Theoretical Computer Science 259 (2001) 271–285

Theoretical Computer Science

www.elsevier.com/locate/tcs

Kolmogorov complexity and cellular automata classification

J.-C. Dubacq, B. Durand, E. Formenti *

Lab. de Inform. du Parallelisme, Ecole Normale Sup. de Lyon, 46 Alleé d'Italie,
F-69364 Lyon Cedex 07, France

Received January 1998; revised March 1999
Communicated by Nivat

Abstract

We present a new approach to cellular automata (CA) classification based on algorithmic complexity. We construct a parameter κ which is based only on the transition table of CA and measures the "randomness" of evolutions; κ is better, in a certain sense, than any other parameter recursively definable on CA tables. We investigate the relations between the classical topological approach and ours. Our parameter is compared with Langton's λ parameter: κ turns out to be theoretically better and also agrees with some practical evidences reported in literature. Finally, we propose a protocol to approximate κ and make experiments on CA dynamical behavior. © 2001 Published by Elsevier Science B.V.

1. Introduction

Cellular automata (CA) are often used for modeling systems consisting of many elementary cells interacting locally with each other. The memory of cells is finite, their interactions are synchronous and occur at discrete time steps.

Notwithstanding the apparent simplicity of the formal definition of CA, they display a wide range of interesting dynamical behaviors. And in fact, the problem of their classification is a central topic in CA theory. In [23], Wolfram heuristically observes the following behaviors:

W_1: evolution to a homogeneous state;
W_2: evolution to a set of space–time patterns which are stable or periodic;
W_3: evolution to an "aperiodic" or "chaotic" space–time pattern;
W_4: evolution to complex localized structures, sometimes long-lived.

It is clear that this classification is neither complete nor well-formalized. In fact, many successive works on CA were an attempt to give it a formal consistency [8, 4].

* Corresponding author.
E-mail addresses: jcdubacq@ens-lyon.fr (J.-C. Dubacq), bdurand@cmi.univ-mrs.fr (B. Durand), enrico.formenti@cmi.univ-mrs.fr (E. Formenti).

PII: S0304-3975(00)00012-8

Successively, some purely topological classifications have been proposed. They are based, for instance, on the structure of attractors [16, 13] or on the equicontinuity property [16].

If the predicate "to have a classification" means to have an algorithm which, given a CA as input, decides to which class this CA belongs to, then any of the above classifications is undecidable [14]. Moreover, these classifications take into account neither the "information" content nor the algorithmic complexity of the evolutions. We propose an alternative approach which is supposed to fill this gap.

The inspiring principle of the classification of Cattaneo et al. [4] is to study how the information contained in the local rule influences the global behavior of the CA. We have the same goal but we use an algorithmic approach rather than a topological one. We introduce a parameter κ which measures the information content of local rules. Its definition is essentially based on concepts taken from the well-established Kolmogorov complexity theory which allows us to prove an interesting optimality result: κ is "better" than any other computable parameter defined on CA local rules (see Theorem 2).

The algorithmic approach is useful in a number of practical considerations. For instance, if one wants to understand which is the better compromise between the size of the table and the size of the observation domain in simulations. An even more concrete example: suppose to simulate coffee percolation using CA. In this case only a small subset of all configurations represents a porous medium. This situation can be studied using our approach, the only requirement is that the set of admissible configurations has to be recursive. We underline that the approach is also easily extensible to arbitrary dimension at the cost of small changes in the formalization.

The idea of using a parameter based on CA tables for classifying CA behavior is not new. As far as we know it was first issued by Langton in [17]. Section 4 discusses Langton's approach; we prove that it is too rough to discriminate chaotic behavior from simple periodic behavior. The discussion is further developed in Section 5.6, where our approach justifies some critics on Langton's parameter coming from experimental studies that recently appeared in literature [7].

One of the most studied dynamical behavior is chaoticity. Here we prove that topological chaos implies non randomness of CA tables (Corollary 1). This fact underlines one of the differences between randomness and topological chaos.

Our study is theoretical. In order to give qualitative and quantitative evaluations of CA evolutions, an experimental protocol which uses the parameter κ is proposed in Section 6. This protocol may be effectively exploited by people who intend to simulate physical systems using CA; its main advantage in this context is that it is based on a well-established theory: the general theory of algorithms.

2. Dynamical systems

A *dynamical system* is a continuous function $f:X\to X$ from a nonempty metric space X to itself. The set X is called the *phase space* of f. For $n\in\mathbb{N}$, the nth iterate

$f^n: X \to X$ is defined by $f^0(x) = x$ and $f^{n+1}(x) = f(f^n(x))$. A point x is *periodic* if there exists $n > 0$ such that $f^n(x) = x$; the least positive integer with this property is called the *period* of x. An *orbit* of initial state x_0, denoted by $O(x_0)$ is the set $O(x_0) = \{x \in X \mid f^n(x_0) = x\}$.

The main goal in the study of dynamical systems is to understand their long-term behavior, that is to understand the structure of their orbits. One of the most appealing dynamical behavior is the so-called *chaotic behavior* or chaos. Even if there is no universally accepted mathematical definition of deterministic chaos, many properties are recognized as possible indicators of chaotic behavior such as: sensitivity to initial conditions, denseness of periodic orbits, transitivity, Lyapunov exponents. For a survey see [9, 1, 15, 3, 6]. In this paper we adopt Devaney's definition of chaos which says that a dynamical system is chaotic if and only if it is transitive and its set of periodic points is dense in the phase space [9, 1]. A dynamical system with a dense set of periodic points is called *regular*.

A system (X, f) is (*topologically*) *transitive* if for any nonempty open sets $U, V \subseteq X$, there exists $n > 0$ such that $f^n(U) \cap V \neq \emptyset$. Intuitively, a system is transitive if for any two points x, y, we can find in any arbitrary small neighborhood of x a point whose orbit reaches any arbitrary small neighborhood of y. In particular, transitivity implies that the system can not be split into two independent subsystems. A system (X, f) is *surjective* if f is surjective.

Remark 1. It is not difficult to see that surjectivity is a necessary property for a system to be regular or transitive and hence for being Devaney-chaotic. This is the seminal idea used for the proof that no chaotic CA has a random table (see Theorem 4).

3. Cellular automata

Formally, a one-dimensional CA (1-D CA) is a triple $\langle \mathbb{S}, N, f \rangle$, where $\mathbb{S} = \{0, 1, \ldots, S - 1\}$ is the set of *states*, $N = \{-r, \ldots, 0, \ldots, r\}$ is the *neighborhood* structure with *radius r*, and $f : \mathbb{S}^{2r+1} \to \mathbb{S}$ is the *local rule*. A state $s \in \mathbb{S}$ is *quiescent* if $f(s \ldots s) = s$. A configuration is a "snapshot" of the state of cells, i.e. a mapping from $\mathbb{Z}$ to $\mathbb{S}$. Denote by $\mathbb{S}^{\mathbb{Z}}$ the set of all configurations. The evolution of the system from time t to time $t + 1$ is given by the *global function* induced by f:

$$\forall c \in \mathbb{S}^{\mathbb{Z}}, \quad F_f(c^{t+1})_i = f(c^t_{i-r}, \ldots, c^t_i, \ldots, c^t_{i+r}).$$

A CA is called surjective if its global function is surjective.

If $\mathbb{S}^{\mathbb{Z}}$ is endowed with the product topology induced by the discrete topology on S, then $\mathbb{S}^{\mathbb{Z}}$ is a Cantor space (i.e. compact, perfect and totally disconnected space). It is easy to see that the following metric induces the product topology on $\mathbb{S}^{\mathbb{Z}}$:

$$\forall x, \ y \in \mathbb{S}^{\mathbb{Z}}, \quad d(x, y) = \sum_{i=-\infty}^{\infty} \frac{\delta(x_i, y_i)}{2^{|i|}},$$

where $\delta(a,b)=1$ if $a\neq b$, 0 otherwise. CA can be seen as dynamical systems on $\mathbb{S}^{\mathbb{Z}}$ since their global function is continuous with respect to the product topology.

A space–time diagram of initial state $c\in\mathbb{S}^{\mathbb{Z}}$ is a graphical representation of an orbit of initial state c:

$$\left|\begin{array}{c|ccccccc|l}
t=0 & \dots & c(-2) & c(-1) & c(0) & c(1) & c(2) & \dots & =c \\
t=1 & \dots & c^1(-2) & c^1(-1) & c^1(0) & c^1(1) & c^1(2) & \dots & =F_f(c) \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
t=k & \dots & c^k(-2) & c^k(-1) & c^k(0) & c^k(1) & c^k(2) & \dots & =F_f^k(c) \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots
\end{array}\right|.$$

The study of space–time diagrams give some hints on the global qualitative behavior of the CA. Most of the early works on CA classification follow this idea [23].

In Remark 1 we saw that surjectivity is necessary for Devaney chaos. Therefore, when comparing chaotic CA and random evolutions we are concerned with a subclass of surjective CA. In order to prove our results we need to reformulate surjectivity as a property on the local rule: balance. A CA with radius r is k-balanced if

$$\forall y\in\mathbb{S}^{2r(k-1)+1},\quad |\{x\in\mathbb{S}^{2rk+1}\mid f(x)=y\}|=S^{2r}.$$

The following proposition states the equivalence between balance and surjectivity.

Proposition 1 (Hedlund [12]). *A 1-D CA is surjective iff it is k-balanced for all $k\in\mathbb{N}$.*

4. Chaos has no edges

Classifying dynamical behavior using a parameter means to have a one-to-one correspondence between intervals and types of behavior. In practical situations one may also require the parameter to be effective, i.e. computable. For example, in the context of CA, effectiveness may be easily achieved if the parameter is defined on the finite objects characterizing the CA itself: the set of states and the local rule.

The first approach of this kind (as far as we know) is due to Langton [17]. Given a CA $\langle S,r,f\rangle$, let s be a quiescent state and let n be the number of tuples for which f outputs s. *Langton's parameter* λ_s for f is $\lambda_s(f)=1-n/S^{2r+1}$ (where S denotes the cardinality of $\mathbb{S}$). Note that the value of $\lambda_s(f)$ is strongly dependent on the choice of s and that not all CA have quiescent states.

Analyzing data from CA evolutions, Langton remarks that there exist "critical values" of λ_s in which chaotic behavior is to be found. Therefore he stated the popular hypothesis of the *edge of chaos* (EOC) [11]:

> In its basic form this is the hypothesis that in the space of dynamical systems of a given type, there will generically exist regions in which systems with simple

> behavior are likely to be found, and other regions in which systems with chaotic behavior are to be found. Near the boundaries of these regions more interesting behavior, neither simple nor chaotic, may be expected.

EOC hypothesis is very appealing for scientists but it has been criticized by many researchers (for a survey see [7]). Moreover λ_s does not discriminate tightly CA dynamics. In fact, take the set E of all CA with local rule f such that $\lambda_s(f)=\frac{1}{2}$. In E one can find a great variety of dynamical behaviors ranging from fixed-point behavior to chaotic one.

5. The algorithmic complexity approach

Let us start with a practical example. Consider Conway's *Game of Life* [2, 10].

```
sum = 0
For i in $[-1,1]\times[-1,1]$
  if neighbor $i$ is alive then add 1 to sum
New_state = dead
if sum = 3 then New_state = alive
if sum = 4 then New_state = Old_state
```

What have we done above? We gave an algorithm which describes how to compute the table rather than give the whole table which consists of 2^9 entries. Can we do better? That is to say, are there shorter algorithms which compute the same table? To answer these questions, we address Kolmogorov complexity, which is concerned with information from an algorithmic point of view. [1]

In the sequel, for the sake of simplicity, we make the following conventions. CA tables are seen as words of length S^{2r+1} over the alphabet $\mathbb{S}$, which are the images of all blocks of size $2r+1$ over the alphabet $\mathbb{S}$ ordered lexicographically. A table completely describes a CA: assuming $\mathbb{S}$ is known, there is no ambiguity in identifying a CA with its table.

5.1. Some basic results on Kolmogorov complexity

Kolmogorov complexity [18], also known as *algorithmic information theory* [5], studies the shortest description of words. Some words can be described by very short programs, while other offer no regularities and need to be fully spelled.

Let us recall the basic definition of Kolmogorov complexity. For any words x and y on a finite alphabet, the Kolmogorov complexity of x given y, according to a machine φ, is $\mathbf{K}_\varphi(x|y)=\min\{l(p),\varphi\langle p,y\rangle=x\}$ where $l(p)$ denotes the length of the binary

[1] For instance, it is better to consider not this program, but its compressed form, which will be really shorter than all 2^9 transitions.

word p. In particular, when y is the empty word, we can drop the "given y" part and just write $\mathbf{K}_{\varphi}(x)$. More intuitively the Kolmogorov complexity of x knowing y according to a machine φ is the size of the smallest program that produces x when applied on y via the machine φ.

A fundamental result of Kolmogorov complexity theory is that there exists a specific Turing machine that yields optimal results (Kolmogorov–Solomonov theorem). More precisely, there exists a machine φ_0, called *additively optimal* such that $\forall\varphi'$, $\exists c_{\varphi'}\in\mathbb{N}$, $\forall x, y, \mathbf{K}_{\varphi_0}(x|y)\leqslant\mathbf{K}_{\varphi'}(x|y)+c_{\varphi'}$. From now on, we fix any additively optimal φ_0 and drop the subscript φ_0 from $\mathbf{K}_{\varphi_0}(x|y)$. It is important to note that $\mathbf{K}(x|y)$ is approximable from above, but *not* computable.

There exist many variants of Kolmogorov complexity, for a survey, see [22]. In our case, any version of Kolmogorov complexity can be used. *Prefix* Kolmogorov complexity (also called self-delimiting) can be used as well as the original version and gives the same results. This variety is essential when defining randomness with infinite words. This is not our case: we are concerned with tables which are finite objects; thus, we can use the prefix variety or even the monotonic one without significant change in the results (see [22]).

5.2. Classification parameters

The goal of this approach is to define a family of *classification parameters* for cellular automata. Reasonable ones should fulfill some conditions: quantify regularities in tables, being effectively constructible, satisfy some natural normalization properties. A classification parameter is a function that assigns to every CA a value. We introduce a family of classification parameters on which we establish general laws.

A classification parameter π is a function taking a CA table as an input, and giving as output a mark that quantifies the degree of complexity of the table. If a table has many regularities, then it should get a low mark and conversely. The main idea behind all the following definitions comes from the theory of Martin-Löf tests for randomness [19, 20].

As for effectiveness we do not ask the parameters to be recursive but we require them to be at least recursively enumerable from above, that is to say, the set $\{\langle x, p, q\rangle,\ \pi(x)\leqslant p/q\}$ is recursively enumerable. Without any loss of generality, we restrict the output values to rational numbers in the interval $[0, 1]$.

Normalization is achieved by quantifying the maximum number of tables that can get a low mark. This quantification should take into account the maximum number of tables, hence the cardinality of $\mathbb{S}$ and the radius r. A uniform normalization is not sufficient because two different parameters, with the same intuitive meaning, could have very different histograms. More formally, we add the following condition:

$$\forall n,\ \forall m\in\mathbb{N}\sum_{\substack{\pi(x)\leqslant m/n\\ l(x)=n}} S^{-n}\leqslant S^{m-n} \tag{1}$$

or, equivalently

$$\forall n,\ \ \forall m \in \mathbb{N}\ \mathrm{Card}|x|\pi(x) \leqslant m/n, l(x) = n| \leqslant S^m.$$

This class of parameters is quite general. Take an example with $\mathbb{S} = \{0, 1\}$. With small changes, Langton's parameter λ can be turned into an algorithmic classification parameter π_λ that fulfills the above requirements. Let $\pi_\lambda = 1 - |2\lambda - 1|$. It is maximal when the number of 0's and 1's are equal, and minimal for a table which is only 0's or only 1's. In Section 5.5 we will see that π_λ is still too rough.

Theorem 1 shows that the set of classification parameters has some good recursivity properties. These properties will be very useful to build a "maximal" (in a sense which will be specified later) classification parameter.

Theorem 1. *There exists an effective enumeration of classification parameters.*

Each integer can be associated with a machine that "computes" (by successive approximations) a parameter, and all parameters can be represented by at least one integer. Remark that if we restrict the parameters to be total recursive, it is not possible to do such an enumeration (total recursive functions cannot be enumerated). The proof of this result is inspired by the enumeration of Martin-Löf tests.

Proof. To describe a parameter π, we have to give an algorithm which, given an *identifier* of the parameter and some table x, enumerates all possible rational values that are greater than the actual value of $\pi(x)$. Our claim is that a number $n \in \mathbb{N}$ is sufficient for the identifier and that each parameter is represented by at least one integer.

In the first part of the proof (step 1), we will turn partial recursive functions of $\mathbb{N} \to \mathbb{N}^3$ into an effective enumeration of partial recursive functions with the property that if the computation finishes with entry n, then it finishes for entry $n - 1$ too. In the second part, we turn this enumeration into an enumeration of classification parameters. The description is the enumeration of the set $\{\langle x, p, q\rangle,\ \pi(x) \leqslant p/q\}$. The proof is straightforward at this point, because the second part of the transformation (step 2) leaves unchanged the functions that already co-enumerates sets corresponding to classification parameters. We are guaranteed that after step 2, only classification parameters still remain.

1. We want to turn an enumeration of partial recursive functions into an enumeration of all partial recursive functions defined on initial segments $[0, n]$. We dovetail the computations on the input partial recursive function f so that at step $n*(n-1)+m+1$ we simulate n steps of the computation of f with input m. The first convergent computation shall be defined as $g(0)$, the second as $g(1)$, and so on. Remark that the range of g is equal to the range of f.
2. This part of the procedure deletes those partial recursive functions which are not classification parameters and leaves unchanged the others. For any g (obtained in the previous step) the algorithm increasingly computes the required set, and at each time when a new value is added, it checks whether condition 1 holds. This can

be done since, in step 1, we have shown that at any time the number of values for which an upper bound has been given (at least one triple $\langle x, p, q\rangle$ has been enumerated) is always finite.

(a) At any time, if $\pi(x)$ has never been given an upper bound, set the upper bound to 1. Take a variable i that will count all the triples produced by the function g.

(b) Increase i and compute $g(i-1)$. We obtain either a triple $\langle x_i, p_i, q_i\rangle$ or nothing. If $g(i-1)$ is not defined, then the current set of values defines a classification parameter, and the set is totally described.

(c) If the function π' defined by all currently computed upper bounds is a classification parameter (it is possible to check this because it is defined only on a finite domain) then jump to step 2e. Else go to 2d.

(d) Enumerate (x_i, p_i, q_i). Memorize $\pi(x_i) \leqslant m_i$ if the previously defined upper bound was higher. Resume the computation at step 2b.

(e) The computation is stopped. The new value is discarded. The set has been completely enumerated. □

This theorem proves that classification parameters can be enumerated by a class of algorithms in the same way that recursive functions are enumerated by Turing machines. In the case of Turing machines, a special machine can play the role of any other: they are called universal machines. In the next section, we prove that there exists a classification parameter called *optimal* which is better (in a certain sense) than any other.

5.3. Our parameter κ

In the following section, $x \in \mathbb{S}^{\mathbb{S}^{2r+1}}$ is a CA table, given as a string of S^{2r+1} elements of $\mathbb{S}$.

Definition 1. An *optimal* algorithmic classification parameter π_0 is such that for all classification parameter π

$$\exists c \in \mathbb{N},\ \forall x \in \mathbb{S}^{\mathbb{S}^{2r+1}}, \quad \pi_0(x) \leqslant \pi(x) + \frac{c}{l(x)}.$$

Beware that we have not yet proved the existence of an optimal parameter.

This definition formalizes the claim that "π_0 is better than any other parameter": those tables which get high marks with π_0 also get high marks with any other parameter π. Our thesis is that an optimal algorithmic classification parameter is a good tool for analyzing CA behavior. The key idea is that tables with many regularities can be described by rather short programs. Thus, we can take one of these programs as a representation of the table and consider its length as a measure of the complexity of the table itself.

Inspired by the construction of Martin-Löf [20], we prove that there exists an optimal parameter, and that it can be expressed in terms of Kolmogorov complexity. This is possible with the help of the adequate normalization property imposed in the definition.

Theorem 2. *There exists at least one optimal classification parameter. One of them that we denote by* κ *can be expressed by* $\kappa(x) = (\mathbf{K}(x|l(x)) + 1)/l(x)$.

Proof. We have to show that this function fulfills our three conditions: approximability from above, the normalization condition, and the one about optimality.

1. Since $\mathbf{K}$ is approximable from above, κ is also approximable from above.
2. The cardinality of $\{x, l(x) = n,\ \mathbf{K}(x|n) \leqslant k\}$ is less than $S^{k+1} - 1$. By setting $k = m - 1$, we get that ard$\{x, l(x) = n, \kappa(x) \leqslant m/n\} \leqslant S^m - 1$. Thus κ is an algorithmic classification parameter.
3. We conclude the proof by showing the optimality. Consider a parameter π_y. y is an identifier for the parameter whose existence can be deduced from Theorem 1. We build a description of x such that $\mathbf{K}(x|l(x)) \leqslant l(x)\pi_y(x) - 1 + c_y$ for all y and all x. Thus we obtain $\kappa(x) \leqslant \pi_y(x) + c_y/l(x)$.

   First, define the set of words (of the same length as x since we use $\mathbf{K}(x \mid l(x))$) such that $\pi_y(z) \leqslant \pi_y(x)$. This set can be enumerated given $\pi_y(x)$, y and $l(x)$. x can be described by its index j in this set together with the data quoted above. An upper bound for j is $S^{l(x)\pi_y(x)}$, since ard$\{z, l(z) = n, \pi_y(z) \leqslant \pi_y(x)\} \leqslant S^{l(x)\pi_y(x)}$.

   Therefore, we can write a string s of size exactly $l(x) - l(x)\pi_y(x) + 1$ beginning by 0s, then a 1, then a representation of j. In this manner, from s and $l(x)$, we can compute $\pi_y(x)$ and j. On the input $1^{l(y)}0ys$, we can compute x knowing only $l(x)$ as extra data. Then $\mathbf{K}(x|l(x)) \leqslant l(1^{l(y)}0ys)$. Therefore $\mathbf{K}(x|l(x)) \leqslant l(x) - (l(x)\pi_y(x)) + 2l(y) + 2$. Since $l(x)\pi_y$ is nonnegative, the required constant is $c_y = 2l(y) + 3$. □

From Theorem 2 one can immediately deduce that κ captures all recursive regularities of the tables.

5.4. Comparison with Wolfram's approach

In this section, we analyze the relations between the complexity of the table of a CA and the degree of randomness of its computation triangles (see Section 3 for the definition).

Recall that a word x is *c-random* if and only if $l(x) - \mathbf{K}(x) \leqslant c$. In the sequel, for the sake of simplicity, we will just call them *random*. In the proof of Theorem 3 we shall use the following well-known inequalities for the complexity of an ordered pair (a,b):

$$K(a,b) \geqslant K(a) + K(b|a) + \mathcal{O}(\log(\min(K(b), K(b))), \tag{2}$$

$$K(a,b) \leqslant K(a) + K(b|a) + \mathcal{O}(\log(\max(K(b), K(b))). \tag{3}$$

It is important to remark that $\mathcal{O}(\cdot)$ terms in the above inequalities are logarithmic in $K(a) + K(b)$ and therefore negligible in most practical cases. We also stress that these addenda depend on the variety of Kolmogorov complexity used (see [18], for more details).

Theorem 3. *For all CA tables x and for all random initial configurations I, if we denote by $\mathbb{T}$ the computation triangle of x over I then*

$$\mathbf{K}(\mathbb{T}) \leqslant l(x)\kappa(x) + l(I) + \mathcal{O}(\max(l(x)\kappa(x), l(I))).$$

Conversely, for all random initial configurations I and for all computation triangles $\mathbb{T}$ (based on I) there exists a table x such that

$$\mathbf{K}(\mathbb{T}) \geqslant l(x)\kappa(x) + l(I) + \mathcal{O}(\min(l(x)\kappa(x), l(I)))$$

(beware that the table x may be not the same as the one who generated the computation triangle $\mathbb{T}$).

Proof. A triangle of computation $\mathbb{T}$ is obtained from the table x of the CA and the initial configuration I. Hence, its complexity is bounded by the sum of the complexity of x and the one of I. More formally, for any $c \in \mathbb{N}$, for any $(c/2)$-random table x and any $(c/2)$-random initial configuration, from inequality (3) it follows that

$$l(x)\kappa(x) + \mathbf{K}(I|l(I)) + c_1 \geqslant \mathbf{K}(\mathbb{T}|l(\mathbb{T})) \geqslant l(x)\kappa(x) + \mathbf{K}(I|l(I)) - c + c_2$$

for some suitable constants $c_1, c_2 \in \mathbb{N}$; note that we have dropped the logarithmic terms for the sake of simplicity. The left part of the inequality is the first part of the theorem.

In order to prove the second part of the theorem one has to note that the computation triangle $\mathbb{T}$ can be computed from the initial configuration and the table of the reference automaton. Conversely, a table x can be deduced from $\mathbb{T}$ and from the initial configuration by the following trivial "table-deduction" algorithm: consider the transition table of this new automaton is initially empty; then, for all transitions pictured in the triangle $\mathbb{T}$, set the transition in the table to its actual value and return the table. Hence, $\mathbf{K}(\mathbb{T}) = \mathbf{K}(x, I) + \mathcal{O}(1)$, where the $\mathcal{O}(1)$ term takes in account the complexity of the "table-deduction" algorithm. As usual we drop the logarithmic terms for simplicity obtaining: $\mathbf{K}(\mathbb{T}) = \mathbf{K}(x) + \mathbf{K}(I|x)$. Since, by hypothesis, all initial configurations I are random, then I is independent from x and hence $\mathbf{K}(I|x) = \mathbf{K}(I) = l(I)$. Using inequality (2) one obtains $\mathbf{K}(\mathbb{T}) \geqslant \mathbf{K}(x) + l(I) \geqslant l(x)\kappa(x) + l(I)$, which completes the proof. □

We would like to refine the remark made at the end of statement of Theorem 3: the table turns out to be unique provided that one has a sequence of computation triangles which are sufficiently large and random. This fact is formalized in the following proposition.

Proposition 2. *Let S be a fixed number of states, α a fixed integer and $(\mathbb{T}_{C_n})$ be a sequence of triangles of base width n with S states indexed by $\mathbb{N}$. The sequence is such that all initial configurations C_n are α-random. Let us note u_n the number of possible tables that generate $\mathbb{T}_{C_n}$. Then $\lim_{n\to\infty} u_n = 1$.*

Proof. The proof is straightforward. When triangles are sufficiently large, all possible words of S letters with a size of $2r+1$ are bound to appear in the computation triangle. If they do not, the blocks of size $2r+1$ can be rewritten with an alphabet of only $S^{2r+1}-1$ symbols instead of S^{2r+1}. The complexity of such strings is upper bounded by $n \log_{S^{2r+1}}(S^{2r+1}-1)$. Hence, the randomness deficiency exceeds by a fixed constant for any fixed integer α (that is to say, the triangle is no more α-random). The construction method leaves no ambiguity for the cellular automaton, yielding a unique table that can generate the computation triangle. □

By Martin-Löf randomness theorem [19], for any c-random triangle (not necessarily a computation triangle) $\mathbb{T}$ of height $1+t$ and base width $2rt+1$, $\mathbf{K}(\mathbb{T}|l(\mathbb{T}))\geqslant(1+t)(1+tr)-c$. Therefore for any fixed CA, the progression of the complexity of $\mathbb{T}$ compared with the one of a random triangle of same size is t compared with t^2. Hence, no CA computation is random. However, we can discuss the different factors that contribute to the final complexity of a triangle of computation. These factors depend on the size of $\mathbb{T}$ and on the size of the CA table. Thus, if one wants to make computer simulations of CA evolutions, one should make a trade-off between these quantities.

A good idea is to minimize the effect of the initial configuration on the global complexity of the computation triangle, and at the same time give the CA enough room to distinguish itself from all the others. So we should use configurations of height $h(S,r)\geqslant(\sqrt{S^{2r+1}/r})$, and of width rt. As a consequence, the order of growth of such observations is in the range $\mathcal{O}([\sqrt{S^{2r+1}/r}, S^{2r+1}/r])$. By this notation, we mean that there exist two constants k_1 and k_2 such that $h(S,r)$ verifies the following inequalities:

$$h(s,r)\leqslant k_1\frac{S^{2r+1}}{r}, \qquad k_2\sqrt{\frac{S^{2r+1}}{r}}\leqslant h(s,r).$$

The constants k_1 and k_2 represent the tolerance of the various factors for a given analysis. We remark that almost all practical observations that we can see in literature are in such a range, for small k_1 and k_2. Our thesis is that κ is a good measure of the complexity of evolutions.

Consider the following example. Suppose that our aim is to model a percolation phenomenon, for instance coffee percolation. The interest of modeling such a phenomenon by a CA is that we can implement the physical rules of interaction between particles by a CA local rule. What turns out is a system which gives evolutions comparable to the classical (discretised) partial differential equations. This approach is quite natural and more direct than numerical simulation of percolation equations. Now, suppose we have a local rule of a CA solving the problem, it is not necessary that it makes good simulations on the whole set of initial configurations, but only on a subset which represents a flow of particles through a porous medium.

This subset of configurations can be specified by a computer program, i.e. a recursive function. Hence, it makes sense to investigate CA behavior restricted to a recursive subset of configurations.

Our algorithmic complexity approach is robust with respect to this situation. The formalization requires only some straightforward changes on the Kolmogorov complexity of initial configurations.

Proposition 3. *For any CA of table x, the complexities of all its computation triangles evolving from an enumerable set C_n of initial configurations of length n are bounded by* $l(x)\kappa(x) + \log \mathrm{Card}|C_n|$.

5.5. *An example of application: balanced CA*

In case of CA, *balance* is a necessary property for chaoticity (since it is equivalent to surjectivity). In the sequel we are particularly interested in the case of 1-balanced CA, i.e. those CA tables in which the number of occurrences of all outputs $a \in \mathbb{S}$ are the same: $S^{2r+1}/S = S^{2r}$ (trivially 1-balance is necessary for chaoticity too).

Theorem 4. *The property of* 1*-balance implies nonrandomness of CA tables.*

Proof. If a CA is 1-balanced then one can give a description of its table which is shorter than $S^{2r+1} \log S$ can be given. We need to give a description of the table which is valid for all CA. The set of CA can be described first by giving for each state $a \in \mathbb{S}$ the difference between the number of inputs giving this state $|x|_a$ and $S^{2r+1}/S = S^{2r}$. We use self-delimited notations for these numbers. Then we add in the description the index of the CA in the set containing all CA that are unbalanced the same way. In the case of a 1-balanced CA, the first part of the description uses only $\mathbf{K}(S) + \mathbf{K}(r)$ bits (as the excess is always 0). Let us now compute the cardinality of the set of 1-balanced CA. We have to choose exactly S^{2r} places in a range of S^{2r+1} for the first element of $\mathbb{S}$. Then we have to place S^{2r} occurrences of the second element of $\mathbb{S}$ in the remaining places, and so on. This yields a number of 1-balanced CA tables that is exactly

$$\prod_{i=0}^{S-2} \binom{(S-i)S^{2r}}{S^{2r}}.$$

An approximation can be found using Stirling's formula. Let $A = S^{2r}$. We obtain

$$\prod_{i=0}^{S-2} \binom{(S-i)A}{A} = \prod_{i=0}^{S-2} \frac{((S-i)A)!}{((S-i-1)A)!A!} \sim S^{SA} \sqrt{\frac{S}{(2\pi A)^{S-1}}}.$$

We would like to compute the randomness deficiency of this class of CA relative to κ. This is the difference between the maximal value of κ for a random table (i.e. 1) and the actual value of κ. The cardinality of the class gives an upper bound on the Kolmogorov complexity of all tables belonging to that class; which can be turned into a lower bound on the randomness deficiency, through a division and a subtraction from 1. As we compute an order of equivalence, some constant terms and smaller-order terms can be removed. The maximal complexity is equivalent to $\log(S^{S^{2r+1}}\sqrt{S/(2\pi S^{2r})^{S-1}})$. If we divide this by $\log S^{S^2 r+1}$ and subtract this from 1, we obtain that the randomness deficiency in terms of κ is lower bounded by

$$\frac{\log(2\pi)(S-1)+2r(S-1)\log S-1/2\log S}{S^{2r+1}\log S}.$$

After a tedious computation, and using the fact that $S \geqslant 2$, we obtain that the randomness deficiency relative to κ (i.e. the quantity $1-\mathbf{K}(x|l(x))/l(x)$) is greater than $\Omega(r/S^{2r})$.[2] □

Corollary 1. *The following properties imply nonrandomness of CA tables*: *topological chaos*, *surjectivity and injectivity.*

Therefore topologically chaotic CA have a κ which is reduced by a certain amount, and hence is not maximal. This means that, compared with the maximal complexity of CA, those that are surjective are not random. This amount is relatively significant when r and S are small but tends to 0 when r and S grow.

5.6. Relations with Langton's parameter

In this section, we investigate which CA have high Kolmogorov complexity. The preceding section shows that cellular automata with $\lambda_s = 1-1/S$ are not random, because their randomness deficiency is at least $\Omega(r/S^{2r})$. However, in a random string the number of occurrences of each state should be about equal *but not exactly equal* [21, 18].

A precise evaluation of Langton's $\lambda_s(x)$ for a c-random table x gives

$$1-\pi_{\lambda_s}(x) = \left(1-\frac{1}{S}\right) - \lambda_s(x) = \mathcal{O}\left(\frac{1}{\sqrt{l(x)}}\right).$$

In Langton's original point of view, λ_s was used to measure the intrinsic level of chaos in CA. Crutchfield et al. pointed out in [7] that experimentally, $\lambda_s = 1-1/S$ seems to be a *local minimum* (instead of a global maximum) for chaoticity in CA, but they agree that most chaotic CA have λ_s not far from $1-1/S$. This result is compatible with our parameter, since it corresponds to the fact that 1-balanced CA have a κ that is not maximum. At the same time, CA with maximum κ are not far from being 1-balanced.

If the experimental study reported in the previously cited paper [7] is sound, then the behavior of κ is a possible mathematical explanation of these outcomes.

[2] This quantity is relative to κ. It is a randomness deficiency of $\Omega(Sr)$ in terms of Kolmogorov complexity $\mathbf{K}$.

6. Protocol proposition

Let us consider a simple example which illustrates very clearly the differences between our algorithmic approach and the topological one. Consider the two cellular automata "identity" and "xor" (elementary rule 204 and 90, respectively). For us, both of them are "simple" because their tables can be drastically compressed. This corresponds to some intuition: the dynamics of both these cellular automata can be easily recognized, their regularities are clear and easy to understand. On the other hand, from a topological point of view, these automata are very different: "xor" is expansive, it is transitive, and hence topologically chaotic, according some of the most popular definitions of topological chaos. But this chaoticity does not correspond to the fact that evolutions are algorithmically complex; this last fact is taken into account by our approach.

The reader could object that we have considered only a finite number of cellular automata as an example and this is not correct since our parameter is not absolute; we should have considered an infinite family of cellular automata. This can be done by replacing "identity" and "xor" by the infinite family of additive cellular automata. For us all these CA are simple because their tables are compressible, and experimentally their evolutions are very special and can be recognized, having simple regularities. On the other hand, in the topological sense, we can find many different dynamical behaviors inside the family of additive CA. It can be very interesting to find out which CA are at the same time topologically chaotic and algorithmically complex. This requires to make practical calculations using κ. Here is the problem: κ is not computable but only approximable by above. We suggest to overcome this problem using an approximation of κ and propose the following experimental protocol.

Experimental protocol. We suggest to replace the evaluation of κ (defined via Kolmogorov complexity) by the compression ratio of its table; we propose to use any practically efficient compression algorithm.

We plan to apply this method in some practical cases, and check if the above approximation of algorithmic chaos agrees with intuitive observations.

Finally, because of the small value of the randomness deficiency, practical experimentations on *large* tables will probably fail to detect the small value of the logarithmic gap. Nevertheless, when experimentations are made on rather small tables, we think that it is possible to apprehend it.

Acknowledgements

We are very grateful to Pr. Vladimir Andreevich Uspensky for many helpful discussions on varieties of Kolmogorov complexity and randomness.

References

[1] J. Banks, J. Brooks, G. Davis, G. Cairns, P. Stacey, On Devaney's definition of chaos, Amer. Math. Monthly 99 (1992) 332–334.

[2] E.R. Berlekamp, J.H. Conway, R.K. Guy, Winning Ways for Your Mathematical Plays, Vol. 2, Academic Press, New York, 1982.

[3] F. Blanchard, P. Kurka, A. Maas, Topological and measure-theoretic properties of one-dimensional cellular automata, Physica D 103 (1997) 86–99.

[4] G. Braga, G. Cattaneo, P. Flocchini, C. Quaranta Vogliotti, Pattern growth in elementary cellular automata, Theoret. Comput. Sci. 45 (1995) 1–26.

[5] C. Calude, Information and Randomness, Springer, Berlin, 1994.

[6] G. Cattaneo, E. Formenti, L. Margara, J. Mazoyer, A shift-invariant metric on S^Z inducing a non-trivial topology, in: I. Privara, P. Rusika (Eds.), Mathematical Foundations of Computer Science (MFCS'97), Lecture Notes in Computer Science, Vol. 1295, Bratislava, Springer, Berlin, 1997.

[7] J.P. Crutchfield, P.T. Haber, M. Mitchell, Revisting the edge of chaos: evolving cellular automata to perform computations, Comput. Systems 7 (1993) 89–130.

[8] K. Čulik, J. Pachl, S. Yu, On the limit set of cellular automata, SIAM J. Comput. 18 (1989) 167–175.

[9] R.L. Davaney, Introduction to Chaotic Dynamical Systems, 2nd Edition, Addison-Wesley, Reading, MA, 1989.

[10] B. Durand, Zs. Róka, The game of life: universality revisited, in: J. Mazoyer, M. Delorme (Ed.), Cellular Automata: A Parallel Model, Kluwer, Dordrecht, 1999.

[11] H. Gutowitz, C. Langton, Mean field theory and the edge of chaos, in proc. 3rd Europ. Conf. on Art. Life, 1995.

[12] G.A. Hedlund, Endomorphism and automorphism of the shift dynamical systems, Math. Systems Theory 3 (1969) 320–375.

[13] M. Hurley, Attractors in cellular automata, Ergodic. Theory Dynamical Systems 10 (1990) 131–140.

[14] J. Kari, Rice's theorem for the limit set of cellular automata, Theoret. Comput. Sci. 127 (1994) 229–254.

[15] C. Knudsen, Chaos without nonperiodicity, Amer. Math. Monthly 101 (1994) 563–565.

[16] P. Kurka, Languages, equicontinuity and attractors in cellular automata, Ergodic Theory Dynamical Systems 17 (1997) 417–433.

[17] C. Langton, Computation at the edge of chaos: phase transitions and emergent computation, Physica D 42 (1990) 12–37.

[18] M. Li, P. Vitányi, An Introductioin to Kolmogorov Complexity and its Applications, 2nd Edition, Springer, Berlin, 1997.

[19] P. Martin-Löf, The definition of random sequences, Inform. and Control 9 (1966) 602–619.

[20] P. Martin-Löf, Complexity oscillations in infinite binary sequences, Z. Wahrsch. Ver. Geb. 19 (1971) 223–230.

[21] V.A. Uspensky, A.L. Semenov, A.Kh. Shen, Can individual sequences of zeros and ones be random?, Russian Math. Surveys 45 (1990) 121–189.

[22] V.A. Uspensky, Kh. Shen, Relations between varieties of Kolmogorov complexities, Math. Systems Theory 29 (3) (1996) 270–291.

[23] S. Wolfram, Computation theory of cellular automata, Comm. Math. Phys. 96 (1984) 15–57.