

# B(E)3M33UI — Exercise

## Constraint satisfaction problems

Radek Mařík, Jiří Spilka

May 16, 2017

### 1 Constraint satisfaction problems (CSP)

The goal is to understand the CSP and use it to implement a Sudoku solver.

In the **planning** exercise we have searched for a sequence of actions that have reached a predefined goal, where the sequence (path) was important. On contrary, in the **CSP** we need to find a goal (state of world that satisfies constraints), but the path to solution is not important.

**Constraint satisfaction problems:**

- Define possible worlds in term of **variables** and their **domains**
- Specify **constraints** to represent real world problems
- Verify whether a possible world satisfies a set of constraints

More formally:

- A finite set  $\mathcal{V}$  of variables  $V_i, i = 1, \dots, n$
- A non-empty domain  $D_i = \text{dom}(V_i)$  of possible values for each variable  $V_i \in \mathcal{V}$
- A finite set of constraints  $C_1, C_2, \dots, C_m$ . Each constraint  $C_i$  limits the values that variables can take for subsets of the variables

#### 1.1 CSP for Sudoku – preliminaries

The sudoku puzzle is composed of 81 squares; the following labeling is typically used:

- columns: 1 – 9
- rows: A – I
- unit: collection of nine squares (column, row, or box)
- peers: squares that share a unit

**A puzzle is solved if the squares in each unit are filled with a permutation of the digits 1 to 9.**

Example of grid:

```

A1 A2 A3| A4 A5 A6| A7 A8 A9
B1 B2 B3| B4 B5 B6| B7 B8 B9
C1 C2 C3| C4 C5 C6| C7 C8 C9
-----+-----+-----
D1 D2 D3| D4 D5 D6| D7 D8 D9
E1 E2 E3| E4 E5 E6| E7 E8 E9
F1 F2 F3| F4 F5 F6| F7 F8 F9
-----+-----+-----
G1 G2 G3| G4 G5 G6| G7 G8 G9
H1 H2 H3| H4 H5 H6| H7 H8 H9
I1 I2 I3| I4 I5 I6| I7 I8 I9

```

**Task 1:** For the sudoku: define variables, domain, and set of constraints.

**Task 2:** Sketch a constraints graph.

**Notes:**

- A minimal partial graph for column, row, and box will suffice. This should help you to realize: the complexity and that it could be solved as map coloring problem (cf. lectures).

**Task 3:** Can we solve the sudoku using some ordinary search, e.g. depth-first-search?

**Task 4:** Implement algorithm that is able to solve the following sudoku (easy and hard).

```

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

```

**Notes:**

- Use either **AC-3** or **Backtracking**, cf. lectures.
- Implement some heuristics, e.g. Minimum remaining values (variable selection), Least constraining value (value ordering), or Forward checking (inference).

The solutions:

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

## 2 Happy solving!

Complete the exercise as a homework, ask questions on the forum, and upload the solution via BRUTE!