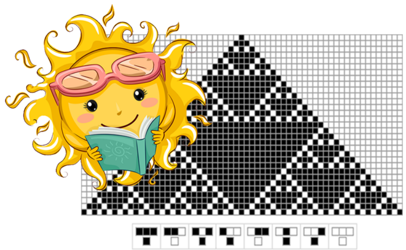
ISOMORPHISM IN CELLULAR AUTOMATA

Vicky Vikrant†
Mentor: Sukanya Mukherjee⁕

†National Institute of Technology, Tiruchirappalli, India
⁕Institute of Engineering & Management, Kolkata, India



August 22, 2022

This work focuses on the isomorphism of cellular automata (CAs). Two cellular automata are said to be isomorphic if their configurations evolve in the similar way. As a model, here we use non-uniform elementary cellular automata under null boundary and discover few inherent properties of CAs to decide whether the given cellular automata are isomorphic.

Cellular automata are discrete dynamical systems which produce complex global behaviour using simple local computation. The configurations of a cellular automaton (CA) evolves with time. A configuration is said to be *reachable* if it has some predecessor configuration; otherwise, the configuration is *non-reachable*. Let G be a CA and C be the configuration space. Let $G(x) = y$ and $G(z) \neq x$ where y is reachable configuration and y is reachable from x; Now, c is not reachable from any configuration z. Therefore, x is non-reachable. In finite CA, every configuration ultimately reaches to some *cycle*. The configuration which is used to form a cycle, is called a *cyclic* configuration. The length of a cycle is determined by the number of cyclic configurations it possesses. The configuration which is not cyclic, is called as *acyclic*. The cycle structure of a CA is the collection of the number of cycles along with their lengths. If all the configurations of a CA are cyclic, then such CA is *reversible*; otherwise, the CA is *irreversible*.

Let G_1 and G_2 be two ECAs of same size having same configuration space C. G_1 and G_2 and are said to be *isomorphic* if there exists a bijective mapping $\pi : C \to C$ such that $G_1(x) = y$ iff $G_2(\pi(x)) = \pi(y)$ where $\forall x, y \in C$. It is very challenging to figure out π and not much work has been found on the isomorphism in cellular automata. Thus we are motivated to find some intrinsic properties of cellular automata which play the instrumental role to decide of isomorphism in CAs. In our work, we use *reachability tree*, which is a rooted and edge-labelled binary tree that decides the reachable and non-reachable configurations of CA. This tool is used to develop some properties on the isomorphism in cellular automata.

Property1: Two CAs G_1 and G_2 are said to be *isomorphic* if both be reversible or both be irreversible but the converse is not always true.

If **Property1** is not satisfied by the given G_1 and G_2, then they are not isomorphic. Therefore, we test the reversibility of the given non-uniform CAs (using the algorithm of $O(n)$) and if we find that G_1 is reversible CA and G_2 is irreversible CA, then they are not isomorphic. Let $(10, 150, 90, 20)$ and $(2, 150, 90, 20)$ are given CAs. Here, $(10, 150, 90, 20)$ is reversible CA but $(2, 150, 90, 20)$ is irreversible CA as 2 can not be the first rule of any reversible CA. Therefore, Let $(10, 150, 90, 20)$ and $(2, 150, 90, 20)$ are not isomorphic.

As **Property1** is a necessary condition to decide isomorphism in cellular automata, we need to figure out some more properties on CAs when G_1 and G_2 both be reversible or both be irreversible.

Now, in reversible CAs, as all configurations are cyclic, for deciding isomorphism, we need to check whether they have same number of cycles along with same lengths.

Property2: Let G_1 and G_2 be reversible cellular automata. They are said to be isomorphic iff both the CAs have the same cycle structure.

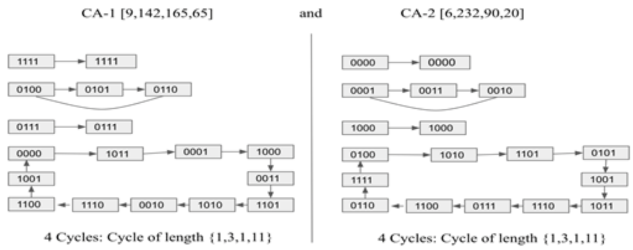


Figure 1: Reversible CAs

Here, $(9, 142, 165, 65)$ and $(6, 232, 90, 20)$ are isomorphic CAs as they have same cycle structure $[2(1), 1(3), 1(11)]$. Though few works have been reported on the computing of cycle structure of reversible CAs but its inherent hardness still makes it as a challenging problem. So, we are motivated to figure out some properties of isomorphic reversible CAs.

Next we focus on irreversible CAs. To study the isomorphism in irreversible CAs, other than cycle structures of those CAs, the count of acyclic configurations play an instrumental role.

Property3: Let G be a CA of size n. In the reachability tree of that CA, if k^i denote the number of non-reachable edge(s) at the level i of that tree, then the total number of non-reachable configurations(s) of that CA is $\sum_{i=0}^{n-1} k^i \times 2^{n-1-i}$.

Property4: Let G_1 and G_2 be irreversible cellular automata. They are said to be **isomorphic** if both the CAs have the same number of non-reachable configurations but the converse is not always true.

If **Property4** is not satisfied by the given G_1 and G_2, then they are not isomorphic. To check the count of non-reachable configurations, we should use the reachability tree as a tool.

Here, $(1, 135, 92, 5)$ and $(10, 60, 86, 20)$ are not isomorphic cellular automata as they have total number of non reachable configurations 3 and 4 respectively. Next, we need to deduce some more properties of CAs for determining isomorphism when they have sane number of non-reachable configurations.