'Jameela': 'COD', 'Kiran': 'SC', 'Marium': 'COD', 'Naila': 'SC'}

The Unary Constraint Applie is = {'Baber': 'RL', 'Daud': 'SC', 'Faisal': 'SC', 'Jameela': 'SC', 'Kiran': 'SC', 'Marium': 'COD',
'Naila': 'SC'}

if the MRV heuristic is applie then baber would be assigned first!

The LCV with Marium as first variable= {'Marium': 'COD', 'Daud': 'GTA', 'Faisal': 'COD', 'Jameela': 'COD', 'Kiran': 'SC', 'Babe
r': 'RL', 'Naila': 'SC'}
```