CS171 Project Progress report #2: Forward Checking

Name: Haoming Huang, Jason Chen-Ju Hsieh

Date: February 20, 2016

1 Scope

The purpose of this assignment is to implement the forward checking component of Sudoku solver. Our team will be continuing off of the provide Java shell and changes made from the last assignment.

2 Progress

We've refactored the code to return solver status. We moved the status variable to BTSolver with default value error and have it update depending on timeout or success.

We tried to implement forwarding checking based on existing functions. After the solver assigns a value to a variable, it first calls assignmentCheck() and checks that no neighboring variables (down, across, and within smaller boxes) are assigned to the same values. If not, return false so the solver can backtrack. Otherwise, remove the assigned value from the domains of all neighboring variables and return true. However this is not enough to maintain consistency. We've changed it so that it performs a check on all variables for assignment, and only checking neighboring variables of assigned values. It also only remove values from domains only if the variable is assigned. If one of the neighboring variables have the same value as the variable itself, return false and backtrack.

We've looked ahead and changed the function to calculate time taken to consider preprocessing time, also added function to return preprocessing times of arc consistency.

3 Problems & Questions

It was difficult to understand how to use the tools already present to implement forward checking. It was unclear at first whether some already existing component(s) can be used or most of it is yet to be coded. At first we thought that assignmentCheck() is necessary as part of forward checking but eventually we found another way. It was also difficult to make sure that FC is keeping everything consistent, because it was initially not consistent enough, and it's hard to thinking about and visually represent what the solver is doing or what the state is. Eventually we arrived at the correct result.

4 Results

The program takes in correct inputs (input file, output file, timeout, and FC token), correctly solves the Sudoku puzzle using the back track search implementation and uses the forwarding checking consistency check to remove inconsistent domain values to speed up the searching process, and outputs a file with the correct format. Back track search with FC implemented is extremely fast with less node counts especially on larger puzzles

Appendix:

```
private boolean forwardChecking()
{
	for(Variable v : network.getVariables())
	{
		if(v.isAssigned())
		{
			for(Variable vOther : network.getNeighborsOfVariable(v))
			{
				if (!vOther.isAssigned())
					vOther.removeValueFromDomain(v.getAssignment());
				if (v.getAssignment() == vOther.getAssignment())
					return false;
			}
		}
	}
	return true;
}
```