**CS 171 Sudoku Project Shell: Python Edition**

TL;DR:

* Main File to modify is btsolver.py

1. RUNNING THE SHELL

In order to run the solver, enter

python main.py "NameOfInputFile" "NameOfOutputFile" "timeout" tokens

For example:

python main.py input.txt output.txt 600 FC ACP LCV MRV

The tokens will be saved in a tokens attribute for the solver class in btsolver.py so you can examine what tokens you have entered and act accordingly to break tiebreakers.

The input file will have 3 numbers followed by a visualization of the board.

The three numbers will be the board size, number of rows in each block, and the number of columns in each block

To generate a new Sudoku puzzle:

python problem\_generator.py "NameOfInputFile" "NameOfOutputFile"

For example, you can type

python problem\_generator.py input.txt output.txt

The input file will consist of 4 numbers

M N P Q

* M is the number of cells that will be initialized
* N is the size of the NXN grid
* P is the number of rows in each block
* Q is the number of columns in each block

EXAMPLE:

25 9 3 3

The output file will be your input file when running the solver

These are the files you will be working with in this project:

* **Variable.py** 
  + represents a variable in the CSP
* **Domain.py** 
  + represents the domain of a variable
* **Constraint.py** 
  + represents a constraint in the CSP
* **ConstraintNetwork.py** 
  + represents the CSP model
* **BTSolver.py** 
  + a backtracking solver
* **Trail.py** 
  + a trail of assignments that have occurred
* **Gameboard.py** 
  + represents the sudoku board and checks for error conditions
* **filereader.py** 
  + reads from input files
* **file\_writer.py** 
  + writes to the output files

The main file you will be modifying is the **btsolver.py.** This file implements that constraint satisfaction problem solvers you have studied in class and is an implementation of the pseudocode for Recursive Bactracking on **page 142** in **chapter 5** of the Russell-Norvig book.

1. TO DO For BTSolver

In order to improve upon the backtracking search, we can introduce heuristics and consistency checks. Having heuristics allows us to speed up our search by only checking promising assignments first. Meanwhile, consistency checks significantly prune the search space, reducing the number of potential assignments to check.

Variable Selection Heuristics

* Minimum Remaining Value: Select variable with fewest remaining values
* Degree: Select the variable involved with the largest number of constraints on other unassigned variables

Value Selection Heuristics

* Least Constraining Value: Select value which is involved with least conflicts with neighboring domains

Consistency Checks:

* Forward Checking: If you assign a variable X, if another variable Y is involved in a constraint with X, you prune the domain of Y with any inconsistent values Arc Consistency: For every pair of variables participating in constraints with each other, every value in one variable's domain must be satisfiable by at least one value in the other variable's domain.

1. TRAIL

You may notice the use of a trail in BTSolver. The trail is an important part of BTSolver as it is needed in order to enable backtracking. Whenever a variable is modified, a copy of the old value of the variable will be put onto the trail. When undo is called, changes made after the latest marker are undone, and the last marker is popped.

In order to set the heuristics based on the tokens passed as arguments, you can read **btsolver.py** and **main.py** to see the functions that can be utilized in order to solve this task.

You may notice some text files located in the directory. The **example\*.txt** and **PE\*.txt** are examples of generated boards. The **example\_generator.txt** file is an example of the format you need in order to generate a custom board. The **result\*.txt** are examples of how the output files will look like.

There are two files called **file\_reader.py** and **filewriter.py** that are used to read from files and write the results to a file. You should not need to modify these files but you can look at them in case you are curious.