Theoretical Computer Science 659 (2017) 36–52

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Leader election on two-dimensional periodic cellular automata

Nicolas Bacquey

GREYC – Université de Caen Basse-Normandie, ENSICAEN, CNRS, Campus Côte de Nacre, Boulevard du Maréchal Juin CS 14032 Caen cedex 5, France

ARTICLE INFO

Article history:
Received 6 March 2015
Received in revised form 19 April 2016
Accepted 29 October 2016
Available online 22 November 2016
Communicated by N. Ollinger

Keywords:
Cellular automata
Leader election
Bi-periodical configuration
Equivalence classes
Uniform computation

ABSTRACT

This article explores the computational power of bi-dimensional cellular automata acting on periodical configurations. It extends in some sense the results of a similar paper dedicated to the one-dimensional case. More precisely, we present an algorithm that computes a "minimal pattern network", *i.e.* a minimal pattern and the two translation vectors it can use to tile the entire configuration. This problem is equivalent to the computation of a leader, which is one equivalence class of the cells of the periodical configuration.

0. Introduction

Cellular automata are a well-studied computational model. Its uniform and local properties capture a large range of natural problems, and the model is simple enough to allow definitions of algorithms, or complexity classes. However, the infinite nature of its underlying structure may raise some issues when confronted with finite objects. There are two naive ways of solving these issues: either working on a finite subset of the automaton, or requiring the whole infinite configuration to be periodical.

Periodical configuration is quite a natural concept in the context of tiling problems. As a significant example, it was first (erroneously) conjectured by Hao Wang in [13] that each finite set of tiles that tiles the plane can always do it by some periodical configuration. Periodical configurations have also been extensively studied on cellular automata, when those are considered as dynamical systems: typically, questions of undecidability of the injectivity or reversibility of the transition function have been studied [5]. On the other hand, periodical configurations are much less studied from the point of view of computation, with some notable exceptions, such as the density classification problem [7].

A natural problem would be to compute a "minimal period" of a periodical configuration. However, performing computations on periodical configurations is somehow counterintuitive, because one cannot easily define essential notions, such as the origin and termination of the computation, or the time complexity of an algorithm. These difficulties are mainly due to the fact that unlike what happens on classical models such as Turing machines, you cannot choose a single cell to start the computation or bear its result. This difficulty is overridden when the input of a cellular automaton is bounded by persistent symbols, because then you can identify cells that are on the border of the computation area, and use them as starting or

E-mail address: nicolas.bacquey@unicaen.fr.

http://dx.doi.org/10.1016/j.tcs.2016.10.021

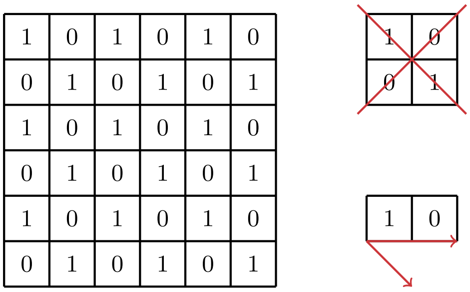

Fig. 1. One cannot extract the 2×2 square pattern from the configuration, because all cells marked with 1 are *indistinguishable* from each other. Nevertheless, it is possible to extract the 2×1 pattern, with its two tiling vectors.

stopping points. However, one cannot use such tricks when the configuration is periodical; in that case, all notions related to computation should be global.

This paper presents an extension of a previous work [1] dedicated to the simpler one-dimensional case. In [1] we exhibited a one-dimensional cellular automaton that computes in polynomial time a minimal period of an infinite one-dimensional periodic configuration. We now want to deal with bi-periodical configurations of dimension 2, *i.e.* configurations that have two independent vectors of periodicity. A natural starting point would be to compute the *minimal pattern* of a given configuration, *i.e.* the smallest pattern with which we are able to rebuild the whole configuration by translation along the two orthogonal axis. This problem is unsolvable in the case of cellular automata of dimension ≥ 2, as it is briefly suggested in Fig. 1. Instead, we will solve the problem of exhibiting a *minimal pattern network*, *i.e.* a minimal pattern and the two translation vectors it can use to tile the entire configuration.

We can see that this minimal pattern network can be computed through the election of a leading equivalence class of cells. This leader election will be the main subject of this article. While leader election on two dimensional cellular automata has already been extensively studied [11], all the algorithms that already exist cannot be applied to our model, because they heavily rely on the existence of computational borders, that frame the finite computational area. Due to the periodical nature of the configurations we study, such borders cannot exist in our framework. In that sense, this article presents a new algorithm that performs leader election on a broader class of configurations.

After having clearly defined the problem of leader election on periodical cellular automata, we will present some algorithmic tools that fit our needs. Finally, we will present an algorithm that performs leader election in polynomial time.

1. Context and basic definitions

1.1. The computational model

We will use along this article the standard definition of cellular automata (CA) as a tuple $\mathcal{A} = (d, \mathbf{Q}, V, \delta)$ (see [6]). In these lines, we will work with $d = 2$, *i.e.* with cellular automata whose underlying network is $\mathbb{Z}^2$. $\mathbf{Q}$ denotes the set of *states*, and V is the standard *Moore neighbourhood*.[1] The local transition function of the automaton is denoted by $\delta : \mathbf{Q}^V \to \mathbf{Q}$. As we work with cellular automata from the point of view of *language recognition*, we will identify a particular subset $\Sigma \subseteq \mathbf{Q}$ as the *input alphabet*. A *configuration* is an application $C : \mathbb{Z}^2 \to \mathbf{Q}$. We also introduce the *global transition function* $F_\delta : \mathbf{Q}^{\mathbb{Z}^2} \to \mathbf{Q}^{\mathbb{Z}^2}$ defined by the global synchronous application of δ over configurations of $\mathbb{Z}^2$.

We suppose that the reader is familiar with the notions of signals and computation layers on cellular automata. If it is not the case, we strongly encourage the reading of [6] or [9] for such general matters on cellular automata.

Definition 1. We define a *Toric-Cellular Automaton (toric-CA)* as a cellular automaton whose initial configuration (and therefore any subsequent configuration) is bi-periodic (*i.e.* periodic in two independent directions).

Note that this model is equivalent to an automaton that would work on a finite, torus-like cell network.

Definition 2. Let w be a rectangular word over Σ, we denote $C_w \in \mathbf{Q}^{\mathbb{Z}}$ the configuration formed by the uniform repetition of w over $\mathbb{Z}^2$.

Definition 3. We call a *cell* an element of the underlying network $\mathbb{Z}^2$ and the state of $\mathbf{Q}$ associated with it.

Note that the state of a given cell may change through time.

[1] $V = \{(-1, -1), (-1, 0), (-1, 1), (0, -1), (0, 0), (0, 1), (1, -1), (1, 0), (1, 1)\}$.