



## Frontiers

# Characterisation of the elementary cellular automata in terms of their maximum sensitivity to all possible asynchronous updates



Eurico L.P. Ruivo<sup>a,\*</sup>, Marco Montalva-Medel<sup>b</sup>, Pedro P.B. de Oliveira<sup>a</sup>, Kévin Perrot<sup>c</sup>

<sup>a</sup> Universidade Presbiteriana Mackenzie, Faculdade de Computação e Informática, São Paulo, SP, Brazil

<sup>b</sup> Universidad Adolfo Ibáñez, Facultad de Ingeniería y Ciencias, Santiago, Chile

<sup>c</sup> Aix Marseille Univ., Univ. de Toulon, CNRS, LIS, UMR 7020, Marseille, France

## ARTICLE INFO

## Article history:

Received 20 December 2017

Revised 5 June 2018

Accepted 8 June 2018

## Keywords:

Cellular automaton

Asynchronous update

Update digraph

Discrete dynamics

One-step maximum sensitivity

## ABSTRACT

Cellular automata are fully-discrete dynamical systems with global behaviour depending upon their locally specified state transitions. They have been extensively studied as models of complex systems as well as objects of mathematical and computational interest. Classically, the local rule of a cellular automaton is iterated synchronously over the entire configuration. However, the question of how asynchronous updates change the behaviour of a cellular automaton has become a major issue in recent years. Here, we analyse the elementary cellular automata rule space in terms of how many different one-step trajectories a rule would entail when taking into account all possible deterministic ways of updating the rule, for one time step, over all possible initial configurations. More precisely, we provide a characterisation of the elementary cellular automata, by means of their one-step maximum sensitivity to all possible update schedules, that is, the property that any change in the update schedule causes the rule's one-step trajectories also to change after one iteration. Although the one-step maximum sensitivity does not imply that the remainder of the time-evolutions will be distinct, it is a necessary condition for that.

© 2018 Elsevier Ltd. All rights reserved.

## 1. Introduction

Cellular automata (CAs) are locally-defined dynamical systems, discrete with respect to time, space and state variables. They have been studied both from the point of view of their mathematical and computational properties as well as systems capable of simulating real-world phenomena [1], such as disease spread [2], urban growth [3] and fluid dynamics [4]. Even with simply defined local interaction rules, cellular automata may be capable of displaying arbitrarily complex global emergent behaviour; well-known binary cellular automata rules such as *The Game of Life* [5] and elementary cellular automaton *Rule 110*, are known to be capable of simulating a Turing machine [6].

Classically, the time-evolution of a configuration under a cellular automaton local rule is given by synchronously iterating such a rule over the entire configuration. However, the study of the effects of allowing asynchronous updates has been gaining attention in recent years. By taking into account additional update schemes further to the synchronous, the rules typically unveil a richer behavioural set, both in terms of their possible dynamics and capabilities of problem-solving as well as of simulating real-world phe-

nomena; see [7] for a comprehensive review on asynchronism and how it compares to the synchronous case.

From the viewpoint of simulating real-world complex systems, it is usually considered that they are more naturally simulated by asynchronous cellular automata rather than synchronous ones [8], since they generally result from the parts of the system interacting asynchronously based upon action and reaction.

On the other hand, from the computational point of view, some dynamical properties of certain cellular automata rules are partially due to the synchronism rather than due to only the rule itself [9,10], what leads to the natural question of which might be the new capabilities of a rule when allowing asynchronous update schedules [11].

In this context, a naturally relevant question is to understand how changes in the way a rule is updated over the set of configurations of a given length affects, both quantitatively and qualitatively, its dynamical behaviour. Aracena et al. [12] have constructively shown that it is possible to partition the set of possible updates into equivalence classes. The quantity of such classes may be significantly smaller than the number of individual updates in many situations, thus rendering comprehensive studies of all possible updates more tractable in these situations.

One particular way to study asynchronous binary cellular automata is to use the more general framework of *Boolean automata*

\* Corresponding author.

E-mail address: [eurico.ruivo@mackenzie.br](mailto:eurico.ruivo@mackenzie.br) (E.L.P. Ruivo).

networks (BANs), where the asynchronism is coded in the structure of labeled digraphs named *interaction digraphs*. Many dynamical properties, such as the quantification of fixed points [13,14] and the existence of limit-cycles [15], have been studied in such a context.

Here, we address the question of how changing the update schedule affects the number of different dynamics of a cellular automaton, after one iteration of the rule. More precisely, the term *dynamics* here refer to the set of pairs of configurations and their respective image under the CA rule under a given update schedule. In order to do so, we provide a way to compute how many independent update schedules do exist for configurations of a given length, establish the notion of one-step update schedule sensitivity and give necessary and sufficient conditions for an elementary cellular automaton to have maximum one-step sensitivity to changes in updates. We constrain to the well studied family of elementary cellular automata, where each cell can take on two possible states and the state transition of every cell of an automaton depends upon the states of the cell itself and of its immediate neighbours to the left and to the right. We understand that our short-range, one-iteration based study is a key step towards better understanding the longer-range dynamical behaviour of one-dimensional cellular automata, possibly even their limit behaviours.

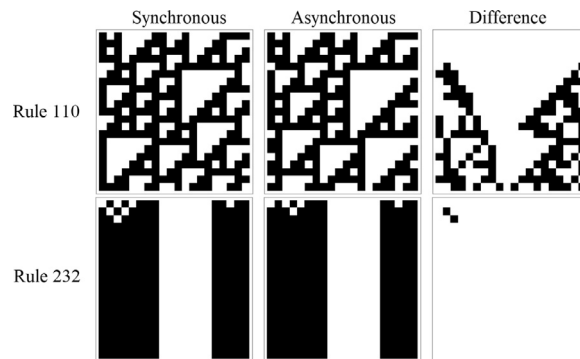
In order to know how much the dynamics of a rule varies for different update schedules, the notion of *update schedule sensitivity* is defined. We believe that such a measure may open new possibilities to study classical problems in the CA literature, previously addressed only in the synchronous update schedule; for instance, the *density classification task*, for which it has been shown that no single CA rule is able to solve it under synchronous update schedule (see [16] for a survey on this problem). The question of whether or not there might be a solution for some asynchronous update schedule is still an open question. Naturally, CAs with many different dynamics due to distinct update schedules, offers many more possibilities to tackle the problem.

Basically, two update schedules are said to be equivalent if for any CA rule and any configuration, the image of the configuration generated by iterating the CA rule under the first update schedule is the same as the one obtained by iterating the CA rule under the second update schedule. Also, a CA rule is said to have *maximum (one-step) update schedule sensitivity* when, for any two non-equivalent update schedules, there will be a configuration for which its image will be distinct for each update schedule. This will be addressed more formally in Section 4, but it is worth showing how this one-step property may affect the global behaviour of the rule along time.

First of all, it is worth noticing that even if the images of a configuration under distinct update schedules may differ in one iteration of the rule, it is possible that after a number iterations the trajectories become the same.

However, the one-step difference is a necessary condition in order for the trajectories of a configuration under distinct update schedules to be distinct, hence checking a rule for maximum one-step update schedule sensitivity is, in reality, telling whether or not a rule is capable of displaying distinct time-evolutions for small changes in a given update schedule.

In order to illustrate the above possibilities, Fig. 1 shows the difference of time-evolutions of ECA rules 110 (well-known for being Turing-universal for synchronous update) and 232 (the majority rule) for two almost equal update schedules: the synchronous one (left) and the one in which the leftmost cell is updated before all other cells (middle), followed by the difference of the time-evolutions (right). Therefore, as said above, maximum update schedule sensitivity is not sufficient, but it is necessary, in order for a rule to present long-term differences in the time-evolution in response for small changes in the update schedule.



**Fig. 1.** Time-evolution ECA rules 110 and 232 for the synchronous update schedule (left) and an asynchronous update schedule where the leftmost cell is updated before the other cells (middle), where time flows downwards and white and black cells represent cells in state 0 and 1, respectively. A one-step difference in the time-evolutions may (rule 110) or may not (rule 232) result in distinct trajectories after a given number of time-steps: the diagrams on the right represent the differences between each pair of time-evolutions (right).

The paper is organised as follows: the next section provides basic definitions regarding cellular automata, general update schedules and update digraphs. Section 3 summarises preliminary results regarding the maximum number of distinct updates schedules and of the distinct dynamics for configurations of a given length. As follows, Section 4 defines the notion one-step sensitivity to the update schedules, and Section 5 then gives a characterisation in terms of the local rules of the elementary cellular automata that display maximum one-step sensitivity to all possible update schedules. Finally, concluding remarks are made in Section 6.

## 2. Basic definitions

### 2.1. Cellular automata, configurations and dynamical equivalence

A *cellular automaton* (CA) is a quadruple  $(S, N, f, d)$ , where  $S = \{0, 1, \dots, k-1\}$  is the *state set*,  $N = (\bar{n}_1, \bar{n}_2, \dots, \bar{n}_m) \in (\mathbb{Z}^d)^m$  is the *neighbourhood vector*,  $f : S^m \rightarrow S$  is the *local function* (or *local rule*) and  $d \in \mathbb{Z}_+$  is the *dimension*. In particular, one-dimensional binary CAs have  $S = \{0, 1\}$ ,  $d = 1$  and, given  $r \in \{\frac{m}{2} : m \in \mathbb{Z}_+\}$ , the rule is said to be *radius- $r$*  when  $N = (-\lceil r \rceil, -\lceil r \rceil + 1, \dots, 0, \dots, \lceil r \rceil - 1, \lceil r \rceil)$ .

An *elementary cellular automaton* (ECA) is a radius-1, binary, one-dimensional cellular automaton. There are 256 ECAs arising from the distinct possible local rules  $f : \{0, 1\}^3 \rightarrow \{0, 1\}$ ; we refer to the set of all ECA rules by  $\mathcal{F}$ . Each ECA local rule  $f$  may be identified by its *Wolfram number* [17] given by

$$W(f) = \sum_{(q_1, q_2, q_3) \in \{0, 1\}^3} f(q_1, q_2, q_3) 2^{(2^2 q_1 + 2^1 q_2 + 2^0 q_3)}$$

A (periodic) *configuration of length  $L$*  is a function  $c : \mathbb{Z}_L \rightarrow \{0, 1\}$  where  $\mathbb{Z}_L$  denotes the set of integers modulo  $L$ . Each index  $i \in \mathbb{Z}_L$  of a configuration is named a *cell* and  $c(i)$ , denoted by  $c_i$  from now on, is the *state* of cell  $i$ . For the ECA rules, the vector  $(i-1, i, i+1)$  is the *neighbourhood* of cell  $i$  and cells  $i-1$ ,  $i$  and  $i+1$  are the *neighbours* of cell  $i$ . Here we denote the vector  $(c_{i-1}, c_i, c_{i+1})$  by  $[c_i]$ . We denote the set of all periodic configurations of length  $L$  by  $\mathcal{C}_L$  and a periodic configuration of length  $L$  simply by the vector  $(c_1, c_2, \dots, c_L)$  characterising  $c(1)$  to  $c(L)$ .

Given a local ECA rule  $f$ , it induces a *synchronous global function* (or *synchronous global rule*)  $F : \mathcal{C}_L \rightarrow \mathcal{C}_L$ , such that  $(F(c))_i = f(c_{i-1}, c_i, c_{i+1}) = f([c_i])$ . That is, the synchronous global rule results from applying the local rule  $f$  to the neighbourhood of each cell  $i$ , synchronously.

Download English Version:

<https://daneshyari.com/en/article/8253405>

Download Persian Version:

<https://daneshyari.com/article/8253405>

[Daneshyari.com](https://daneshyari.com)