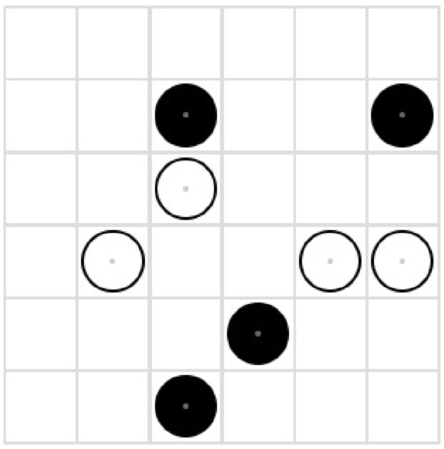
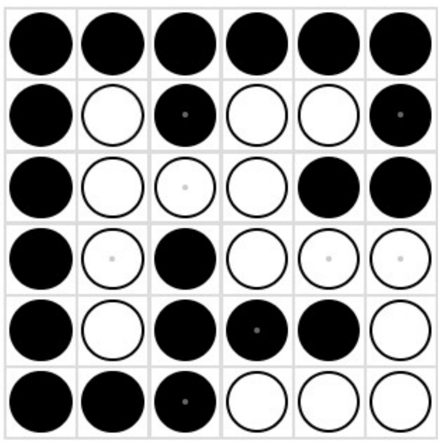
KR: Exercise 1

Yin Yang puzzle


Initial configuration


Solution

Introduction

In this exercise we will play with clingo to find solutions to the Yin Yang puzzle. Yin Yang is a puzzle that consists in coloring each cell in a grid of N x N either as black or white. Some initial cell colors are fixed (see initial configuration above) and the constraints for a final solution are the following:

1. All black cells must be orthogonally connected altogether in a single group
2. All white cells must be orthogonally connected altogether in a single group
3. Areas of 2x2 of the same color are not allowed

Orthogonally connected means that there must be a path from any black cell to another black cell by exclusively making vertical or horizontal movements (diagonal movements are not considered). The same applies for any pair of white cells.

Steps

1. Encode the Yin Yang problem as an ASP program that solves the puzzle for any instance. This program is our Knowledge Base and will be called **yinyangKB.lp**

2. Each puzzle instance will be provided as an ASCII file domXX.txt with the following format. Each line contains **n points** followed by a newline. A dot "." represents a regular grid point to be filled, a "0" represents a white circle and "1" a black circle. As an example, the input file for the initial configuration in the picture above could look like:

```
......
..1..1
..0...
.0..00
...1..
```

You will create a python program called **encode.py** that takes each domXX.txt file as an input and creates an **domXX.lp** file describing the instance as a set of ASP facts using your representation of the problem. An example of use could be:

```
python3 encode.py dom02.txt dom02.lp
```

Warning: domain files domXX.lp can only contain **facts and constants**. Other rules or constraints are not allowed in domain files.

3. Finally, we will translate back the answer set into a complete Yin Yang solution, printing the final result in standard output.

```
111111
101001
100011
101000
101110
111000
```

To print this output we will use the python program **decode.py** as follows:

```
python3 decode.py yinyangKB.lp dom02.lp sol02.txt
```

This program uses two predicates, "gridsize(N)" that specifies the size of the grid NxN, and predicate "_drawcircle(X,Y,C)" that prints in row X and column Y the number 1 if C=black, and the number 0 if C=white. Using the same predicates, we can also display the output graphically using files **display.py** and **drawyinyang.lp** as follows:

```
    python3 display.py yinyangKB.lp dom02.lp
drawyinyang.lp
```

The file **examples.zip** contains 12 domains domXX.txt and their corresponding solutions solXX.txt.

The maximum grade for this exercise is **1.66 points = 16.6% of the course**. Delivery: use the MOODLE assignment (Campus Virtual). Exercises can be made by groups of 2 students at most. If so, only one student is required to deliver the files in moodle, but all source files must contain the names of the two group members.

Delivery: upload all files in a .zip including a README.txt with your names and describing how to use the code, if needed.
Deadline: **March 28th 2025 (23:59)**.

IMPORTANT: Assignments **may be required to pass through a defense exam**. If this is the case, it will be announced in the first, preliminary list of grades.

Maintained by Pedro Cabalar. Last update May 7th, 2024