# Computational Intelligence

Prof. Torsten Schaub, Laferrière François, Cedric-John Martens, Jan Heuer

Universität Potsdam (Wissensverarbeitung und Informationssysteme) — Sommersemester 2020

Praktikumsaufgabe 3 (Yosenabe)

Problem Description. The task of this project is to solve the Japanese grid puzzle Yosenabe[1] using ASP. To illustrate the rules of the game, consider a grid as follows:
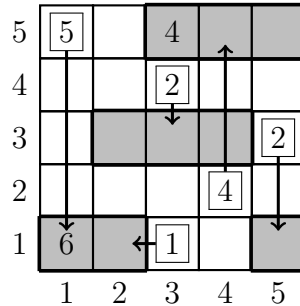
Given such a grid,[2] initially without the arrows, the task is to move each number surrounded by a frame into one of the gray areas along a straight line, respecting the following conditions:

1. The ways of any two moved numbers must not cross or meet at any grid cell.

2. Each gray area must be populated with at least one moved number.

3. An area may be associated with a (positive) goal number, shown within it. If so, the numbers moved into the area must sum up exactly to the goal.

The (unique) solution indicated by arrows above fulfills these conditions. While a number can be moved through an area, such as 4 above, you may assume that a move stops at the first cell w.r.t. its direction of the area into which it leads. That is, regardless of the cell to which number 5 is moved, it cannot help to move 1 to the cell in the lower left corner as the move enters the same area already before.

Representation in ASP. An instance like the grid above consists of facts as follows:

```
cell(1,1).    cell(1,2). ...  cell(5,4).   cell(5,5).   number(1,5,5).
area(1,1,1).  area(2,1,1).                 goal(1,6).   number(3,1,1).
area(2,3,2).  area(3,3,2).  area(4,3,2).                number(3,4,2).
area(3,5,3).  area(4,5,3).  area(5,5,3).  goal(3,4).   number(4,2,4).
area(5,1,4).                                            number(5,3,2).
```

Note that the first two arguments in facts over `area/3` and `number/3` provide grid cells, and the third an area identifier or a number to move, respectively. If an area is associated with a goal number, its identifier is reused as first argument in a fact over `goal/2`, and the goal is given by the second.

---

[1]http://www.nikoli.co.jp/en/puzzles/yosenabe.html

[2]Some Yosenabe puzzles to practice can be found here: http://www.janko.at/Raetsel/Yosenabe/

The moves in a solution shall be provided by atoms over the predicate `target/4` in a stable model, expressing a move in terms of coordinates of the initial cell as well as the cell to which a number is moved. Along with the instruction '`#show target/4.`', a correct solution in file `yosenabe.lp` yields the following `clingo` behavior:

```
$ clingo-5.4.0 instance01.lp yosenabe.lp
clingo version 5.4.0
Reading from instance01.lp ...
Solving...
Answer: 1
target(1,5,1,1) target(3,1,2,1) target(3,4,3,3) \
target(4,2,4,5) target(5,3,5,1)
SATISFIABLE
```

Framework. In the `yosenabe.zip` archive at Moodle you will find seven example instances. You have to submit a file named `yosenabe.lp`, included as template in `yosenabe.zip`, that contains the following line (and no more `#show` statements) so that in the output only the atoms of predicate `target/4` appear:

```
#show target/4.
```

Formalities. You can work on the solution alone or in groups of three people. Different groups have to submit different solutions, in case of plagiarism all groups involved will fail the project. Please submit your encoding by **Friday, August 21th, 2020** via YETI. (All group members have to create a YETI account!) Be sure to submit your encoding in a file named `yosenabe.lp` containing only lowercase letters.

We will test your encoding with the seven provided instances as well as additional instances. Your solution has to correctly encode all solutions for every instance. (However, our test instances usually have a unique solution.) This will be tested automatically by YETI after you uploaded the encoding (with a slight delay). In case your solution is not correct, 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, play some Yosenabe puzzles online[2] to get acquainted with the rules.

- Note that the four directions (up, down, left, right) are entirely symmetric and vary only by coordinate differences between cells. Please avoid code duplication due to referring to directions more specifically than needed; e.g., distinguishing directions by using different names for auxiliary predicates is not a good idea.

- If you are stuck you can contact us. We will do our best to answer all your questions. You can send us questions and remarks via Moodle (forum) or send them via mail to `ci@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.)