# Answer Set Solving in Practice

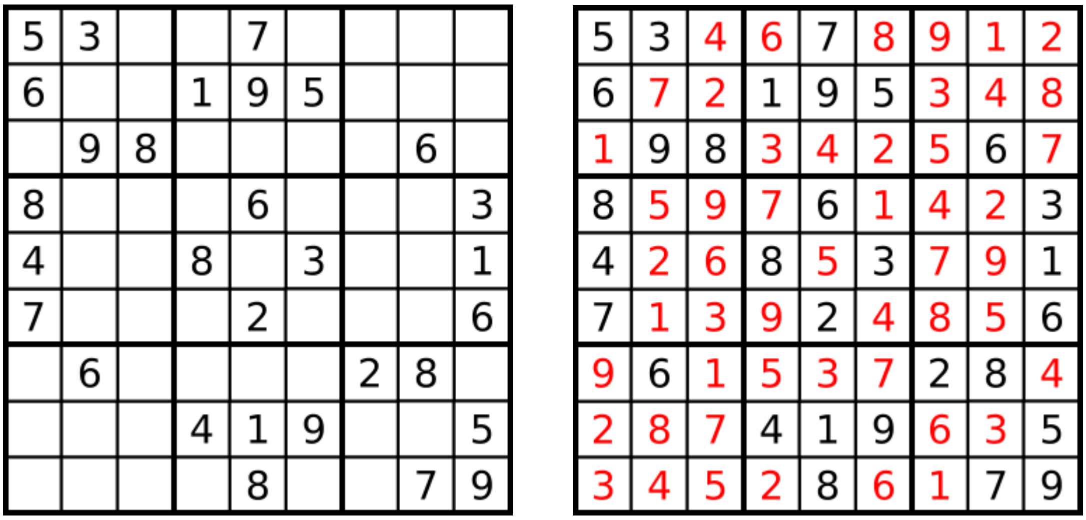
Prof. Torsten Schaub, Javier Romero

University of Potsdam — Winter Semester 2014/2015

Project 1 (Sudoku)

**Problem Description.** The task of this project is to solve a Sudoku[1] puzzle using ASP. The goal of the game is to fill a 9x9 grid with digits so that each column, each row and each of the nine 3x3 sub-grids that compose the grid contains all numbers from 1 to 9. In other words, the grid has to be filled with numbers from 1 to 9 so that the same number does not appear twice in the same row, column or in any of the nine 3x3 subregions of the 9x9 playing board. Initially the grid is partially filled.

One example is shown in Figure 1. The left figure shows the initial configuration, and the right figure shows the same puzzle with solution numbers marked in red.



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

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

Figure 1: A sudoku and its solution.

**Representation in ASP.** The initial state of the grid is represented by facts of predicate `initial/3`:

```
initial(X,Y,N).   % initially cell [X,Y] contains number N
```

For instance, the example shown in Figure 1 is represented by the following facts:

```
initial(1,1,5). initial(1,2,3). initial(1,5,7).
initial(2,1,6). initial(2,4,1). initial(2,5,9). initial(2,6,5).
initial(3,2,9). initial(3,3,8). initial(3,8,6).
initial(4,1,8). initial(4,5,6). initial(4,9,3).
initial(5,1,4). initial(5,3,8). initial(5,6,3). initial(5,9,1).
initial(6,1,7). initial(6,5,2). initial(6,9,6).
initial(7,2,6). initial(7,7,2). initial(7,8,8).
```

[1] http://en.wikipedia.org/wiki/Sudoku

```
initial(8,4,4). initial(8,5,1). initial(8,6,9). initial(8,9,5).
initial(9,5,8). initial(9,8,7). initial(9,9,9).
```

The solution is represented by facts of predicate `sudoku/3`:

```
sudoku(X,Y,N).   % the sudoku in cell [X,Y] contains number N
```

For instance, the solution of Figure 1 consists of the following atoms:

```
sudoku(1,1,5) sudoku(1,2,3) ... sudoku(1,8,1) sudoku(1,9,2)
...
sudoku(9,1,3) sudoku(9,2,4) ... sudoku(9,8,7) sudoku(9,9,9)
```

**Framework.** In the `sudoku.zip` archive at Moodle you will find five example instances. You have to submit a file named `sudoku.lp` that contains the following line (and no more `#show` statements) so that in the output only the atoms of predicate `sudoku/3` appear:

```
#show sudoku/3.
```

**Formalities.** You can work on the solution alone or in groups of two people. Different groups have to submit different solutions, in case of plagiarism all groups involved will fail the project. Please submit your encoding by **Thursday 13th, 2014** via YETI (All group members have to create a YETI account!). Be sure to submit your encoding in a file named `sudoku.lp` containing only lowercase letters.

We will test your encoding with the five provided instances as well as additional instances. Your encoding has to find the correct solutions for all instances. This will be tested automatically by YETI after you uploaded the encoding (with a slight delay). If your solution is not correct then YETI will display an error message. Please correct any errors that occur on your own or contact us if you get stuck.

**Tips:**

- To begin with, it may be easier to represent a 4x4 sudoku and once this is done, modify it to handle the 9x9 case.
- Commands to find all stable models look as follows:

  ```
  $  gringo-4 sudoku.lp example.lp | clasp3 0
  ```

- If you are stuck you can contact us. We will do out best to answer all your questions. You can send us questions and remarks via Moodle (forum/wiki) or send them via mail to `asp@lists.cs.uni-potsdam.de`.
- Start as soon as possible to avoid running out of time. (However, if you still realize that you have problems making it before the deadline, please contact us instead of copying another solution).