The λ Parameter and Sampling

Author(s): Koen Koning & Michael Lees
Edited by: Johannes Blaser & Frederick Kreuk

Assignment Objectives
In this lab exercise you will use the CA logic you have developed in the previous assignment and extend it to perform experiments on the lambda parameter. Please make sure you have a solid grasp on the concept behind the λ parameter, (Shannon) entropy, and transient lengths. This exercise has three main objectives:

1. Extend your CA implementation such that it can build rule tables based on Langton's parameter (λ).
2. Recreate some of Langton's experiments on unidimensional CAs.
3. Explore different values of λ and observe its impact on a CA and the Wolfram CA classifications.

The Implementation

A good starting off point is the code you have developed in lab 2. You should then implement a way to generate a CA rule table based on Langton's λ parameter. What exactly this value entails and what is is useful for is extensively discussed in Langton's 1990 paper: [1]. The slides also cover this as well as practical ways to convert the λ parameter into a rule table.

The two main ways that are covered are the *random-table* method and the *table-walkthrough* method. The random method is conceptually very straight-forward and such you will implement the **table-walkthrough** method in your assignment. Of course you are free to implement both.

Please note that your implementation should neatly support the specification of many of these parameters. Last week you (should have) already implemented a way to specify k and r via the GUI. In this assignment the λ value should be equally adjustable via the GUI as well as any other parameters you have used.

The Experiments

Because the λ value determines what a rule table will look like the λ value also determines the quantitative behaviour of a cellular automaton. To explore exactly this relation you should perform a number of experiments and visualise your findings.

Note that all of the experimental requirements defined in the first assignment's description still hold. Sufficient repetitions; varied initial conditions; **sufficiently high values for k and r**; etc. are all a **must**. Random results are meaningless and as a result are insufficient for a proper experiment (or a good grade).

Some suggestions for an interesting experiment could be as follows:

- Shannon Information Entropy. Note that this can be determined in a variety of ways: per cell, per configuration, on the rules, etc. Do not forget to mention your decisions in your caption.
- Transient Length. I.e. how many iterations does it take for the CA to stop exhibiting new behaviour. Make sure to read up on what ways there are to determine this **efficiently**.

As in the previous assignment your results ought to be summarised in a single figure including properly labeled axes, error bars, and a proper caption. Again, aim for around 250 words. The λ parameter should be the central parameter in your results. Like last week you need not write an entire encyclopedia but rather a succinct but detailed description of your experimental set-up, results, and a discussion thereof.

Grading

You should submit to Canvas at least the following things:

1. Program code
2. Image file (with the caption included or with the caption as a separate text file)
3. Any scripts you used for generating your figure(s).

Make sure that your figure can be reproduced by running a simple, obvious command. Running `python generate_figure.py` is correct, a twenty-step sequence is not.

Your grade is based on both the correctness of your implementation, ease of use, experimental set-up, how interesting your experiments and results are, and how well they are visualised, discussed, and interpreted.

Your code will be tested for plagiarism using special-purpose software.

This assignment makes up 15% of the grade for the practical assignments for the CA part of this course.

Deadline: Wednesday, 19$^{\text{th}}$ of February 2020, 23:59.

References

[1] Chris G Langton. Computation at the edge of chaos: phase transitions and emergent computation. *Physica D: Nonlinear Phenomena*, 42(1):12–37, 1990.