Artificial Intelligence

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

University of Potsdam — Winter Semester 2022/2023

Project 3 (Creek)

Problem Description. The task of this project is to solve a Creek puzzle using ASP. The goal of the game is to to blacken some cells of a rectangular grid so that the following conditions are met:

1. All white cells (non-blackened) must form an orthogonally connected area, i.e., it must be possible to get from any white cell to any other white cell by moving only horizontally and vertically (never diagonally!), and without having to cross a black cell.
2. A number in a circle indicates how many of the 4 adjacent cells must be blackened.

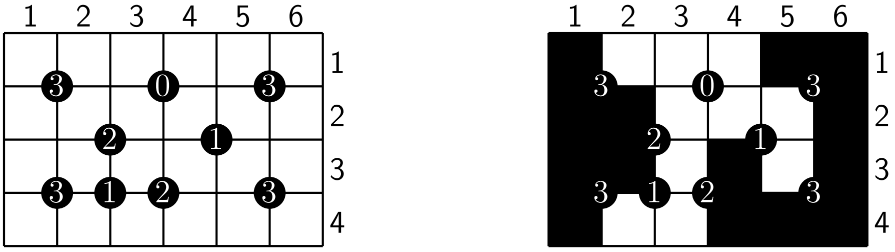

Figure 1: A creek puzzle (left) and a solution (right).

One example is shown in figure 1. The left side shows a 6x4 grid with nine numbers given in nodes. The right side shows a (here the unique) solution for the grid. Observe that the number of blackened cells adjacent to the (nine) nodes corresponds to each given number, and that all white cells are connected to each other. The Creek puzzle can also be tried online at the following site: `http://www.janko.at/Raetsel/Creek/index.htm`

Representation in ASP. We represent the cells of a rectangular grid and the nodes with numbers by facts of the following form:

```
cell(X,Y).    % there is a cell with coordinates (X,Y)
hint(X,Y,N). % N of the (maximum four) cells at coordinates (X,Y),
             %(X+1,Y), (X,Y+1) and (X+1,Y+1) must be blackened
```

For example, the grid shown on the left in Figure 1 left is described by the following facts:

```
cell(1..6,1..4).
hint(1,1,3). hint(2,2,2). hint(3,1,0). hint(4,2,1). hint(5,1,3).
hint(1,3,3). hint(2,3,1). hint(3,3,2).              hint(5,3,3).
```

A solution, i.e. a set (of coordinates) of blackened cells, is represented by atoms of the predicate `black/2`:

```
black(X,Y)   % the cell at the coordinates (X,Y) is blackened
```

The solution shown on the right in figure 1 is described by the following atoms:

```
black(1,1)                                  black(5,1) black(6,1)
black(1,2) black(2,2)                                  black(6,2)
black(1,3) black(2,3)            black(4,3)            black(6,3)
black(1,4)                       black(4,4) black(5,4) black(6,4)
```

Framework. The directory `asp` contains the files that you need for the project. In the directory `asp/instances` you can find the instances, and in the directory `asp/solutions` you can find their solutions in `json` format.

You have to submit a file named `creek.lp`, included as a template in the directory asp, that contains the following line (and no more `#show` statements), so that in the output only the atoms of predicate `black/2` appear:

```
#show black/2.
```

You can check if your encoding solves correctly all instances by running the `Python` script `test.py` as follows:

```
python3.6 asp/test.py -e asp/creek.lp -i asp/instances -s asp/solutions -t 180
```

In this case, the timeout for each instance is set to 180 seconds, but you can use any other value instead.

For help, type `python3.6 asp/test.py --help` .

We recommend you to work locally in your computer, using your own installation of clingo.

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. Your solution has to correctly encode all solutions for every instance. In fact, our test instances usually have several solutions. This is tested automatically by the script `test.py`.

We will send you further instructions about the submission process from Moodle.

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 either via Moodle or by email.

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.