Constraint Satisfaction Problem

Group member: Biao Wang

Programming language: Python
Hardware: macOS with apple M1 chip. 8G RAM

The program focuses on solving a Sudoku puzzle as a constraint satisfaction problem. It implements states and variables as objects derived from the State and Variable classes, respectively.

State Class:
- Board: A 2D list representing the Sudoku board, containing the numbers currently placed on the board.
- Assignment List: A list comprising all the assignments made after the initial state, documenting the progression of the puzzle solution.

Variable Class:
- Coordinate: Indicates the position of the variable within the Sudoku grid.
- Priority Value: Specifies the priority level assigned to the variable during the puzzle-solving process.
- Domain: Represents the set of permissible values that can be assigned to the variable at a given point in the solution process.

Constraint implementation:

The constraint and forward checking of this CPS problem is embedded within the process of establishing the variable domain. During the initialization of variables, their domains are restricted to exclude values that already exist within the corresponding row, column, or block.

The implementation of the Minimum Remaining Values (MRV) and degree heuristics, along with the resolution of tie-breaks, is carried out through the utilization of the "priority" attribute for each variable. This priority value is calculated using the formula:

$$Piority = remain\ value\ count * 10000 + degree * 100 + y * 10 + x$$

The count of remaining values holds a more significant weight than the degree, which in turn, holds more weight than the x and y coordinates. Consequently, variables with fewer remaining values are given precedence. In cases where variables have an identical number of remaining values, the one with a lower degree is chosen.

Furthermore, a global variable is used to track the number of assignments.

Result of instance 1:

```
 -------------------------
|     1 |     2 |       |
|     5 |     6 |   3   |
| 4 6   |     5 |       |
-------------------------
|       | 1   4 |       |
| 6     | 8     | 1 4 3 |
|       |   9   | 5   8 |
-------------------------
| 8     |   4 9 |   5   |
| 1     | 3 2   |       |
|     9 |       | 3     |
-------------------------
The 1 assignment:
The variable (5,6) is selected
The domain of this variable: [7]
The domain size: 1
The degree of this variable is: 8
The value 7 is assigned to it

The 2 assignment:
The variable (5,5) is selected
The domain of this variable: [5]
The domain size: 1
The degree of this variable is: 9
The value 5 is assigned to it

The 3 assignment:
The variable (6,6) is selected
The domain of this variable: [3]
The domain size: 1
The degree of this variable is: 8
The value 3 is assigned to it

The 4 assignment:
The variable (8,6) is selected
The domain of this variable: [8]
The domain size: 1
The degree of this variable is: 9
The value 8 is assigned to it


-------------------------
| 3 7 1 | 4 8 2 | 6 9 5 |
| 9 2 5 | 7 1 6 | 8 3 4 |
| 4 6 8 | 9 3 5 | 7 1 2 |
-------------------------
| 5 8 3 | 1 6 4 | 9 2 7 |
| 6 9 2 | 8 5 7 | 1 4 3 |
| 7 1 4 | 2 9 3 | 5 6 8 |
-------------------------
| 8 3 7 | 6 4 9 | 2 5 1 |
| 1 5 6 | 3 2 8 | 4 7 9 |
| 2 4 9 | 5 7 1 | 3 8 6 |
-------------------------
The program takes: 0.009580850601196289
```

Result of instance 2:

```
 -------------------------
|     5 |   1   |       |
|     2 |     4 |   3   |
| 1   9 |       | 2   6 |
-------------------------
| 2     |   3   |       |
|   4   |       | 7     |
| 5     |     7 |     1 |
-------------------------
|       | 6   3 |       |
|   6   | 1     |       |
|       |   7   |   5   |
-------------------------
The 1 assignment:
The variable (2,2) is selected
The domain of this variable: [7, 8]
The domain size: 2
The degree of this variable is: 12
The value 7 is assigned to it

The 2 assignment:
The variable (2,1) is selected
The domain of this variable: [6, 8]
The domain size: 2
The degree of this variable is: 11
The value 6 is assigned to it

The 3 assignment:
The variable (3,2) is selected
The domain of this variable: [3, 8]
The domain size: 2
The degree of this variable is: 10
The value 3 is assigned to it

The 4 assignment:
The variable (1,2) is selected
The domain of this variable: [8]
The domain size: 1
The degree of this variable is: 10
The value 8 is assigned to it

-------------------------
| 4 3 5 | 2 1 6 | 8 7 9 |
| 6 8 2 | 7 9 4 | 1 3 5 |
| 1 7 9 | 3 8 5 | 2 4 6 |
-------------------------
| 2 1 7 | 4 3 9 | 5 6 8 |
| 8 4 6 | 5 2 1 | 7 9 3 |
| 5 9 3 | 8 6 7 | 4 2 1 |
-------------------------
| 7 5 8 | 6 4 3 | 9 1 2 |
| 9 6 4 | 1 5 2 | 3 8 7 |
| 3 2 1 | 9 7 8 | 6 5 4 |
-------------------------
The program takes: 0.13703584671020508
```

Result of instance 3:

```
-------------------------
| 6 7   |       |       |
|   2 5 |       |       |
|   9   | 5 6   | 2     |
-------------------------
| 3     |   8   | 9     |
|       |       | 8   1 |
|       | 4 7   |       |
-------------------------
|     8 | 6     |   9   |
|       |       |   1   |
| 1   6 |   5   |   7   |
-------------------------
The 1 assignment:
The variable (3,1) is selected
The domain of this variable: [4, 8]
The domain size: 2
The degree of this variable is: 10
The value 4 is assigned to it

The 2 assignment:
The variable (2,1) is selected
The domain of this variable: [8]
The domain size: 1
The degree of this variable is: 12
The value 8 is assigned to it

The 3 assignment:
The variable (3,3) is selected
The domain of this variable: [1, 3]
The domain size: 2
The degree of this variable is: 8
The value 1 is assigned to it

The 4 assignment:
The variable (1,3) is selected
The domain of this variable: [3]
The domain size: 1
The degree of this variable is: 10
The value 3 is assigned to it

-------------------------
| 6 7 3 | 9 2 4 | 1 8 5 |
| 4 2 5 | 1 3 8 | 7 6 9 |
| 8 9 1 | 5 6 7 | 2 3 4 |
-------------------------
| 3 1 4 | 2 8 6 | 9 5 7 |
| 2 6 7 | 3 9 5 | 8 4 1 |
| 5 8 9 | 4 7 1 | 3 2 6 |
-------------------------
| 7 4 8 | 6 1 2 | 5 9 3 |
| 9 5 2 | 7 4 3 | 6 1 8 |
| 1 3 6 | 8 5 9 | 4 7 2 |
-------------------------
The program takes: 0.16341090202331543
```