Compositional Experimental Analysis of Cellular Automata: Attraction Properties and Logic Disjunction

P. Flocchini* F. Geurts† N. Santoro*

Abstract

In this paper, we analyze attraction properties of elementary (i.e. Boolean, one-dimensional, bi-infinite) cellular automata (for short, CA). To overcome the well-known undecidability constraints met by these systems, in addition to the classical extensive use of computer simulations, we introduce composition: we first characterize basis CA, which we then use as building blocks to understand a whole family of CA-based systems obtained by composing them using logic disjunction. The compositional approach allows deep structured investigations, and it permits to define a new notion of dynamical complexity.

1 Experimental analysis of cellular automata

Cellular automata. Cellular automata (for short, CA) are *massively parallel systems* obtained by composition of myriads of *simple agents* interacting locally, i.e. with their closest neighbors. In spite of their simplicity, the dynamics of CA is potentially very rich, and ranges from attracting stable configurations to spatio-temporal chaotic features and pseudo-random generation abilities. Moreover, from the computational viewpoint, they are universal, that is, as powerful as Turing machines and, thus, classical Von Neumann architectures.

These structural and dynamical features make them very powerful: fast CA-based algorithms are developed to solve engineering problems in cryptography and microelectronics for instance (e.g., [11, 10, 12, 44, 27, 29, 41, 48]), and theoretical CA-based models are built in ecology, biology, physics, telecommunications, and image-processing (e.g., [23, 28, 45]).

*School of Computer Science, Carleton University, Ottawa K1S 5B6, Canada, {flocchin, santoro}@scs.carleton.ca.

†Département d'Ingénierie Informatique, Université catholique de Louvain, Place Sainte Barbe 2, B-1348 Louvain-la-Neuve, Belgium, gf@info.ucl.ac.be.

Experimental analysis. On the other hand, these powerful features make CA difficult to analyze: they resist against most of the analytical tools available in program theory and dynamical system theory. Actually, almost all long-term behavioral properties of dynamical systems, and cellular automata in particular, are unpredictable. This limit not only arises from their possibly chaotic dynamics, which entails the classical sensitivity to initial conditions [16], but from a much stronger limit: undecidability [14, 15, 40]. Sensitivity means that close initial conditions eventually diverge. Undecidabilty implies that no algorithm exists whatsoever to decide whether or not an arbitrary state is attracted to some set, given the general description of some dynamical system.

Two pragmatic solutions exist to this problem: the first one consists in restricting the study to particular cases and sufficiently simple systems that may be used as building blocks; the second one is to make an extensive use of computer simulations. Both solutions are necessary engineering tools in the construction of systems that must be built in order to fulfill a priori conditions or specifications. In the following sections, we shall adopt both of them.

Aim. Our objective is the study of dynamical properties of CA-based systems.

The main property we focus on is attraction, as many behaviors of systems can be rephrased in these terms: from particular sets of states, the *initial condition* or the *attraction basin*, a given system has to reach some other set of states, the *final condition* or the *attractor*, after some evolution time. In control theory, in computer programming or, more abstractly, in language theory, attraction plays the same role in describing how robots, programs, or automata, follow state sequences according to specified functions linking actors to sensors, output to input data, or output to input words.

Moreover, due to the severe theoretic limitations explained above, we carry out the *experimental analysis* of systems in a *compositional* way, i.e. by combination of properties of their components: we analyze attraction properties of basis CA, and we combine these basis CA to obtain the dynamics of a whole family of CA-based systems.

In particular, our aim is to make use of the very nature of the structure of CA to understand their global behavior, since dynamical atttraction-based properties of CA apparently depend more strongly on their structure than on the way initial conditions are chosen. Given two elementary CA (Boolean one-dimensional bi-infinite lattices of cells, the evolution of each cell being influenced by its direct neighbors) with known behaviors, what can be said on the behavior of the CA obtained by composing them (e.g., using a logic disjunction)? Based on this, we give a new definition of complexity based on the structure of systems.

Paper outline. In §2, we introduce the philosophy of our theoretic and pragmatic understanding of composed attraction-based systems. This further motivates our aim to analyze and synthesize CA-based systems by composition. In §3, we present the necessary background, including the definitions of CA, a composition principle, important dynamical properties, the attraction-based classification of CA behaviors, and a general notion of dynamical complexity. In §4, we concentrate on a specific composition operator (the logic disjunction), which allows us to give a complete characterization of the dynamics of all elementary CA: we study CA as compositions of simpler "basis" ones and we interprete these results according to attraction and to the classification introduced before. The results of this analysis reveal new insights for understanding complexity in multi-dimensional dynamical systems and cellular automata. Finally, in §5, we draw some conclusions.

2 Composed, attraction-based systems

Design of attraction-based systems. Program development consists in building a correct program from a formal specification of what it is intended to do. Theoretically, a specification corresponds to the description of a language an automaton should recognize. The development of a program then corresponds to building the automaton. Using the terminolgy used in dynamical systems theory, the language corresponds to the attractor of the automaton.

This classical view can be extended to dynamical systems: given the specification of a problem, design a system whose dynamics solves the problem. Stated otherwise, given a set of states in a state space, elaborate a system whose attractor precisely is this set of states:

$$\begin{array}{ll} & \text{specification} \xrightarrow{\text{design}} \text{program} \\ \text{viz.} & \text{attracting set} \xrightarrow{\text{design}} \text{dynamical system.} \end{array}$$

In control and hybrid systems theory, recent advances combining techniques from both dynamical systems and program theory have already proved useful [34, 4, 49]. Similarly, cellular automata have been used as building blocks of structured systems, in order to derive original solutions of well-known problems in distributed algorithms (synchronization of processes, iterative generation of fractals, etc.) [37, 23].

The advantages of extending software design to dynamical systems are numerous [7]. Fault-tolerance, stability, and stabilization properties are fundamental in real-time systems for instance, and they are naturally expressed as attraction and invariance conditions in the context of dynamical systems [47, 26]. This does not mean that the design of such properties in the context of dynamical systems is immediate. Actually, they often remain quite hard to derive using current mathematical methods of program development. Hence, the techniques and criteria developed for dynamical systems should be exploited in this direction.

The challenge is to derive efficient dynamical systems from specified attraction and invariance properties. This would amount to define a design method, a language in which properties are expressed, and a logic in which a constructive reasoning can be conducted.

Structural composition. Compositionality is to dynamical systems what modularity is to software design. Hence, we think and hope that the compositional analysis of systems could be used as a framework for the systematic design of attraction-based systems.

Decomposing a mathematical object is often crucial to understand it. In program theory, sequential and parallel programs are considered as structurally composed systems. From simple composition operators like sequential composition of guarded-commands in Dijkstra's language [18] to modular approaches in software engineering, composition is of fundamental importance. Parallel composition, "rendez-vous" synchronization, product, sum, are good examples of structuring means of programs, transition systems, or algebraic processes [31, 38, 39]. Compositional analyses of these systems are elaborated, to overcome the impossibility of dealing with huge monolithic systems.

Surprisingly, dynamical systems are often studied as complex mathematical objects, without trying to decompose them into simpler components. Some exceptions exist, particularly in the field of hybrid systems [42]. Usually, composition designates successive applications of functions to given states: f and g being two functions defined on X, $x \in X$, $(f \circ g)(x) = f(g(x))$. Another structural way to combine systems is the simple Cartesian product of function; for instance, $(f \times g)(x, y) = (f(x), g(y))$. Finally, independent variables can be mixed together, as in any matrix-vector multiplication.

Compositional analysis. Basically, the compositional analysis of a dynamical system consists in determining some global property concerning dynamical or computational aspects of the system by combination of individual properties of its components, which are expected to be simpler [24].

The following diagram illustrates the idea; S_i being the components of a composed system $S = \star_i S_i$, I denoting an individual property, G a global property, we want to find a way to combine the individual properties, viz. $\diamond$, to characterize the global property:

$$\begin{tikzcd} S_i \arrow[r, "I"] \arrow[d, "\star"'] & I(S_i) \arrow[d, "\diamond"] \\ \star_i S_i \arrow[r, "G"'] & \diamond_i I(S_i). \end{tikzcd}$$

To this end, composition operators must be defined, as well as interesting local and global properties. Then, the difficult part is to find $\diamond$, that is, how composition propagates through properties. Moreover, it is not always possible to express the relationship between components and composed systems so easily: it is sometimes necessary to add some global information in addition to the local properties.

Related work. The method we propose and the techniques we introduce are strongly influenced by program theory, where they prove most useful.

In computing science, programs are defined as relations, and predicate-transformers express their semantics. Standard composition operators are introduced to build nondeterministic sequential programs, and their execution effect must be determined [18, 30, 19]. Parallel programs require further composition operators. There, the compositional analysis amounts to prove theorems extending specific properties from basic systems to composed ones [1, 43]. For instance, a program is represented by the set of its possible execution traces; composition is then also defined in terms of traces [38, 17].

Similar compositional approaches are also proposed in [2, 9, 24, 51, 52] for the analysis of different kinds of systems.

3 Framework: cellular automata and their dynamics

In this section, we briefly recall useful definitions and the classification of dynamics used in the rest of the following sections.

3.1 Cellular automata

CA are totally discrete dynamical systems. They are discrete *in space*; in fact, they are regular lattices of sites (or cells) whose values range in a finite set. They are also discrete *in time*: their iterative dynamics is described in terms of difference equations, as opposed to continuous-time systems based on differential equations. Furthermore, CA are homogenenous systems: at each time step, each cell applies the same rule to its neighborhood in order to compute its new value.

We consider one-dimensional CA, i.e. linear bi-infinite lattices of cells. Each cell takes its value in the local state space $X = \{0, 1, \ldots, k-1\}$. A *configuration* is a bi-infinite sequence of $X^{\mathbb{Z}}$ specifying a state for each cell, i.e. $x = (\cdots x_{-1}, x_0, x_1 \cdots)$. The *neighborhood* of a cell $i \in \mathbb{Z}$ is $(i-r, \ldots, i-1, i, i+1, \ldots, i+r)$ or, simply, $(i-r : i+r)$.

All cells are updated synchronously. At each step, every cell looks at the value of its neighbors (r to the left, r to the right) plus itself and computes its next value as a function of this neighborhood. This function is a *local transition function* $g : X^{2r+1} \mapsto X$. There are clearly $k^{k^{2r+1}}$ different local functions.

The *global transition function* defines the next state of each cell as the local function applied to the states of its neighborhood:

$$\begin{array}{ll} f & : \; X^{\mathbb{Z}} \mapsto X^{\mathbb{Z}} \\ \text{s.t.} & \forall i \in \mathbb{Z}, f_i(x) = g(x_{i-r:i+r}). \end{array}$$

In the following, we restrict our attention to *elementary cellular automata*, i.e. with $r = 1$ and $k = 2$. With these parameters, there are 256 different elementary cellular automata. Each transition function is a Boolean function of three Boolean variables (the neighborhood) and is thus expressed as a *transition (or rule) table* with eight entries.

Traditionally [54], a unique integer is associated to each transition function, and is used to label the corresponding CA:

$$\sum_{a,b,c \in \{0,1\}} g(a,b,c) \cdot 2^{4a+2b+c}.$$

Example 1
The following table corresponds to rule $47 = 1 + 2 + 4 + 8 + 32$.

$x_{i-1}^t x_i^t x_{i+1}^t$	000	001	010	011	100	101	110	111
x_i^{t+1}	1	1	1	1	0	1	0	0

⊠

3.2 Composition of cellular automata

Using the connected product introduced in [20, 24], we can give another definition of CA.

Definition 1 (Cellular automaton)
A cellular automaton f is structured as follows:

$$f = \otimes_R g$$

where $J = \mathbb{Z}$ is the lattice of cells; $R = \{(i, i-1), (i, i), (i, i+1) \mid i \in J\}$ describes the neighborhood of each cell; $\forall i \in J, X_i = \{0, 1\}$ is the local state space; $\forall i \in J, g_i = g : X^3 \mapsto X$ is the local transition function.

In general, the analysis of connected products is difficult. Here, we concentrate on the specific connected product describing CA.

Composing CA can be realized in two ways: either *globally*, i.e. considering global functions as black boxes, or *locally*, i.e. at the level of local transition functions. More formally, let $\star$ and $\star'$ be two composition operators. The composition of several CA $\otimes_R g_i$ can be defined outside the connected products (this is the usual global manner), or inside the product:

$$\star_i(\otimes_R g_i) \text{ or } \otimes_R (\star'_i g_i).$$

From an engineering viewpoint, local functions are building blocks, and they are supposed to be well understood, whereas global functions are to be built, and composing global functions is beyond the scope of this paper. Hence, our concern here is the study of local compositions composed within the connected product: f and g being two local transition functions, $\star$ being a local composition operator, we study the system $\otimes(f \star g)$. If G is a global property, and I an individual property, the objective is to find $\diamond$ such that

$$G(\otimes_R(\star'_i g_i)) = \diamond_i I(\otimes_R g_i).$$

However, this kind of local composition clearly adds a difficulty to the analysis of systems, because the property composition operator $\diamond$ has to "jump" over the connected product. This is equivalent to finding intermediate G' and $\diamond'$ such that

$$G(\otimes_R(\star'_i g_i)) = G'(\diamond'_i(\otimes_R g_i)) = \diamond_i I(\otimes_R g_i).$$

3.3 Dynamical properties

In the study of dynamical systems, two properties are important: invariance and attraction [3, 53].

Invariants are sets of states that have infinite internal histories, i.e. stable sets. Their structure represents trajectories between inner states; it organizes and, thus, strongly influences the resulting dynamics.

Definition 2 (Invariance)
The invariant of a system f is the greatest J satisfying

$$J \subseteq f(J) \cap f^{-1}(J),$$

where $f(J) = \{y \mid \exists x \in J \text{ s.t. } f(x) = y\}$ *and* $f^{-1}(J) = \{x \mid \exists y \in J \text{ s.t. } f(x) = y\}$.

Attraction establishes relations between initial and final or asymptotic states of infinite histories.

Definition 3 (Attraction)
The set P is attracted to Q by f iff $\cap_i \overline{\cup_{i \leq j} f^j(P)} = Q$, *where* $\overline{A}$ *expresses the closure of* A.

3.4 Attraction-based classification of dynamics

The following classification of long-term CA behaviors determines six phenomenological classes grouped into three families: periodic (types $\mathcal{N}$, $\mathcal{F}$, $\mathcal{P}$), shifting (type $\mathcal{S}$, divided into types $\mathcal{S}_{\mathcal{F}}$ and $\mathcal{S}_{\mathcal{P}}$), and aperiodic (type $\mathcal{A}$) behaviors; typical evolutions are illustrated in Fig. 1.

Type $\mathcal{N}$: CA evolving to null configurations. This class contains CA that quickly evolve to homogeneous configurations, i.e. without information (all ones or zeroes), after finite transients.

Type $\mathcal{F}$: CA evolving to fixed points. This second class contains CA that evolve to fixed points after finite transients. Of course, class $\mathcal{N}$ is a particular case of class $\mathcal{F}$.

Type $\mathcal{P}$: CA with periodic behaviors. This class contains CA that evolve to periodic configurations, after finite transients. It contains the two previous ones.

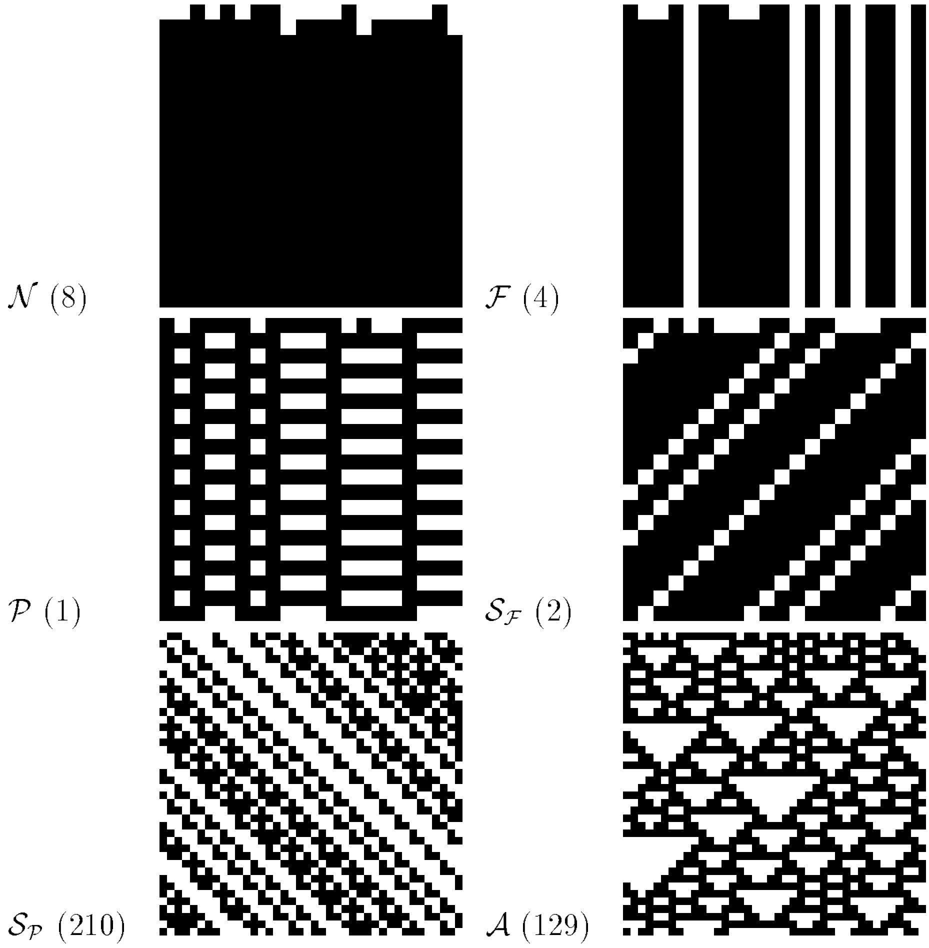

Figure 1: CA classification: typical evolutions of CA classes $\mathcal{N}$, $\mathcal{F}$, $\mathcal{P}$, $\mathcal{S}_{\mathcal{F}}$, $\mathcal{S}_{\mathcal{P}}$, $\mathcal{A}$. In this figure, the horizontal axis represents a portion of the CA state space, the vertical axis represents the temporal evolution (top-down); black dots are 0's and white dots are 1's.

Type $\mathcal{S}$: CA with generalized subshift behaviors. This class contains CA that, starting from an initial configuration in particular subspaces of the configuration space, evolve to configurations where a generalized alternating subshift behavior occurs. Formally, A CA is a generalized alternating subshift rule if the corresponding global function f is such that there is a closed invariant subset Σ_1 of $2^{\mathbb{Z}}$ such that

$$\forall x \in \Sigma_1, f^n(x) = \rho^m(x)$$

where $n \in \mathbb{N}$ and $m \in \mathbb{Z}$, and $\rho : \mathcal{C} \mapsto 2^{\mathbb{Z}}$ is the classical shift, i.e. $\forall x \in \mathcal{C}, \forall i, \rho(x)_i = x_{i+1}$ [8, 6].

Remark 1

- On the complement of the subspace Σ_1 where the CA acts like a shift, the behavior can be regular or totally irregular.

- As shown in Fig. 1, the family of shifting behaviors contains a broad range of behaviors, from very simple ones like the dynamics of rule 2, to very complex ones like the dynamics of rules 210. This is the reason why we distinguish the simplest ones, for which $n = m = 1$ in the above definition (they are denoted by $\mathcal{S}_{\mathcal{F}}$), from the other ones (denoted by $\mathcal{S}_{\mathcal{P}}$). Of course, $\mathcal{S} = \mathcal{S}_{\mathcal{F}} \cup \mathcal{S}_{\mathcal{P}}$. In the following, we will distinguish these two families only when necessary.

Type $\mathcal{A}$: CA with complex or aperiodic behaviors. A configuration is aperiodic if it is not eventually periodic (neither periodic nor one of its forward iterations). Qualitatively, what we observe is a number of different patterns growing, vanishing and moving toward the future. In general, a broad range of behaviors can show up: from random noise, total disorder, and spatio-temporal chaos to some kind of regularity or intermittency in which diverse forms can propagate. Aperiodicity entails that almost the whole domain is visited through successive iterations.

Note that these types can be formally defined in terms of transfinite atttraction [21, 25].

3.5 Complexity

The question of complexity is not easy to address, even for elementary CA; the many existing classifications evidence this. Classification is outside the purpose of this paper; the reader is referred to [25] to get more details than the brief summary given above.

Sometimes, using classical tools, the resulting classification is not intuitive. For example, it is not rare to consider subshift behaviors as chaotic. Of course, using the center-first metric ($\forall x, y \in 2^{\mathbb{Z}}, d(x,y) = \sum_i |x_i - y_i| \cdot 2^{|i|}$), it is possible to prove that they are chaotic *à la* Devaney [16], but when you observe them, what you can see is very simple instead: configurations are smoothly shifted along the lattice (see Fig. 1).

That was our main motivation to introduce a formal classification of CA behaviors corresponding to phenomenological considerations. We needed new tools, namely transfinite attraction and shifted Hamming distance; the former allowing a classification in terms of transient length before convergence to cycles, the latter showing that (periodic) subshifts are simple [21, 24]. Based on this, the following complexity hierarchy can be defined.

Definition 4 (Complexity)
On the set $\{ \mathcal{N} , \mathcal{F} , \mathcal{P} , \mathcal{S}_{\mathcal{F}} , \mathcal{S}_{\mathcal{P}} , \mathcal{A} \}$, *the following ordering is established:* $\mathcal{N} < \mathcal{F} < \mathcal{P} < \mathcal{S}_{\mathcal{F}} < \mathcal{S}_{\mathcal{P}} < \mathcal{A}$. *The complexity* $\Upsilon(g)$ *of a CA based on the local rule* g *is the place of its long-term behavior in the defined ordering.*

The central question arising from this is:

$$\boxed{\text{How to understand } \mathcal{A}\text{ ?}}$$

We propose a compositional approach, in order to characterize complex behaviors. This is the aim of the next section.

4 Composition: disjunction of CA rules

CA are massively parallel homogenenous dynamical systems, composed of many interacting agents. As mentioned in the introduction, their theoretical analysis is most of the time impossible, due to undecidability arguments. Two solutions are used here to overcome this problem: composition and simulation. We introduce a *local composition operator* allowing the generation of all elementary CA rules by combination of simpler basis rules, the dynamical behavior of which is easily characterized. By means of extensive simulations, we then study the dynamics of all combinations of these basis rules. This leads to a clear understanding of all elementary CA, and to a new definition of dynamical complexity based on the structure of systems.

4.1 Composition space

We define a structure of linear space on the set of CA rules whose addition operation corresponds to the set union of 1-transition sets (sets of neighborhoods mapped to 1 by the local transition function). The composition of two CA rules r_1 and r_2 is thus obtained by the componentwize disjunction $\vee$ applied to the vectors v_1 and v_2 of $\{0,1\}^8$ representing the rule tables of r_1 and r_2. The resulting rule is the sum $r_1 + r_2$.

The canonical basis $\mathcal{B} = \{n, n', n^+, n^-, f, p, s^+, s^-\}$ of this linear space corresponds to the set of rules $\{32, 128, 8, 64, 4, 1, 2, 16\}$: the rule table of each basis rule contains a unique one.

As every CA rule realized as a Boolean vector from $\{0,1\}^8$ can be uniquely expressed as a linear combination of basis rules, with coefficients 0 or 1, we study the dynamics of CA in terms of the dynamics of basis rules, using the disjuncntive composition, with respect to the classification of elementary CA behaviors given above.

In the following, we restrict ourselves to the study of at most four basis rules composed together. This is essentially due to de Morgan's duality law: the negation of a disjunction of two values corresponds to the conjunction of the negated values.

1. Each basis rule contains a unique 1. Let such a rule be realized by the local transition function $g(a,b,c)$, and negation be denoted by overlined symbols, the global dynamics of 1s in $g(a,b,c)$ is equivalent to that of 0s in $\overline{g(\overline{a},\overline{b},\overline{c})}$.

2. Let rule g be the disjunction of two basis rules r_1 and r_2, the dynamics of 1s in $g(a,b,c)$ is equivalent to that of 0s in $\overline{g(\overline{a},\overline{b},\overline{c})} = \overline{r_1(\overline{a},\overline{b},\overline{c})} \wedge \overline{r_2(\overline{a},\overline{b},\overline{c})}$. Again, knowing the dynamics of r_1, r_2, and their disjunction leads to the dynamics of the dual of this disjunction.

3. This process can be stopped with the composition of four basis rules, as more than four 1s is equivalent to less than four 0s, which corresponds to the dual of a case already studied before.

4.2 Basis rules

The basis contains elements from each of the first four classes in our classification (see Table 1).

Table 1: Labels, classes, complexities of the basis rules

Rule	Label	Class	Complexity (Υ)
32	n	$\mathcal{N}$	1
128	n'	$\mathcal{N}$	1
8	n^+	$\mathcal{N}$	1
64	n^-	$\mathcal{N}$	1
4	f	$\mathcal{F}$	2
1	p	$\mathcal{P}$	3
2	s^+	$\mathcal{S}_\mathcal{F}$	4
16	s^-	$\mathcal{S}_\mathcal{F}$	4

Type $\mathcal{N}$: Rules 8, 32, 64 and 128 are the elementary null rules. Rules 8, 64 are symmetric, i.e. equal under left-to-right transformation: $g_{n^+}(a,b,c) = g_{n^-}(c,a,b)$; this property is important when we observe the sum of rules.

Type $\mathcal{F}$: Rule 4 is the basis fixed point rule.

Type $\mathcal{P}$: Rule 1 is the simplest periodic rule. It plays a particular role: when it does not appear in the composition of a rule, the CA is so-called *quiescent*, i.e. $f_1(\cdots 000 \cdots) = \cdots 000 \cdots$ globally, or $g_1(000) = 0$ locally.

Type $\mathcal{S}$: Rules 2 and 16 are the elementary subshift rules. They both behave like a shift on a subset $\Sigma_0 = (0^*001)^{\mathbb{Z}}$ of the configuration space, where 0^* stands for any repetition of symbol 0, including the empty repetition. They have opposite directions: 2 is the left shift on Σ_0, while 16 is the right shift. They are symmetric under left-to-right transformation: $g_{s^+}(a,b,c) = g_{s^-}(c,b,a)$.

If rule $r = 2^i$ is realized by the local transition function $g(a,b,c)$, the dual rule $\overline{g(\overline{a},\overline{b},\overline{c})}$ corresponds to rule $255 - 2^{7-i}$. Hence, from the basis rules and the duality principle mentioned above, we obtain the dynamics of rules 251, 254, 239, 253, 223, 127, 191, and 247 for free.

4.3 Composition of two basis rules

By extensive simulation of all disjunctions of two different basis rules starting from random initial configurations, we observe interesting regularities that are summarized in Table 2. This gives us the dynamics of the following rules: $\forall r_i = 2^i, r_j = 2^j \in \mathcal{B}$ with $i \neq j$, rule $r_i + r_j$ and, by duality, rule $255 - 2^{7-i} - 2^{7-j}$.

Table 2: Local $\vee$-composition (2 rules)

		32	128	8	64	4	1	2	16
		n	n'	n^+	n^-	f	p	s^+	s^-
		$\mathcal{N}$	$\mathcal{N}$	$\mathcal{N}$	$\mathcal{N}$	$\mathcal{F}$	$\mathcal{P}$	$\mathcal{S}_\mathcal{F}$	$\mathcal{S}_\mathcal{P}$
n	$\mathcal{N}$	-	$\mathcal{N}$	$\mathcal{N}$	$\mathcal{N}$	$\mathcal{F}$	$\mathcal{P}$	$\mathcal{S}_\mathcal{F}$	$\mathcal{S}_\mathcal{F}$
n'	$\mathcal{N}$	-	-	$\mathcal{N}$	$\mathcal{N}$	$\mathcal{F}$	$\mathcal{A}$	$\mathcal{S}_\mathcal{F}$	$\mathcal{S}_\mathcal{F}$
n^+	$\mathcal{N}$	-	-	-	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{S}$	$\mathcal{S}_\mathcal{F}$	$\mathcal{S}_\mathcal{F}$
n^-	$\mathcal{N}$	-	-	-	-	$\mathcal{F}$	$\mathcal{S}$	$\mathcal{S}_\mathcal{F}$	$\mathcal{S}_\mathcal{F}$
f	$\mathcal{F}$	-	-	-	-	-	$\mathcal{P}$	$\mathcal{S}_\mathcal{P}$	$\mathcal{S}_\mathcal{P}$
p	$\mathcal{P}$	-	-	-	-	-	-	$\mathcal{S}_\mathcal{P}$	$\mathcal{S}_\mathcal{P}$
s^+	$\mathcal{S}_\mathcal{F}$	-	-	-	-	-	-	-	$\mathcal{A}$
s^-	$\mathcal{S}_\mathcal{P}$	-	-	-	-	-	-	-	-

An interesting general law that we can derive from the observation of Table 2 is the following.

Observation 1 (Law of complexity conservation)
In the local disjunctive composition of two basis rules, the complexity never decreases:

$$\forall x, y \in \mathcal{B}, \Upsilon(x \vee y) \geq \max(\Upsilon(x), \Upsilon(y)).$$

In other words, the complexity of the composition of two basis rules is never smaller than the complexities of the single rules. In particular, we have the following:

Observation 2 (Law of complexity increase)
The complexity of the local disjunctive composition of two basis rules increases (e.g., $\Upsilon(x \vee y) > \max(\Upsilon(x), \Upsilon(y))$) iff one of the following conditions holds:

- $x = n^+$, $y = n^-$; (symmetric null rules)
- $x = s^+$, $y = s^-$; (symmetric shifting rules)
- $x \in \{n^+, n^-, n'\}$, $y \in \mathcal{P}$. (periodicity)

Remark 2

- It is important to emphasize that composing two symmetric rules increases the complexity. Two symmetric null rules leads to a fixed point rule, and composing two symmetric shifting rules leads to chaos. This qualitative fact has been noticed in another context by [50], devoted to an analytical compositional result on the complexity of classical low-dimensional chaotic dynamical systems. Moreover, in [22], the authors compare a well-known chaotic dynamical system, namely Cantor's relation, to the composition of symmetric shifting CA rules.

- The compositions with the periodic basis rule p are also interesting since it has already been observed that the presence of the transition $g_p(000) = 1$ in the rule could have a special role in the complexity of the rule (see e.g. [35]).

Another property concerns the composition of non-symmetric null rules: composed together, they always give rise to null rules.

Observation 3
The sum of any null rule with n or n' is a null rule:

$$\forall n_i \in \{n, n'\}, n_j \in \{n^-, n, n', n^+\}, (n_i \vee n_j) \in \mathcal{N} \ .$$

A characteristic of the basis fixed point rule f is that, when it is composed with a less complex rule, the result still belongs to $\mathcal{F}$.

Observation 4
The sum of any null rule with f is an $\mathcal{F}$ rule, that is:

$$\forall n_j \in \{n^-, n, n', n^+\}, (n_j \vee f) \in \mathcal{F} \ .$$

4.4 Combination of three basis rules

Repeating the same procedure as for binary disjunctions, that is, simulating rules with random initial conditions, we obtain all disjunctions of three different basis rules. Again, we double the explored space by duality. This gives us the following combinations of three different basis rules $r_i = 2^i, r_j = 2^j, r_k = 2^k$: $r_i + r_j + r_k$ and $255 - 2^{7-i} - 2^{7-j} - 2^{7-k}$. The observations are summarized in Table 3.

First, in order to have a subshift rule, it is necessary to have a shift basis rule in the sum.

Observation 5
The presence of s^+ or s^- in the sum is necessary to get $\mathcal{S}$ rules.

Observation 6
The presence of s^+ or s^- creates $\mathcal{S}_{\mathcal{F}}$ rules when combined with only null basis rules, more complex $\mathcal{S}_{\mathcal{P}}$ when combined with f or p basis rules. The only exceptions to this observation are given by $\{n^+f, n^-f\}$ combined with $\{s^+, s^-\}$.

Observation 7 (Second law of complexity conservation)
In the local disjunctive composition of three basis rules, the complexity never decreases:

$$\forall x \in \mathcal{B}^2, \forall y \in \mathcal{B}, \Upsilon(x \vee y) \geq \max(\Upsilon(x), \Upsilon(y))$$

except in cases $n^+f \vee \{s^-, p\}$ and $n^-f \vee \{s^+, p\}$.

Table 3: Local $\vee$-composition (3 rules)

		n^+	n^-	f	s^+	s^-	p
nn'	$\mathcal{N}$	$\mathcal{N}$	$\mathcal{N}$	$\mathcal{F}$	$\mathcal{S}_\mathcal{F}$	$\mathcal{S}_\mathcal{F}$	$\mathcal{A}$
nn^+	$\mathcal{N}$		$\mathcal{F}$	$\mathcal{F}$	$\mathcal{S}_\mathcal{F}$	$\mathcal{S}_\mathcal{F}$	$\mathcal{A}$
$n'n^+$	$\mathcal{N}$		$\mathcal{F}$	$\mathcal{F}$	$\mathcal{S}_\mathcal{F}$	$\mathcal{S}_\mathcal{F}$	$\mathcal{A}$
$n'n^-$	$\mathcal{N}$			$\mathcal{F}$	$\mathcal{S}_\mathcal{F}$	$\mathcal{S}_\mathcal{F}$	$\mathcal{A}$
nn^-	$\mathcal{N}$			$\mathcal{F}$	$\mathcal{S}_\mathcal{F}$	$\mathcal{S}_\mathcal{F}$	$\mathcal{A}$
n^+n^-	$\mathcal{F}$			$\mathcal{F}$	$\mathcal{S}_\mathcal{F}$	$\mathcal{S}_\mathcal{F}$	$\mathcal{P}$
nf	$\mathcal{F}$				$\mathcal{S}_\mathcal{P}$	$\mathcal{S}_\mathcal{P}$	$\mathcal{P}$
$n'f$	$\mathcal{F}$				$\mathcal{S}_\mathcal{P}$	$\mathcal{S}_\mathcal{P}$	$\mathcal{P}$
n^+f	$\mathcal{F}$				$\mathcal{S}_\mathcal{F}$	$\mathcal{P}$	$\mathcal{F}$
n^-f	$\mathcal{F}$				$\mathcal{P}$	$\mathcal{S}_\mathcal{F}$	$\mathcal{F}$
np	$\mathcal{P}$				$\mathcal{S}_\mathcal{P}$	$\mathcal{S}_\mathcal{P}$	
$n'p$	$\mathcal{A}$				$\mathcal{S}_\mathcal{P}$	$\mathcal{S}_\mathcal{P}$	
n^+p	$\mathcal{S}_\mathcal{P}$				$\mathcal{S}_\mathcal{P}$	$\mathcal{S}_\mathcal{P}$	
n^-p	$\mathcal{S}_\mathcal{P}$				$\mathcal{S}_\mathcal{P}$	$\mathcal{S}_\mathcal{P}$	
fp	$\mathcal{P}$				$\mathcal{S}_\mathcal{P}$	$\mathcal{S}_\mathcal{P}$	

	s^+s^-
n	$\mathcal{P}$
n'	$\mathcal{A}$
n^+	$\mathcal{A}$
n^-	$\mathcal{A}$
f	$\mathcal{A}$
p	$\mathcal{P}$

Observation 8 (Third law of complexity conservation)
In the local disjunctive composition of three basis rules, the complexity never decreases:

$$\forall x, y, z \in \mathcal{B}, \Upsilon(x \vee y \vee z) \geq \max(\Upsilon(x), \Upsilon(y), \Upsilon(z))$$

except in cases $n^+ \vee f \vee \{s^-, p\}$, $n^- \vee f \vee \{s^+, p\}$, $n' \vee p \vee \{s^+, s^-\}$ and $s^+ \vee s^- \vee \{n, p\}$.

Another general observation that can be derived from Table 3 is the following.

Observation 9
The fixed point rules of $\mathcal{F}$ obtained by compositions of basis rules never contain a basis shift rule in their composition.

When only null rules are composed, we have simple behaviors.

Observation 10
The sum of null basis rules leads to null rules. When pair n^+n^- is present, the sum belongs to $\mathcal{F}$.

This also means that the presence of both n^+ and n^- increases complexity, at least up to $\mathcal{F}$ rules. In the same vein, as in the composition of two rules, the sum of s^+ and s^- has a special effect: it creates complex behavior because of the symmetric nature of the rules.

Observation 11
The presence of s^+s^- is a sign of complexity; it means that we are in presence of a complex rule.

We can conclude that the presence of subshift basis rules in the sum gives to the resulting rule a subshift character, a simple subshift if combined with a null rule, a complex subshift if combined with a fixed point rule. Furthermore, they create complex behaviors if combined together. In the absence of $\mathcal{S}$ rules, f leads to $\mathcal{F}$ and n leads to $\mathcal{N}$. Notice again that the presence of p, i.e. $g(000) = 1$, introduces some complexity in a nonregular way.

4.5 Combination of four basis rules

We now focus on disjunctions of four different basis rules where we observe less regular combinations than in the binary and ternary compositions detailed in §§4.3 and 4.4. The experimental results we get are summarized in Table 4.

Table 4: Local $\vee$-composition (4 rules)

$\mathcal{F}$	$\mathcal{P}$	$\mathcal{S}_{\mathcal{F}}$	$\mathcal{S}_{\mathcal{P}}$	$\mathcal{A}$
n^+n^-fp	fps^+s^-	$nn'n^+s^+$	n^+fps^+	nn^+n^-p
$n'n^+fp$	n^+fps^-	$nn'n^-s^-$	n^-fps^-	nn^+fp
$n'n^-fp$	n^-fps^+	$nn'n^-s^+$	$nfps^+$	nn^-fp
$n^+n^-fs^+$	nps^+s^-	$nn'n^+s^-$	$nfps^-$	$nn'fp$
$n^+n^-fs^-$	nn^+n^-f	nn^+fs^+	nn^-ps^-	nn^-fs^+
$nn'n^+f$	$n'n^+fs^-$	nn^-fs^-	nn^+ps^+	nn^+fs^-
$nn'n^-f$	$nn's^+s^-$	$n'n^-fs^-$	$n'n^-s^+s^-$	$n^+n^-ps^+$
$n'n^+n^-f$	$n'n^-fs^+$	nn^+ps^-	$n'n^+s^+s^-$	$n^+n^-ps^-$
$n'n^+n^-s^+$	$n'n^+n^-p$	nn^-ps^+	nfs^+s^-	$n'n^+ps^-$
$n'n^+n^-s^-$		$nn'ps^+$	$n^+ps^+s^-$	$n'n^-ps^+$
$nn'n^+n^-$		$n'n^+ps^+$	$n^-ps^+s^-$	$n'fps^+$
		$n'n^-ps^-$	$n'n^+fs^+$	$n'fps^-$
		$nn'ps^-$	$nn'fs^+$	$n^+fs^+s^-$
		$nn^-s^+s^-$	$nn'fs^-$	$n^-fs^+s^-$
		$nn^+s^+s^-$	$nn'n^-p$	$n^+n^-s^+s^-$
			$nn'n^+p$	$n'ps^+s^-$
			$nn^+n^-s^-$	$n'fs^+s^-$
			$nn^+n^-s^+$	

Observing these experimental results of 4-compositions does not bring many insights, using this specific disjunctive composition. Maybe a more appropriate, nonlinear composition operator would be useful. We are investigating this question.

Nonetheless, observe that the pair s^+s^- never appears in class $\mathcal{F}$, which somehow confirms the low complexity of class $\mathcal{F}$, and the intrinsic complexity of mixing symmetric shifts.

It is interesting to notice how complexity can increase when more and more rules are added on top of each other, starting from a very simple rule.

Example 2
An example of increase in the complexity is rule 90 obtained from the null rule 8 combined with the basis rules 64, 2 and 16. The transition to chaos is obtained, in this case, through a gradual increase of the complexity.

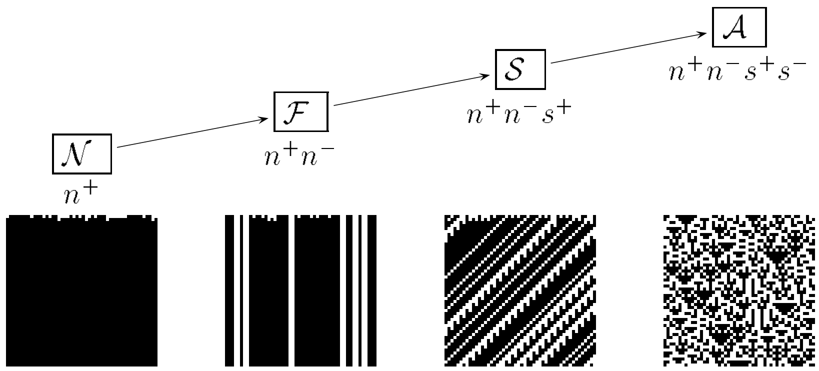

Figure 2: Increase in the complexity: rule $90 = 8 + 64 + 2 + 16 = n^+ + n^- + s^+ + s^-$

⊠

Of course, this increase of complexity is not commutative: the intermediate steps from one rule to another can vary a lot, as shown in the next example.

Example 3
Figure 3 shows two paths from rule s^+ leading to rule 202: the first one succesively adds 8, 64 and 128, passing through $\mathcal{S}_{\mathcal{F}}$, $\mathcal{S}_{\mathcal{F}}$ and $\mathcal{F}$; the second one adds 8, 128, 64, and gets through classes $\mathcal{S}_{\mathcal{F}}$, $\mathcal{S}_{\mathcal{F}}$ and $\mathcal{F}$. This figure also shows how, in both cases, the increase of complexity clearly happens when the symmetric null rules n^+ and n^- are present.
⊠

4.6 A new definition of complexity

The notion of chaos is still not well defined and understood in discrete-time discrete-space multi-dimensional dynamical systems like cellular automata although, recently, a considerable effort has been spent on trying to formalize the notion [13, 32, 33, 35, 36, 46, 54].

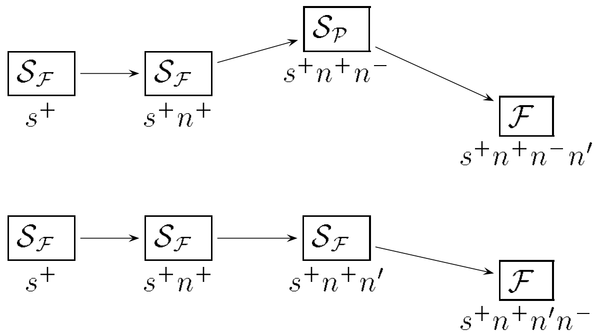

Figure 3: Different complexity paths leading to rule 202

As mentioned in the introduction, the asymptotic behavior of CA is easy to study and to understand only for a very limited set of CA due to the particular simple form they have; in general, the only way of studying their dynamics is by means of simulations, due to severe undecidability restrictions.

Based on the observations stated above where the presence of symmetric rules increases complexity, we now give a new definition.

Definition 5 (Structural complexity)
A system f is complex if it is composed of (at least) two subsystems whose dynamics are symmetric.

This definition holds for a large part of class $\mathcal{A}$ CA. It involves the components of a system to be characterized, which strongly differs from other well-known definitions based on mathematical properties or physical measures.

Indeed, mathematical properties concern the dynamics of systems and involve metric and topological properties of their attractors (e.g. Devaney's chaos is based on the following three properties: sensitivity to initial conditions, topological transitivity, density of periodic points [16]). Physical measures define observable parameters involving the evolution of (sets of) states during sufficiently long periods: for instance, Lyapunov exponents measure the way initially nearby states tend to diverge along their evolution, Perron-Frobenius operators describe how state densities evolve; very often, they cannot be analytically computed a priori and, contrarily, they have to be computed by simulation, a posteriori [5, 46].

Our definition of complexity does not involve any computation on potential evolutions of systems, nor any verification of mathematical properties of their attractors. The complexity we define is just based on the verification of structural properties of systems, together with some simple dynamical properties of their components.

5 Concluding remarks and research directions

The concern of this paper was the analysis of attraction properties of CA. To this end, we proposed to study these complex dynamical systems by an experimental compositional approach: given a system, decompose it to obtain individual subsystems; for each one of them, study its dynamical (viz. invariance and attraction) properties; finally, deduce the global properties by composition of the individual analyzes. In mathematical terms, f and g being two local transition functions, $\star$ (resp. $\diamond$) being a local (resp. global) operator, we searched for an algebraic expression of the form $\otimes(f \star g) = \otimes(f) \diamond \otimes(g)$, such that the dynamics of the $\diamond$-composition could be characterized.

Simulations linking local and global operations led to important observations. In particular, the experimental results showed that the disjunctive composition of symmetric rules (e.g. left and right shifts) generates complex orbits (for a more detailed analysis, see [22]). On the other hand, the exact influence of the elementary periodic rule p in the composition is still unclear, and this interesting question remains open.

One important question is whether these experimental results can be confirmed analytically. Indeed, some specific cases can be analyzed in a straightforward way, but the inherent distributed interactions of CA entail difficulties that are unsurmountable without adequate probabilistic and algrebraic tools [24]. Currently, we are investigating these complementary approaches: the definition of an information flow on the lattice of cells, characterizing another kind of attraction; the topologies induced by other metrics; the local structure theory defined on probabilistic CA.

We have considered homogenenous CA: a very interesting open problem is to use the compositional approach to analyze heterogeneous, or *hybrid*, models (i.e., using possibly different local functions at different cells of the automata) [10, 11]. Some research in this direction has already started [20].

Admittedly, our case-study remains academic, and composition-in-the-large is far from being solved. In this paper, we proposed research directions which should be seen as challenges and benchmarks for the future of compositional analysis and synthesis of systems. By this work, we hope that both positive and negative results could emerge. Indeed, it is necessary to validate or invalidate ideas and techniques, and to evaluate the effective usefulness of composition regarding real applications.

Acknowledgements. This research has been partially supported by an international *CNR* fellowship, by the Belgian *Fonds National de la Recherche Scientifique*, and by *NSERC* under grant # A2415.

References

[1] M. Abadi and L. Lamport. Composing specifications. *ACM Trans. on Programming Languages and Systems*, 15:73–132, 1993.

[2] Y. Aizawa and I. Nishikawa. Towards the classification ot the patterns generated by one-dimensional cellular automata. In G. Ikegami, editor, *Chaotic Dynamical Systems*, pages 210–222. World Scientific, 1986.

[3] E. Akin. *The General Topology of Dynamical Systems.* American Mathematical Society, 1993.

[4] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P. H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138(1):3–34, 1995.

[5] C. Beck and F. Schlögl. *Thermodynamics of Chaotic Systems*, volume 4 of *Cambridge Nonlinear Science Series.* Cambridge University Press, 1993.

[6] G. Braga, G. Cattaneo, P. Flocchini, and G. Mauri. Complex chaotic behavior of a class of subshift cellular automata. *Complex Systems*, 7:269–296, 1993.

[7] J. L. Casti. *Reality Rules: Picturing the World in Mathematics (2 vol.).* John Wiley & Sons, 1992.

[8] G. Cattaneo, P. Flocchini, G. Mauri, and N. Santoro. Chaos and subshift rules in neural networks and cellular automata. In *Proc. International Symposium on Nonlinear Theory and its Applications, Hawaii*, volume 4, pages 1153–1156. IEICE, 1993.

[9] G. Cattaneo, R. Nani, and G. Braga. A tool for the analysis of cellular automata. In J. Mazoyer, editor, *Workshop on Cellular Automata, 25-26 November 1991, Lyon, France*, pages 7–12. ESPRIT BRA WG 3166, ASMICS, 1991.

[10] K. Cattell and J. C. Muzio. Analysis of one-dimensional linear hybrid cellular automata over GF(q). *IEEE Transactions on Computers*, 45:782–792, 1996.

[11] K. Cattell and J. C. Muzio. Synthesis of one-dimensional linear hybrid cellular automata. *IEEE Transactions on Computer-Aided Design*, 15:325–335, 1996.

[12] S. Chakraborty, D. Roy Chowdhury, and P. Pal Chaudhuri. Theory and applications of non-group CA for synthesis of easily testable finite state machines. *IEEE Transactions on Computers*, 45(8):769–781, 1996.

[13] H. Chaté and P. Manneville. Criticality in cellular automata. *Physica D*, 45:122–135, 1990.

[14] K. Culik II, L. P. Hurd, and S. Yu. Computation theoretic aspects of cellular automata. *Physica D*, 45:357–378, 1990.

[15] K. Culik II, L. P. Hurd, and S. Yu. Formal languages and global cellular automaton behavior. *Physica D*, 45:396–403, 1990.

[16] R. L. Devaney. *An Introduction to Chaotic Dynamical Systems.* Addison-Wesley, 2nd edition, 1989.

[17] V. Diekert and G. Rozenberg, editors. *The Book of Traces.* World Scientific, 1995.

[18] E. W. Dijkstra. *A Discipline of Programming.* Prentice Hall, 1976.

[19] E. W. Dijkstra and C. S. Scholten. *Predicate Calculus and Program Semantics.* Springer-Verlag, 1990.

[20] P. Flocchini and F. Geurts. Searching for chaos in cellular automata: Compositional approach. In R. J. Stonier and X. H. Yu, editors, *Complex Systems, Mechanism of Adaptation*, pages 329–336. IOS Press, 1994.

[21] P. Flocchini and F. Geurts. Searching for chaos in cellular automata: New tools for classification. In R. J. Stonier and X. H. Yu, editors, *Complex Systems, Mechanism of Adaptation*, pages 321–328. IOS Press, 1994.

[22] P. Flocchini and F. Geurts. Compositional complexity in cellular automata: a case study. Technical Report TR-96-29, School of Computer Science, Carleton University, 1996. http://www.scs.carleton.ca/scs/tech_reports/1996/list.html.

[23] M. Garzon. *Models of Massive Parallelism. Analysis of Cellular Automata and Neural Networks.* Springer-Verlag, 1995.

[24] F. Geurts. *Compositional Analysis of Iterated Relations: Dynamics and Computations.* Springer-Verlag, to appear.

[25] F. Geurts and P. Flocchini. Towards chaos in complex systems, attraction-based classification of cellular automata. Technical Report RR 95-3, Département INGI, U.c. Louvain, 1995.

[26] M. G. Gouda. The triumph and tribulation of system stabilization. In J. M. Hélary and M. Raynal, editors, *Proc. of the 9th International Workshop on Distributed Algorithms*, volume 972 of *LNCS*, pages 1–18. Springer-Verlag, 1995.

[27] P. Guan. Cellular automaton public-key cryptosystem. *Complex Systems*, 1:51–57, 1987.

[28] H. Gutowitz, editor. *Cellular Automata, Theory and Experiment.* MIT Press/North-Holland, 1991.

[29] H. A. Gutowitz. Cryptography with dynamical systems. In N. Boccara, E. Goles, S. Martinez, and P. Picco, editors, *Cellular Automata and Cooperative Systems*, volume 396 of *NATO ASI Ser. C: Math. Phys. Sci.*, pages 237–274. Kluwer Academic Publishers, 1993.

[30] C. A. R. Hoare. Some properties of predicate transformers. *Journal of the ACM*, 25(3):461–480, 1978.

[31] C. A. R. Hoare. *Communicating Sequential Processes.* Prentice Hall, 1985.

[32] E. Jen. Aperiodicity in one-dimensional cellular automata. *Physica D*, 45:3–18, 1990.

[33] K. Kaneko. Chaos as a source of complexity and diversity in evolution. Technical report, Dept. Pure and Appl. Sc., University of Tokyo, 1993.

[34] R. Kumar, V. Garg, and S. I. Marcus. Predicates and predicate transformers for supervisory control of discrete event dynamical systems. *IEEE Trans. Autom. Control*, 38(2):232–247, 1993.

[35] W. Li and N. H. Packard. The structure of the elementary cellular automata rule space. *Complex Systems*, 4:281–297, 1990.

[36] W. Li, N. H. Packard, and C. G. Langton. Transition phenomena in cellular automata rule space. *Physica D*, 45:77–94, 1990.

[37] B. Martin. *Construction Modulaire d'Automates Cellulaires.* PhD thesis, Laboratoire de l'Informatique du Parallélisme, E.N.S.Lyon; Université Claude Bernard Lyon 1, 1993.

[38] A. Mazurkiewicz. Trace theory. In W. Brauer, W. Reisig, and G. Rozenberg, editors, *Petri Nets: Applications and Relationships to Other Models of Concurrency*, volume 255 of *LNCS*, pages 279–324. Springer-Verlag, 1986.

[39] R. Milner. *Communication and Concurrency.* Prentice Hall, 1989.

[40] C. Moore. Unpredictability and undecidability in dynamical systems. *Physical Review Letters*, 64(20):2354–2357, 1990.

[41] S. Nandi, B. K. Kar, and P. Pal Chaudhuri. Theory and application of cellular automata in cryptography. *IEEE Transactions on Computers*, 43(12):1346–1357, 1994.

[42] A. Nerode and W. Kohn. Models of hybrid systems: Automata, topologies, controllability, observability. In R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, editors, *Hybrid Systems*, volume 736 of *LNCS*, pages 317–356. Springer-Verlag, 1993.

[43] T. T. Nguyen. A relational model of demonic nondeterministic programs. *International Journal of Foundations of Computer Science*, 2(2):101–131, 1991.

[44] P. Pal Chaudhuri, D. Roy Chowdhury, S. Nandi, and S. Chattopadhyay. *Theory and Applications in Additive Cellular Automata.* IEEE Press, 1997 (to appear).

[45] M. Phipps. From local to global: the lesson of cellular automata. In D. DeAngelis and L. Gross, editors, *Individual-Based Approaches in Ecology: Concepts and Models.* Chapman and Hall, 1992.

[46] D. A. Rand. Measuring and characterizing spatial patterns, dynamics and chaos in spatially extended dynamical systems and ecologies. *Phil. Trans. R. Soc. Lond. A*, 348:497–514, 1994.

[47] M. Schneider. Self-stabilization. *ACM Computing Surveys*, 25(1):45–67, 1993.

[48] M. Serra, T. Slater, J. C. Muzio, and D. M. Miller. The analysis of one-dimensional linear cellular automata and their aliasing properties. *IEEE Transactions on Computer-Aided Design*, 9:767–778, 1990.

[49] M. Sintzoff. Abstract verification of structured dynamical systems. In *Proc. of Hybrid Systems III*, LNCS. Springer-Verlag, 1996.

[50] M. Sintzoff and F. Geurts. Compositional analysis of dynamical systems using predicate transformers (summary). In *Proc. International Symposium on Nonlinear Theory and its Applications, Hawaii*, volume 4, pages 1323–1326. IEICE, 1993.

[51] M. Sintzoff and F. Geurts. Analysis of dynamical systems using predicate transformers: Attraction and composition. In S. I. Andersson, editor, *Analysis of Dynamical and Cognitive Systems*, volume 888 of *LNCS*, pages 227–260. Springer-Verlag, 1995.

[52] B. Voorhees. *Computational Analysis of One-Dimensional Cellular Automata.* World Scientific, 1996.

[53] S. Wiggins. *Introduction to Applied Nonlinear Dynamical Systems and Chaos*, volume 2. Springer-Verlag, 1990.

[54] S. Wolfram. *Theory and Applications of Cellular Automata.* World Scientific, 1986.