

Programming Assignment 3

Constraint Satisfaction Problems

How to run?

1. Make “csp_puzzle_main” executable using command:

```
chmod +x csp_puzzle_main
```

2. Now run by command:

```
./csp_puzzle_main
```

This program takes no arguments and prints solution for both puzzles by using two approaches: Backtracking and MRV.

Solution:

For **Jobs Puzzle**:

Steve : ['nurse', 'police officer']
Pete : ['clerk', 'actor']
Roberta : ['guard', 'teacher']
Thelma : ['chef', 'boxer']

For **house Puzzle**:

House 1 : {'color': 'yellow', 'country': 'Norwegian', 'drink': 'water', 'eat': 'kit kats', 'animal': 'fox'}
House 2 : {'color': 'blue', 'country': 'Ukrainian', 'drink': 'tea', 'eat': 'hershey bars', 'animal': 'horse'}
House 3 : {'color': 'red', 'country': 'English', 'drink': 'milk', 'eat': 'smarties', 'animal': 'snails'}
House 4 : {'color': 'ivory', 'country': 'Spaniard', 'drink': 'orange juice', 'eat': 'snickers', 'animal': 'dog'}
House 5 : {'color': 'green', 'country': 'Japanese', 'drink': 'coffee', 'eat': 'milky way', 'animal': 'zebra'}

Number of States searched to find solution:

	Backtracking(without MRV)	Minimum Remaining Values (MRV)
Jobs Puzzle	2976	486
Houses Puzzle	14065	3471

It is clear from above observation that total number of states searched to find solution reduces when MRV heuristic is used.

Explanation:

Jobs Puzzle:

Variables = Persons : All persons are taken as variable i.e. 4 variables. And hash-able is used to represent them as following:

```
{"Roberta": 0, "Thelma": 0, "Steve": 0, "Pete": 0}
```

A hash-table where 0 means value is not assigned.

Domains: For each variable a list of all combination of pair of jobs taken, as a person can take exactly two jobs

```
{"Roberta": [], "Thelma": [], "Steve": [], "Pete": []}
```

where [] is a list which stores all possible combinations (56) for particular variable.

Consistency check method: consistency_jobs()

It checks for whether given assignment is legal i.e. it satisfies with every given constraint and a job is not assigned to more than one person. For each constraint “if” condition is used, if given assignment contradicts with any of “if” condition, method returns false otherwise if all if conditions are passed method returns true.

Solution: In solution each person corresponds to list of two jobs. This is stored in assignment(hashtable) where person is “key” and list of two jobs is corresponding “value”.

Example of value: ['nurse', 'police officer']

Houses Puzzle:

Variables= House number corresponding to their positions as :[1, 2, 3, 4, 5]

Its a list where 1 is for House one from left and 2 is for House two and so on.

Domains: Similar to jobs problem for each variable a list of all combinations of 5 categories is taken, so total possible combination comes to be 3125. (5*5*5*5*5)

```
domain = {1:[],2:[],3:[],4:[],5:[]}
```

where [] is a list which store combinations from following 5 categories:

```
country = ["Norwegian", "Spaniard", "English", "Ukranian", "Japanese"]
eats = ["kit kats", "hershey bars", "smarties", "snickers", "milky way"]
drinks = ["water", "orange juice", "tea", "coffee", "milk"]
colors = ["yellow", "red", "green", "ivory", "blue"]
animals = [ "fox", "dog", "snails", "horse", "zebra"]
```

Consistency check method: consistency_houses()

Similar to jobs puzzle here also “if” condition is used to define each constraint and if given assignment fails any “if” condition then method returns false and if neither of “if” condition return false that means all given assignment satisfy all constraints and method returns True.

Solution: In solution Each House number corresponds to list of 5 values from 5 different categories. This is stored in assignment(Hashtable) where 1,2,3,4,5 corresponds to “keys” and list of 5 values from 5 categories is “value”

Example of value : {'color': 'red', 'country': 'English', 'drink': 'milk', 'eat': 'smarties', 'animal': 'snails'}

