NORWEGIAN UNIVERSITY OF SCIENCE AND TECHNOLOGY
TDT4136 – INTRODUCTION TO ARTIFICIAL INTELLIGENCE

Assignment 5: Solving Constraint Satisfaction Problems

Per Magnus Veierland
permve@stud.ntnu.no

October 30, 2015

1. The *Python* program `vi.app.sudoku` implements a Constraint Satisfaction Problem (CSP) solver based on the BACKTRACKING-SEARCH algorithm described in **AIMA** and implemented in the `vi.csp` namespace. It reuses a generalized REVISE algorithm called REVISE* and a generalized AC-3 algorithm called GENERALIZED-ARC-CONSISTENCY which were both developed for IT3105. The program can be run in two modes; binary or general. In binary mode the Sudoku puzzle input will be parsed and 972 binary arc constraints will be constructed for 81 variables (see Table 1); while in general mode 27 general arc constraints will be constructed for 81 variables (see Table 2).

3. In the easier puzzles there are no backtracking failures. This is likely because they have several valid solutions, which makes it more likely that early assumptions will be correct, and that the puzzle values are such that they allow for a rapid domain reduction for the non-fixed variables.

   With the harder puzzles there is a greater number of backtracking failures. This is likely because there are few valid solutions, and because the resulting constraints are not able to easily reduce the domain sizes. This makes it more likely that early assumptions will be incorrect such that the search fails later on, which results in backtracking.

   When using general arc constraints where each constraint involves a whole row, column or box; there are no backtracking failures. This is because the GENERALIZED-ARC-CONSISTENCY algorithm is able to fully reduce all domains to unity in one application, meaning that all variables are sufficiently constrained such that the puzzle can be solved by pure inference.

Puzzle	Variables	Constraints	Backtrack Calls	Backtrack Failures
`easy.txt`	81	972	1	0
`medium.txt`	81	972	1	0
`hard.txt`	81	972	25	21
`veryhard.txt`	81	972	31	27

Table 1: Sudoku BACKTRACKING-SEARCH with binary arc constraints.

Puzzle	Variables	Constraints	Backtrack Calls	Backtrack Failures
`easy.txt`	81	27	1	0
`medium.txt`	81	27	1	0
`hard.txt`	81	27	1	0
`veryhard.txt`	81	27	1	0

Table 2: Sudoku BACKTRACKING-SEARCH with general arc constraints.

8	1	7	6	9	3	2	5	4
5	2	3	7	4	1	9	8	6
9	6	4	5	2	8	3	1	7
6	7	1	8	5	9	4	2	3
3	4	8	2	6	7	1	9	5
2	9	5	1	3	4	7	6	8
7	5	6	3	1	2	8	4	9
1	3	9	4	8	5	6	7	2
4	8	2	9	7	6	5	3	1

(a) `easy.txt`

6	3	5	9	2	7	1	4	8
4	8	2	1	6	5	9	7	3
9	7	1	3	8	4	2	6	5
5	2	9	7	1	6	3	8	4
8	4	6	2	9	3	5	1	7
7	1	3	5	4	8	6	9	2
2	9	4	8	5	1	7	3	6
1	6	7	4	3	2	8	5	9
3	5	8	6	7	9	4	2	1

(b) `medium.txt`

8	9	2	3	5	1	7	6	4
1	3	4	8	7	6	5	2	9
5	7	6	4	9	2	3	1	8
7	1	5	6	2	9	4	8	3
4	6	3	5	1	8	2	9	7
2	8	9	7	4	3	6	5	1
3	5	1	2	8	4	9	7	6
9	4	7	1	6	5	8	3	2
6	2	8	9	3	7	1	4	5

(c) `hard.txt`

3	4	7	9	6	1	8	2	5
6	5	8	4	2	3	9	7	1
9	1	2	5	7	8	4	3	6
5	3	4	2	9	7	6	1	8
7	6	9	1	8	4	2	5	3
2	8	1	3	5	6	7	9	4
1	7	6	8	3	2	5	4	9
8	9	3	7	4	5	1	6	2
4	2	5	6	1	9	3	8	7

(d) `veryhard.txt`

Figure 1: Sudoku solutions