



Cellular Automaton: Rule 110

Parallel Computing

Goals

★ Learn basic constructions of C++.

★ **Relevant videos:**

- Hello C++
- Primitive types
- Expressions
- Statements
- Functions and references

Deliverables

1. The code on your Github repository generated by clicking here: <https://classroom.github.com/a/SHxf1g2L>
2. **Reviewer:** Paul Aromolaran (Github: PaulAroo)
3. **No automated review**

Exercise 1 – Rule 110

1. Implement the cellular automaton Rule110 (file `rule110.cpp`) as presented on Wikipedia. Write the simulation in a loop using two arrays of cells (the current array and the next array). Note that the first and last cell of the array is always '0'.
2. Decompose the program into several functions:

```
print(cells);  
simulate(steps, current, next);  
simulate_step(current, next);
```

Exercise 2 – Pattern Counting

We now seek to count the number of times a pattern occurs during the simulation. A pattern is simply a sequence of 0 and 1, for instance the pattern 11100 occurs at index 2 in the sequence 0111100. If the pattern occurs in several places during one iteration, count it only once!

Exercise 3 – Command-line arguments

In order to facilitate the automatic testing of your code, provide the following arguments:

- `--iter 100`: The number of iterations for the simulation loop.
- `--init file.txt`: read the file `file.txt` which contains the initial configuration of the automaton. Example:

```
10
0100111000
```

The first line contains the size of the automaton and the second line the initial configuration.

- `--pattern file.txt`: read the file `file.txt` which contains the pattern to detect in the automaton. It follows the same format than the file passed to `--init`.

Output: print the number of times the pattern was detected and do not print anything else!