Journal of Computational Science 11 (2015) 245–257

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

On the decomposition of stochastic cellular automata

Witold Bołt[a,b,*], Jan M. Baetens[b], Bernard De Baets[b]

[a] Systems Research Institute, Polish Academy of Sciences, Newelska St. 6, 01-447 Warsaw, Poland
[b] KERMIT, Department of Mathematical Modelling, Statistics and Bioinformatics, Ghent University, Coupure Links 653, B-9000 Gent, Belgium

ARTICLE INFO

Article history:
Received 16 February 2015
Received in revised form 18 August 2015
Accepted 9 September 2015
Available online 24 September 2015

Keywords:
Stochastic cellular automata
Complexity analysis
Continuous cellular automata
Decomposition

ABSTRACT

In this paper we present two interesting properties of stochastic cellular automata that can be helpful in analyzing the dynamical behavior of such automata. The first property allows for calculating cell-wise probability distributions over the state set of a stochastic cellular automaton, *i.e.* images that show the average state of each cell during the evolution of the stochastic cellular automaton. The second property shows that stochastic cellular automata are equivalent to so-called stochastic mixtures of deterministic cellular automata. Based on this property, any stochastic cellular automaton can be decomposed into a set of deterministic cellular automata, each of which contributes to the behavior of the stochastic cellular automaton.

1. Introduction

Cellular automata (CAs) are often used for constructing models in a variety of fields of application, including chemistry, biology, medicine, physics, ecology and the study of socioeconomic interactions. In many of these settings, stochastic CAs (SCAs) are considered due to the stochastic nature of the phenomenon under study or due to a lack of understanding of the exact rules driving the phenomenon [1,2]. Therefore, a better understanding of the dynamics of SCAs is crucial. Only a few methods for dealing with the analysis of models based on SCAs have been developed. In many practical applications, especially in cases where the averaged behavior of the system is of concern, sampling methods, relying on extensive computer simulations, are sufficient [3–5]. Techniques built on the mean-field theory can be used to study the long-term behavior of SCAs [6]. The theory of Markov chains can be applied to provide analytical tools for analyzing the model's behavior [7], although in practice, due to the theoretical and computational complexity of such tools, the application scope is limited.

This paper is devoted to providing effective analytical tools based on deterministic CAs for the analysis of multi-state SCAs. Although the theoretical foundations of the presented results are already (at least partially) known in the literature [7–9], so far the applications are limited. Therefore, the main aim of this paper is to provide a complete, formal description of the discussed properties in a form that is suitable for applications and that does not require a strong mathematical background, as well as to present examples that can motivate further applications in the domain of systems modeling. The methods presented here are developed in the context of 1D SCAs on finite lattices, but can be easily generalized to the case of higher dimensions and infinite spaces.

Two main results are presented in this paper. The first one involves constructing images that show the cell-wise probability distribution over the state set, at any time step. The method is based on associating an SCA with a deterministic, continuous CA (CCA). The second result shows the equivalence of SCAs and stochastic mixtures of deterministic CAs. Based on this finding, any SCA can be decomposed into a finite set of deterministic CAs, each of them contributing to the behavior of the stochastic system. An effective method for finding a decomposition is presented. It allows to uncover the deterministic component in the mixture with the highest impact on the behavior of the SCA.

This paper is organized as follows. We start with some preliminaries and definitions in Section 2. In Section 3, we introduce the concept of CCAs and the formalism enabling the analysis of multi-state CAs. Section 4 contains the definition of multi-state SCAs and holds the main results of this paper. The paper is concluded with Section 5, discussing the experimental results that illustrate our results. A summary is presented in Section 6.

2. Preliminaries

Informally, a CA is a discrete dynamical system in which the space is subdivided into discrete elements, referred to as cells. At

* Corresponding author at: Systems Research Institute, Polish Academy of Sciences, Newelska St. 6, 01-447 Warsaw, Poland.
E-mail address: witold.bolt@hope.art.pl (W. Bołt).

http://dx.doi.org/10.1016/j.jocs.2015.09.004
1877-7503/

every consecutive, discrete time step, each cell is assigned one of N states using a deterministic rule, which depends only on the previous state of the considered cell and the states of its neighboring cells [10].

Formally speaking, let the state set $\mathcal{S}$ be a finite set of $N>1$ elements. Elements of the set $\mathcal{C}=\{c_i \mid i=1,\ldots,M\}$ will be referred to as cells. Every cell c_i is assigned a state $s(c_i,t)\in\mathcal{S}$ at each time step $t\in\mathbb{N}$, according to a local, deterministic rule. The vector $s(\cdot,t)\in\mathcal{S}^M$ will be referred to as the configuration, and as the initial configuration when $t=0$. The sequence $(s(\cdot,0), s(\cdot,1),\ldots)$ will be referred to as the space-time diagram of the CA. For technical reasons, we impose periodic boundary conditions, but our results do not depend on this assumption.

The function $A:\mathcal{S}^M\to\mathcal{S}^M$ satisfying, for every $t\in\mathbb{N}$:

$$s(\cdot,t+1)=A(s(\cdot,t)), \tag{1}$$

will be referred to as a global CA rule or simply a CA, if there exists a radius $r\in\mathbb{N}$ and a function $f:\mathcal{S}^{2r+1}\to\mathcal{S}$ satisfying:

$$s(c_i,t+1)=f(s(c_{i-r},t),\ldots,s(c_{i+r},t)), \tag{2}$$

for every i and $t\in\mathbb{N}$. Such a function f will be referred to as a local rule. Note that a local rule uniquely defines the global rule, while for a given global rule, multiple local rules exist. Additionally, it is assumed that $r\ll M$. The vector $(c_{i-r}, c_{i-r+1}, \ldots, c_{i+r-1}, c_{i+r})$ will be referred to as the neighborhood of cell c_i and $R=2r+1$ will denote the neighborhood size. For the sake of simplicity, $s(c_{i-r},\ldots,c_{i+r},t)\in\mathcal{S}^R$ will denote the state of the neighborhood of cell c_i at time step t, and will be referred to as the neighborhood configuration of c_i at time step t.

3. Continuous CAs

There exist multiple ways of extending the definition of CAs to cover infinite state sets. Examples of such approaches include coupled map lattices (CMLs) [11] and so-called fuzzy CAs [12–14]. In this section, we present continuous CAs (CCAs), which can be seen as a generalization of the ideas presented in [12]. Our formalism is based on a polynomial representation of discrete CA rules. We start with formulating the continuous counterparts of binary CAs. After that we present a generalization to cover multi-state CAs.

3.1. Binary CAs

Binary CAs are widely studied [15,16], because they allow to evolve complex patterns and exhibit complex behavior despite their intrinsic simplicity. The state set of such a CA A is $\mathcal{S}=\{0,1\}$. We will now formally define and characterize its local rule $f:\mathcal{S}^R\to\mathcal{S}$. Let $l=(l_i)_{i=1}^{2^R}$ be a binary vector. We consider a system of equations:

$$f(s_{i,1},\ldots,s_{i,R})=l_i, \tag{3}$$

where $(s_{i,1},\ldots,s_{i,R})$ is a binary vector such that $i=1+\sum_{j=0}^{R-1}s_{i,R-j}\,2^j$. As can be seen, such a system of equations is uniquely defined by the vector l. The vector l will be referred to as the lookup table (LUT) of the local rule f. It is not difficult to check that such a system uniquely defines the function f, since it lists all of the possible input configurations, and maps them to corresponding values by components of the vector l.

Following [12], we know that the function f can be expressed as a polynomial, which is of interest for our purposes. In order to define it, we introduce two auxiliary functions. We start with the function $\mathrm{ind}:\{1,\ldots,2^R\}\to\{1,2\}^R$. It is defined in such a way that $\mathrm{ind}(i)[m]$ is the m-th digit, incremented by one, read from left to right, of the binary representation of the integer $i-1$, padded with ones on the left, so that it always has length R. Consequently, it holds that:

$$i=1+\sum_{m=1}^{R}(\mathrm{ind}(i)[R-(m-1)]-1)\,2^{m-1}. \tag{4}$$

The values of $\mathrm{ind}(i)$ for $R=3$ and $i\in\{1,\ldots,8\}$ are shown below:

$$\mathrm{ind}(1)=(1,1,1),\quad \mathrm{ind}(2)=(1,1,2),\quad \mathrm{ind}(3)=(1,2,1),$$
$$\mathrm{ind}(4)=(1,2,2),\quad \mathrm{ind}(5)=(2,1,1),\quad \mathrm{ind}(6)=(2,1,2),$$
$$\mathrm{ind}(7)=(2,2,1),\quad \mathrm{ind}(8)=(2,2,2).$$

The function ind is related to the binary representation of integers. In [12] a simpler formulation using the function bin, which yields the binary representation of an integer, is used. The construction presented here, although a bit more complicated in the binary setting, allows for a smoother generalization to the multi-state case.

Using the function ind, we now define the function $n:\mathcal{S}\times\mathbb{N}\times\mathbb{N}\to\mathcal{S}$, which for $s\in\mathcal{S}$ and $m,i\in\mathbb{N}$, is given by:

$$n(s,m,i)=\begin{cases}s, & \text{if } \mathrm{ind}(i)[m]=2,\\ 1-s, & \text{if } \mathrm{ind}(i)[m]=1.\end{cases}$$

Note that we will use vectors of states of the form $(s_1,\ldots,s_R)\in\mathcal{S}^R$ and for simplicity, for any $m\in\{1,\ldots,R\}$, we will write $n(s_m,i)$ instead of $n(s_m,m,i)$. Using the functions ind and n, we can write the polynomial representation of the local rule f as:

$$f(s_1,\ldots,s_R)=\sum_{i=1}^{2^R}l_i\left(\prod_{m=1}^{R}n(s_m,i)\right). \tag{5}$$

The following example shows the explicit form of this polynomial for a member of the family of elementary CAs (ECAs).

Example 1 *(Elementary CAs).* Binary, 1D CAs with neighborhood radius $r=1$ are commonly referred to as ECAs [15]. There are 256 such ECAs. Treating the LUT entries l_i as digits of an integer written in base 2, we can enumerate the local rules of ECAs. By convention, the binary vectors are read in the reverse order, *i.e.* $(l_i)_{i=1}^{8}$ is interpreted as $(l_8,l_7,\ldots,l_1)_2$. For example, given the LUT $l=(0,1,1,0,1,0,0,1)$ of ECA 150, and denoting the Boolean complement as $\bar{s}=1-s$, its local rule can be written, according to Eq. (5), as:

$$f_{150}(s_1,s_2,s_3)=\bar{s}_1\,\bar{s}_2\,s_3+\bar{s}_1\,s_2\,\bar{s}_3+s_1\,\bar{s}_2\,\bar{s}_3+s_1\,s_2\,s_3. \quad \square$$

Using the above notation, a CCA can be defined analogously to a binary CA, with two notable differences. Firstly, the state set of a CCA is the unit interval, *i.e.* $\mathcal{S}=[0,1]$, and, secondly, the local rule $f:[0,1]^R\to[0,1]$ is given by Eq. (5) with coefficients $l_i\in[0,1]$. We will refer to such a vector $(l_i)_{i=1}^{2^R}$ as a generalized LUT. It is easy to check that this definition of a CCA is formally correct. Indeed, the values of the function f in Eq. (5) are guaranteed to belong to the unit interval if $l_i\in[0,1]$ for all i and $s_m\in[0,1]$ for all $m\in\{1,\ldots,R\}$. Note that this construction is directly related to the one presented in [12], where fuzzy CAs are constructed as polynomials representing fuzzified logical functions. Following the same line of reasoning, an alternative polynomial representation for the local rules of binary CAs is presented in [17], which is consistent with a logical representation of the local rules.

In order to introduce the formalism that is needed in the multi-state setting, we present a slightly modified way of representing binary CAs compared to the one obtained through Eq. (5). Let us assume that the state set is given by $\mathcal{S}_2=\{(1,0),(0,1)\}\subset\mathbb{R}^2$. Then the local rule is a function $f:\mathcal{S}_2^R\to\mathcal{S}_2$ and can be represented as a vector function $f=(f_1,f_2)$, where $f_j:\mathcal{S}_2^R\to\{0,1\}$. Note that any $s=(s_1,s_2)\in\mathcal{S}_2$ satisfies $s_2=1-s_1$. Similarly, $f_2=1-f_1$, so that the local