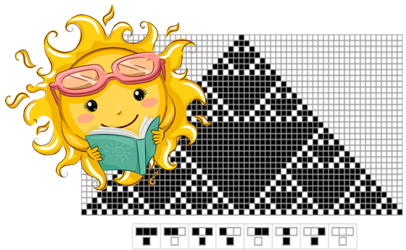
UNDERSTANDING THE CONVERGENCE IN ALPHA-ASYNCHRONOUS CELLULAR AUTOMATA

Bheemakonda Raja Maheswar[†] and Revanth D. Rampal[†]
Mentor: Souvik Roy[⁎]

[†]RGUKT-AP, RK Valley, Andhra Pradesh, India
[⁎]Indian Institute of Engineering Science and Technology, Shibpur, India



August 29, 2022

This paper contains the keen observations and research on factors of convergence, converging rules and attractors of Alpha-Asynchronous Cellular automata. Our experimentation will be on elementary cellular automata from alpha values 0.1 to 1.0 for a set of 88 minimal rules. In this process our main focus is to track converging rules and record them. To get this information we developed a program that takes alpha values and rules as input and returns above mentioned information as output. Our second focus will be on recording the attractors. In this project we consider 2-state (0 or 1) 2-neighborhood one Dimensional Cellular automata. In which cell state transition takes place with help of $0 - 255$ Binary rules. In simple words an initial configuration is set and some rule is applied from the given rules to it to reproduce further configurations and self replication. For every possible neighborhood there will be a respective state assigned to it. According to this evolution takes place in configurations.

Generally in Cellular Automata cell updation or state transition takes place synchronously considering time as discrete, but in case of asynchronous Cellular Automata the cell updation takes place asynchronously irrespective of time. Convergence is a phenomenon of where a system cannot come back to its initial configuration after moving out of the configuration. Rather the system, in course of its evolution, settles down to a configuration (convergence point) or to a small set of configurations. A range of physical systems show this convergence phenomenon. During the evolution of α-asynchronous cellular automata at one particular stage all configurations converge to a fixed point (fixed configuration). We call this fixed point. Our main interest is on type 4 (see in Figure. 1),

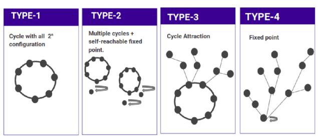


Figure 1: 4 types of evolutions observed in α- asynchronous CA

to recognize all of the converging rules under different alpha values and we record them. We developed a program which generates output that consists of CA sizes that shows convergence for every alpha value of a set of rules under periodic boundary conditions. There are 256 rules in ECA but some of the rules show the evolution of the same behavior. So we take rules of unique behavior and avoid the similar types. The program takes rules as input. And outputs two text files. First File consists of information about for every rule under all alpha values whether the rule shows convergence or not. Second File consists of all attractors with respect to rule only if that rule Shows convergence. From the information obtained by the first File we plot a table that shows convergence properties of the given 88 rules under all alpha values i.e. from 0.1 to 1.0 with CA sizes 4 to 10. From the Figure. 2, the information we took the rules as convergible if and only if they are totally covergible under all given conditions. To show an example of table, In Figure. 2 rule 0 characteristics

Rule Number	Alpha	CA Size 4	CA Size 5	CA Size 6	CA Size 7	CA Size 8	CA Size 9	CA Size 10
RULE 0	0.1	✓	✓	✓	✓	✓	✓	✓
	0.2	✓	✓	✓	✓	✓	✓	✓
	0.3	✓	✓	✓	✓	✓	✓	✓
	0.4	✓	✓	✓	✓	✓	✓	✓
	0.5	✓	✓	✓	✓	✓	✓	✓
	0.6	✓	✓	✓	✓	✓	✓	✓
	0.7	✓	✓	✓	✓	✓	✓	✓
	0.8	✓	✓	✓	✓	✓	✓	✓
	0.9	✓	✓	✓	✓	✓	✓	✓
	1.0	✓	✓	✓	✓	✓	✓	✓

Figure 2: Rule-0 satisfying all the given condition.

are shown under every condition i.e., Alpha value and CA size the rule shows convergence.(converged represents tick mark, first tick tells for alpha 0.1 and CA size 4 the evolution is converged). Because of this we finalize that Rule 0 shows convergence. After performing multiple tests and program executions we found the following rules that show convergence in α -Asynchronous Cellular Automata.The list of rules (47 rules) show complete convergence- 0, 2, 4, 8, 10, 12, 22, 24, 32, 36, 38, 40, 42, 44, 54, 56, 72, 74, 76, 78, 90, 104, 106, 108, 128, 130, 132, 136, 138, 140, 142, 146, 150, 152, 154, 156, 160, 162, 164, 168, 170, 172, 178, 184, 200, 204, 232. Attractors of a rule cannot be recorded because we use a *randomizer function* to the cells of eca, because of this every time we execute the program, the randomizer function may produce 0th or originally existing state. Due to this phenomena, the evolution of cellular automata is different for every cycle of execution.

If the random value generated by the randomizer function is less than alpha value for a cell in Lattice then that cell state is set to zero in Alpha Asynchronous CA. (Remaining cells follow the transition function in other words, RULE).

$$Random(0.1\ to\ 1.0) < \alpha \tag{1}$$

Here we are taking an example of CA size 12 assuming some alpha value. When we execute the program 2 times, we look two scenario (see in Figure. 3).

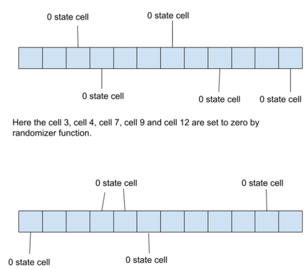


Figure 3: Execution of the program for 2 times

In second case (see Figure 3, Cell 1, Cell 4, Cell 5, Cell 6, Cell 11 are set to state 0. Due to this phenomena it is impossible to get the same attractors every time. So with this reason we are unable to record attractors.