Chapter 5
Symmetry and Complexity of Cellular Automata: Towards an Analytical Theory of Dynamical System

Klaus Mainzer and Carl von Linde-Akademie

Technische Universität München, München, Germany
mainzer@cvl-a.tum.de

Stephen Wolfram declared computer experiments with pattern formation of cellular automata as "new kind of science" (NKS). It is obviously a great merit of NKS to highlight the experimental approach in the computational sciences [26]. But we claim that even in the future quasi-empirical computer experiments are not sufficient [12]. Cellular automata must be considered complex dynamical systems in the strictly mathematical sense with corresponding equations and proofs. In short, we also need analytical models of cellular automata, in order to find precise answers and predictions in the universe of cellular automata. In this sense, our approach goes beyond Wolfram's NKS.

In our approach cellular automata (CA) are defined as complex dynamical systems. The geometrical representation of the eight CA-rules as a Boolean cube allows precise definitions of a complexity index and universal symmetries. It can be proved that the 256 one-dimensional cellular automata are classified by local and global symmetry classes of cellular automata. There is an exceptional symmetry group with universal computability which we call the "holy grail" in the universe of cellular automata. Although the four automata of this group are completely deterministic, their long-term behavior cannot be predicted in principle with respect to the undecidability of Turing's halting problem. Many analytical concepts of complexity research (e.g., attractors, basin of attractors, time series, power spectrum, fractality) are defined for cellular automata. But there are also surprising phenomena in the CA-world (isles of Eden) without analytical representation in dynamical systems.

1 Dynamics in the Universe of Cellular Automata

Because of their simplicity, rules of cellular automata can easily be understood. In a most simple version, we consider *two-state one-dimensional cellular automata* (CA) made of identical cells with a *periodic boundary condition*. In this case, the object of study is a ring of coupled cells with $L = I+1$ cells, labeled consecutively from $i = 0$ to $i = I$ (Fig. 1(a)). Each cell i has two *states* $u_i \in \{0, 1\}$, which are coded by the colors blue and red, respectively. A clock sets the pace in discrete times by iterations or generations. The state u_i^{t+1} of all i at time $t + 1$ (i.e. the

H. Zenil (Ed.): *Irreducibility and Computational Equivalence*, ECC 2, pp. 47–65.
DOI: 10.1007/978-3-642-35482-3_5

next generation) is determined by the states of its nearest neighbors u_{i-1}^t, u_{i+1}^t, and itself u_i^t at time t (Fig. 1(c)), i.e. by a *Boolean function* $u_i^{t+1} = N(u_{i-1}^t, u_i^t, u_{i+1}^t)$, in accordance with a prescribed *Boolean truth table* of $8 = 2^3$ distinct 3-input patterns (Fig. 1(d)).

1.1 From Simple Local Rules to Global Complex Patterns

These eight 3-input patterns can nicely be mapped into the eight vertices of a toy cube (Fig. 1(b)), henceforth called a *Boolean cube* [3]. The output of each prescribed 3-input pattern is mapped onto the corresponding colors (red for 1, blue for 0) at the vertices of the Boolean cube (in Fig. 1(d) yet unspecified). Since there are $2^8 = 256$ distinct combinations of eight bits, there are exactly 256 Boolean cubes with distinct vertex color combinations. Thus, we get a gallery of picturesque toy cubes.

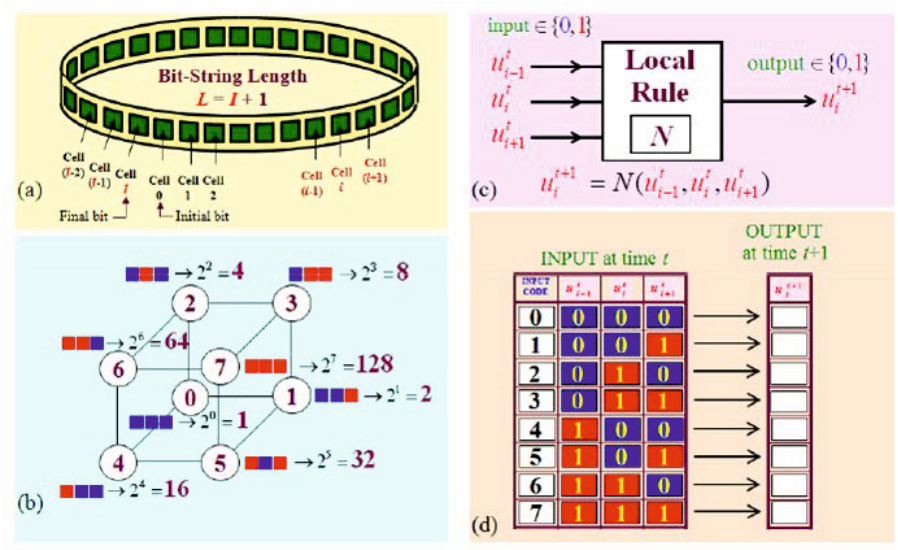

Fig. 1. Scheme of a two-state one-dimensional Cellular Automaton with local rule N

It is convenient to associate the 8-bit patterns of each Boolean function with a decimal number N representing the corresponding 8-bit word, namely $N = \beta_7 \cdot 2^7 + \beta_6 \cdot 2^6 + \beta_5 \cdot 2^5 + \beta_4 \cdot 2^4 + \beta_3 \cdot 2^3 + \beta_2 \cdot 2^2 + \beta_1 \cdot 2^1 + \beta_0 \cdot 2^0$ with $\beta \in \{0,1\}$. Notice that since $\beta_i = 0$ for each blue vertex in Fig. 1(b), N is simply obtained by adding the weights (indicated next to each pattern in Fig. 1(b)) associated with all red vertices. For example, for the Boolean cube shown in Fig. 2(b), we have $N = 0{\cdot}2^7{+}1{\cdot}2^6{+}1{\cdot}2^5{+}0{\cdot}2^4{+}1{\cdot}2^3{+}1{\cdot}2^2{+}1{\cdot}2^1{+}1{\cdot}2^0 = 2^6{+}2^5{+}2^3{+}2^2{+}2^1 = 110$.

For the example of local rule 110, the ring and the colored vertices of the Boolean cube are shown in Fig. 2(a)-(b). Given any initial binary bit-configuration at time $t = 0$, the local rule N is used to update the state u_i^{t+1} of each cell i at time $t + 1$, using the states of the three neighboring cells $i - 1$,

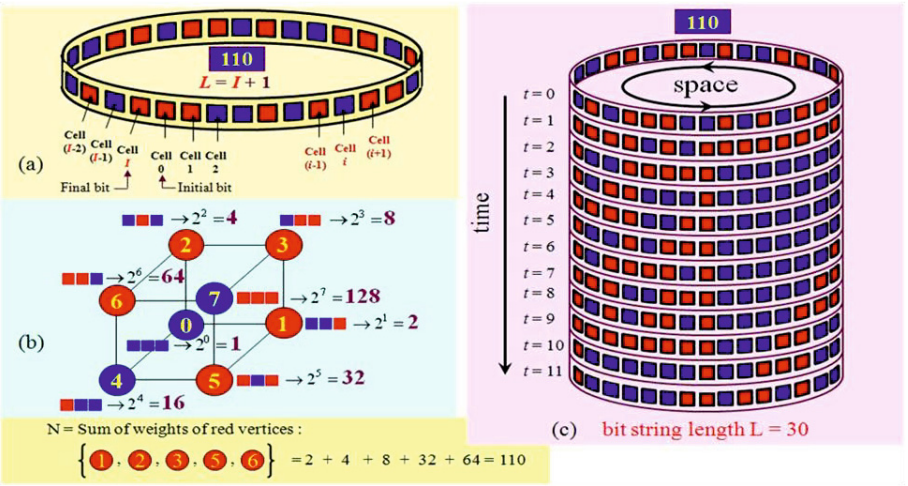

Fig. 2. Example of local rule 110

i, and $i+1$, centered at location i, respectively. The *space-time pattern* for the initial state is shown in Fig. 2(c) for $t = 0, 1, 2, \ldots, 11$.

In principle, one can draw and paint the patterns of cellular automata following these rules step by step. Modern computers with high speed and capacity allow extensive computer experiments to study pattern formations of these automata. Stephen Wolfram discovered remarkable analogies with patterns in physics and biology [26]. In the world of cellular automata many phenomena of the physical world seem to evolve. Some automata generate symmetric patterns reminding us of the coloring in sea shells, skins or feathers. Other automata reproduce rhythms like oscillating waves. Some of these automata stop their development after a finite number of steps, independent of their initial state, and remain in a constant color state like a system reaching at an equilibrium state for all future steps. Some automata develop complex patters reminding us of the growth of corals or plants, depending sensitively on tiny changes of the initial states. This phenomenon is well-known as the butterfly-effect, when local events lead to global effects in chaotic and unstable situations (e.g., weather and climate). Even these chaotic patterns can be generated by cellular automata.

One can try to classify these patterns with respect to their outward appearance like zoologists and botanists distinguishing birds and plants in taxonomies. But sometimes, outward features are misleading. Fundamental question arises: Are there laws of complex pattern formation for cellular automata like in nature? Can the development of complex patterns be predicted in a mathematically rigorous way like in physics? We argue for a mathematically precise explanation of the dynamics in cellular automata. Therefore, they must also be characterized by complex dynamical systems determined with differential equations like in physics. This is, of course, beyond the scope of elementary rules of toy worlds. But, we should keep this perspective in mind.

1.2 Cellular Automata as Dynamical Systems

For maximum generality, each cell i is assumed to be a *dynamical system* with an intrinsic state x_i, an output y_i and *three inputs* u_{i-1}, u_i, and u_{i+1} where u_{i-1} denotes the input coming from the left neighboring cell $i-1$, u_i denotes the self input of cell i , and u_{i+1} denotes the input coming from the right neighboring cell $i+1$ in the ring of Fig. 1(a). Each cell evolves with its prescribed dynamics and its own time scale. When coupled together, the system evolves consistently with its own rule as well as the rule of interaction imposed by the coupling laws.

Each *input* is assumed to be a *constant integer* $u_i \in \{-1, 1\}$, and the *output* y_i *converges* to a *constant* either -1 or 1 from a zero initial condition $x_i(0) = 0$. Actually, it takes a finite amount of time for any dynamical system to converge to an *attractor*. But, for the purpose of idealized cellular automata, each attractor is assumed to be reached instantaneously. Under this assumption and with respect to the *binary input* and *output*, our *dynamical system* can be defined by a *nonlinear map* which is uniquely described by a *truth table* of three input variables (u_{i-1}, u_i, u_{i+1}). The choice of $\{-1, 1\}$ and not $\{0, 1\}$ as binary signals is crucial, because the state x_i and output y_i evolves in *real time* via a carefully designed *scalar ordinary differential equation*. According to this differential equation, the output y_i which is defined via an output equation $y_i = y(x_i)$ tends to either 1 or -1 after the solution x_i (with zero initial state $x_i(0) = 0$) reaches a *steady state*. In this way, the *attractors* of the *dynamical system* can be used to encode a *binary truth table*.

Aside from the *cell's intrinsic time scale* (which is of no concern in cellular automata), an external clocking mechanism is introduced to reset the input u_i of each cell i at the end of each clock cycle by feeding back the steady state *output* $y_i \in \{-1, 1\}$ as an updated *input* $u_i \in \{-1, 1\}$ for the *next iteration*. This mechanism corresponds to the *periodic boundary condition* of a one-dimensional cellular automaton in Fig. 1(a).

Although cellular automata are concerned only with the ring's evolutions over *discrete times*, any computer used to simulate cellular automata is always a *continuous time system* with very small but non-zero time scale. Computers use transistors as devices, and each cellular automata iteration involves the physical evolution of millions of transistors with its own $u_i \in \{-1, 1\}$ intrinsic dynamics. These transistors evolve in accordance with a large system of nonlinear differential equations governing the entire internal computer circuit and return the desired output after converging to their attractors in a non-zero amount of time.

These considerations lead us to the important result that, even in *discrete systems* like cellular automata, there are *two different time scales* involved. The first one applies to the rule N while the second applies to the global patterns of evolution. In order to understand the complex dynamics of global patterns, it is necessary to analyze both times scales. By unfolding the truth tables of cellular automata into an appropriate nonlinear dynamical system, we can exploit the theory of *nonlinear differential equations* to arrive at phenomena based on a precise mathematical theory, and not only on empirical observations.

For this purpose, we substituted the binary symbol 0 by the -1, and the input and output values 0 and 1 in the truth table of Fig. 1(d) by the real numbers -1 and 1, respectively. An advantage of working with the numeric rather than the symbolic truth table is the remarkable insights provided by the equivalent Boolean cube representation. Here, the eight vertices of the cube $(-1,-1,-1)$, $(-1,-1,1)$, $(-1,1,-1)$, $(-1,1,1)$, $(1,-1,-1)$, $(1,-1,1)$, $(1,1,-1)$ and $(1,1,1)$ are located exactly at the coordinates $(u_{i-1},\ u_i,\ u_{i+1})$ of a *coordinate system* with the origin located at the center of the cube. The vertex $n = 0, 1, 2, \ldots, 7$ corresponding to row n of the truth table is coded blue if the output is -1 , and red if the output is 1.

The choice of $\{-1, 1\}$ instead of $\{0, 1\}$ as binary signals is necessary, when the truth table is mapped onto a dynamical system where the states evolve in real time via an ordinary differential equation which is always based on the real number system. Each cell i is coupled only to its left neighbor cell $i-1$ and right neighbor cell $i+1$. As a dynamical system, each cell i has a *state variable* x_i, an *output variable* y_i, and three constant *binary inputs* u_{i-1}, u_i and u_{i+1} (Fig. 3).

Thus, the *dynamical system* is determined by a

$$\begin{aligned} \textit{state equation}\colon \dot{x_i} &= f(x_i, u_{i-1}, u_i, u_{i+1}) \\ x(0) &= 0 \text{ (initial condition)} \\ \textit{output equation}\colon y_i &= y(x_i) \end{aligned}$$

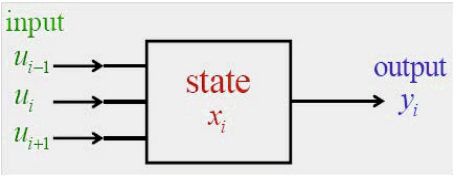

Fig. 3. Cell as dynamical system with state variable x_i, an output variable y_i, and three constant binary inputs u_{i-1}, u_i, and u_{i-1}

Every cellular automata can be mapped into a nonlinear dynamical system whose attractors encode precisely the associated truth table $N = 0, 1, 2, 3, \ldots, 255$. Function f models the *time-depend change of states* and is defined by a scalar ordinary differential equation of the form

$$\dot{x} = g(x_i) + w(u_{i-1}, u_i, u_{i+1}) \text{ with } g(x_i) \triangleq -x_i + |x_i + 1| - |x_i - 1| \,.$$

There are many possible choices of *nonlinear basis functions* for $g(x_i)$ and $w(u_{i-1}, u_i, u_{i+1})$. We have chosen the absolute value function $|x| = x$ for positive numbers x and $|x| = -x$ for negative numbers x as nonlinear basis function, because the resulting equation can be expressed in an optimally compact form, and it allows us to derive the solution of the state equation in an explicit form. The scalar function $w(u_{i-1},\ u_i,\ u_{i+1})$ can be chosen to be a composite function w(σ) of a single variable $\sigma \triangleq b_1\ u_{i-1} + b_2\ u_i + b_3\ u_{i+1}$ with $w(\sigma) \triangleq \{z_2 \pm |[z_1 \pm$

$|z_o + \sigma|]|\}$. This function is used to define the appropriate differential equation for generating the truth table of all 256 Boolean cubes. Thus, each rule of a cellular automaton corresponds to a particular set of *six real numbers* $\{ z_o, z_1, z_2; b_1, b_2, b_3\}$, and *two integers* ± 1. Only eight bits are needed to uniquely specify the differential equation associated with each rule N of a cellular automaton.

It can be proven that once the parameters defining a particular rule N are specified, then for any one of the eight inputs u_{i-1}, u_i, and u_{i+1} listed in the corresponding truth table of N, the solution x_i of the scalar differential equation will either increase monotonically from the initial state $x_i = 0$ towards a *positive equilibrium value* $\bar{x_i}(n) \geq 1$, henceforth denoted by attractor $Q_+(n)$, or decrease monotonically towards a *negative equilibrium state* $\bar{x_i}(n) \leq -1$, henceforth denoted by attractor $Q_-(n)$, when the input (u_{i-1}, u_i, u_{i+1}) is chosen from the coordinates of vertex n of the associated Boolean cube, or equivalently, from row n of the corresponding truth table, for $n = 0, 1, 2, \ldots, 7$ [3]. Vertex n is painted red whenever its equilibrium value $\bar{x_i}(n) \geq 1$, and blue whenever $\bar{x_i}(n) \leq -1$, then the color of all eight vertices for the associated *Boolean cube* will be uniquely specified by the *equilibrium solutions* of the eight associated *differential equations*.

In general, we can summarize: once the parameters associated with a particular rule of a cellular automaton are specified, the corresponding *truth table* or *Boolean cube*, will be uniquely generated by the *scalar differential equation* alone. If the output equation of the dynamical system is $y_i = y(x_i) \triangleq \frac{1}{2}\,(|x_i + 1| - |\, x_i - 1|)$, then $y_i = +1$ when $x_i \geq 1$, and $y_i = -1$ when $x_i \leq -1$. The steady-state output at equilibrium is given explicitly by the formula $y_i = \text{sgn}\,\{w(\sigma)\}$ for any function $w(\sigma) \triangleq w(u_{i-1}, u_i, u_{i+1})$ with signum function $\text{sgn}\,(x) = +1$ for positive numbers x, $\text{sgn}\,(x) = -1$ for negative numbers x and $\text{sgn}\,(0) = 0$.

For the particular $w(\sigma)$ in Fig. 4 the output (color) at equilibrium is given explicitly by the

$$\textit{attractor color code}\text{: } y_i = sgn\,\{z_2 \pm |[z_1 \pm |z_0 + \sigma|]|\}\,.$$

Fig. 4 contains 4 examples of dynamical systems and the rules they encode, each one identified by its rule number $N = 0, 1, 2, \ldots, 255$. The truth table for each rule N is generated by the associated dynamical system defined in upper portion of each quadrant, and not from the truth table, thereby proving that each dynamical system and the rule of the cellular automaton it encodes are one and the same. The truth table for each rule in Fig. 4 is cast in a format with only $2^{2^3} = 256$ distinct 1x3 neighborhood patterns. Each color picture consists of 30 x 61 pixels, generated by a 1-dimensional cellular automaton with 61 cells and a boundary condition with a specific rule N.

As an example, let us examine one of the rules from Fig. 4, rule 110, which will later on be identified as the simplest universal Turing machine known to date. With its differential equation, one can identify $\sigma = b_1\ u_{i-1} + b_2\ u_i + b_3\ u_{i+1}$ with $b_1 = 1$, $b_2 = 2$, and $b_3 = -3$, and $w(\sigma) \triangleq \{z_2 \pm |[z_1 \pm |z_o + \sigma|]|\}$ with $z_2 = -2$, $z_1 = 0$, and $z_o = -1$. Thus, the attractor color code is explicitly given by $y_i = \text{sgn}[-2 + |u_{i-1} + 2u_i - 3u_{i+1} - 1)|]$.

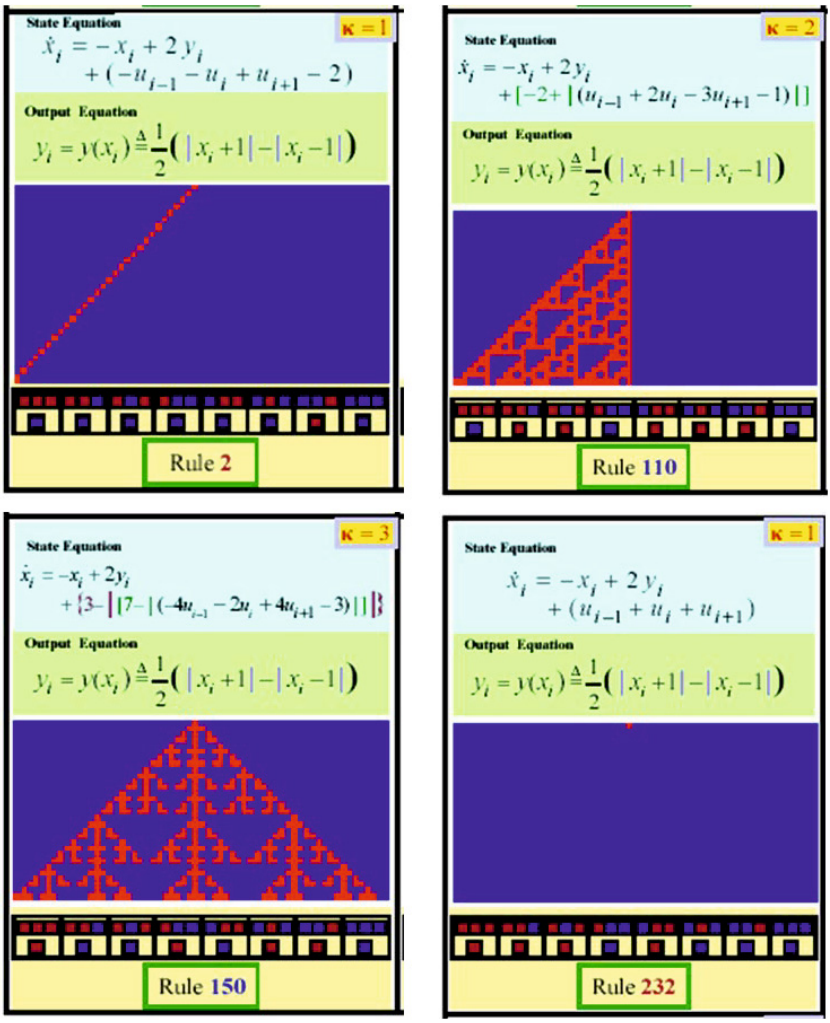

Fig. 4. Cellular automata with rules 2, 110, 150, 232 as dynamical systems. The initial condition is $x(0) = 0$.

1.3 Digital Dynamics with Difference Equations

The dynamics of dynamical systems are modeled with continuous differential equations. For computing the dynamics for digital cellular automata, a program must use a "do loop" instruction which feedbacks the output y_i^t of each cell at iteration t back to its inputs to obtain the output y_i^{t+1} at the next iteration $t+1$. Using the superscripts t and $t+1$ as iteration number from one to the next generation, we can express each rule N explicitly in the form of a *nonlinear difference equation* with

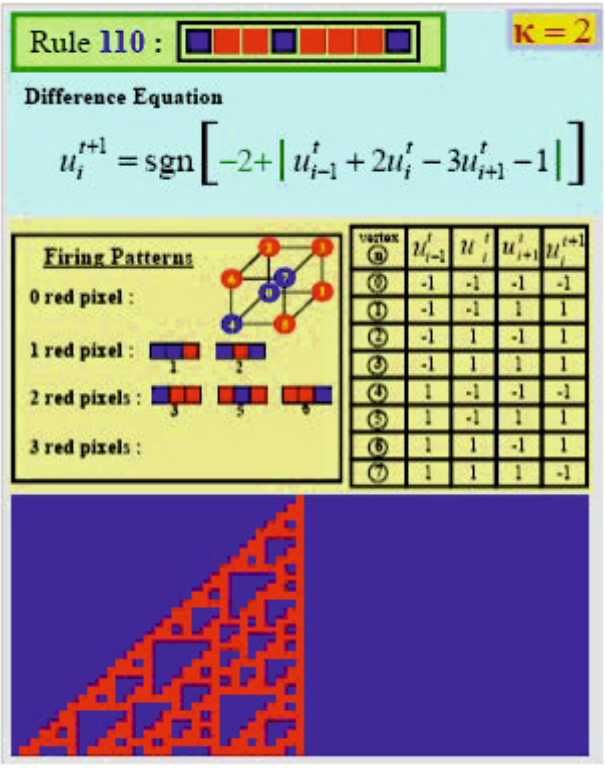

Fig. 5. Cellular automaton as dynamical system with difference equation

$$u_i^{t+1} = sgn\{z_2 + c_2|[z_1 + c_1|(z_0 + b_1 u_{i-1}^t + b_2 u_i^t + b_3 u_{i+1}^t)|]|\},$$

where the *eight parameters* $\{z_o, z_1, z_2; b_1, b_2, b_3; c_1, c_2\}$ are given for each rule. Thus, the first main result is that each of 256 1-dimensional cellular automata which were studied by Stephen Wolfram experimentally can be generated from a single scalar nonlinear differential equation or a corresponding nonlinear difference equation with at most eight parameters. These equation are also universal in the sense of a universal Turing machine (UTM), because we will later on see that at least one of the 256 rules (for example, rule 110) is capable of universal computation [4]. For rule 110 (Fig. 5), we get $u_i^{t+1} = \mathrm{sgn}\ (-2+|u_{i-1}^t + 2\ u_i^t - 3\ u_{i+1}^t - 1|)$. This kind of difference equation can be understood with elementary knowledge in basic mathematics, although it demonstrates important features of nonlinear dynamics.

2 Complexity in the Universe of Cellular Automata

The colored toy cubes contain all information about the complex dynamics of cellular automata. An important advantage of the Boolean cube representation

is that it allows us to define an index of complexity [3]. Each one of the 256 cubes is obviously characterized by different clusters of red or blues vertices which can be separated by parallel planes. On the other hand, the separating planes can be analytically defined in the coordinate system of the Boolean cubes. Therefore, the complexity index of a cellular automaton with local rule N is defined by the minimum number of parallel planes needed to separate the red vertices of the corresponding Boolean cube N from the blue vertices. Fig. 6 shows three examples of Boolean cubes for the three possible complexity indices $\kappa = 1, 2, 3$ with one, two and three separating parallel planes. There are 104 local rules with complexity index $\kappa = 1$. Similarly, there are 126 local rules with complexity index $\kappa = 2$ and only 26 local rules with complexity index $\kappa = 3$. This analytically defined complexity index is to be distinguished from Wolfram's complexity index based on phenomenological estimations of pattern formation.

2.1 Complexity Index of Cellular Automata

In the context of colored cubes of cellular automata, *separability* refers to the number of cutting (parallel) planes separating the vertices into clusters of the same color. For rule 110, for example, we can introduce two separating parallel planes of the corresponding colored cube which are distinguished in Fig.6b by two different colors: The red vertices 2 and 6 lie above a yellow plane. The blue vertices 0, 4, and 7 lie between the yellow and a light blue plane. The red vertices 3, 1, and 5 lie below the light blue plane. It is well-known that the cellular automaton of rule 110 is one of the few types of the 256 automata which are *universal Turing machines*. In the sense of Wolfram's class 3 of computer experiments, it produces very complex patterns [26].

An example of an automaton which can only produce very simple patterns is rule 232. There is only *one separating plane* cutting the corresponding Boolean cube for separating colored points (Fig.6a): Red vertices 3, 5, 6, and 7 lie above a light blue plane. The blue vertices 0, 1, 2, and 4 lie below the light blue plane. A colored Boolean cube with *three parallel separating planes* is shown in Fig. 6c, representing the cellular automaton of rule 150: The blue vertex 6 lies above a green plane. The red vertices 2, 4, and 7 lie between a yellow plane and the green plane. The blue vertices 0, 3, and 5 lie between the yellow plane and a light blue plane. The blue vertex 1 lies below the light blue plane. Obviously, it is not possible to separate the 8 vertices into three colored clusters and at the same time separate them by two parallel planes, no matter how the planes are positioned.

A rule whose colored vertices can be separated by only one plane is said to be *linearly separable*. An examination of the 256 Boolean cubes shows that 104 among them are linearly separable. The remaining 152 rules are not linearly separable. Each rule can be separated by various numbers of parallel planes. In general, there is a unique integer κ, henceforth called the *complexity index* of rule N, which characterizes the geometrical structure of the corresponding Boolean cube, namely the minimum number of parallel planes that is necessary to separate the colored vertices. All linearly separable rules have a *complexity*

index κ=1. An analysis of the remaining 152 linearly non-separable rules shows that they have a complexity index of either 2 or 3. For example, rule 110 has a complexity index κ=2 whereas rule 150 has a complexity index κ=3. No rule with complexity index κ=1 is capable for generating complex patterns, even for random initial conditions. The emergence of complex phenomena significantly depends on a minimum complexity of κ=2. In this sense, complexity index 2 can be considered the threshold of complexity for 1-dimensional cellular automata.

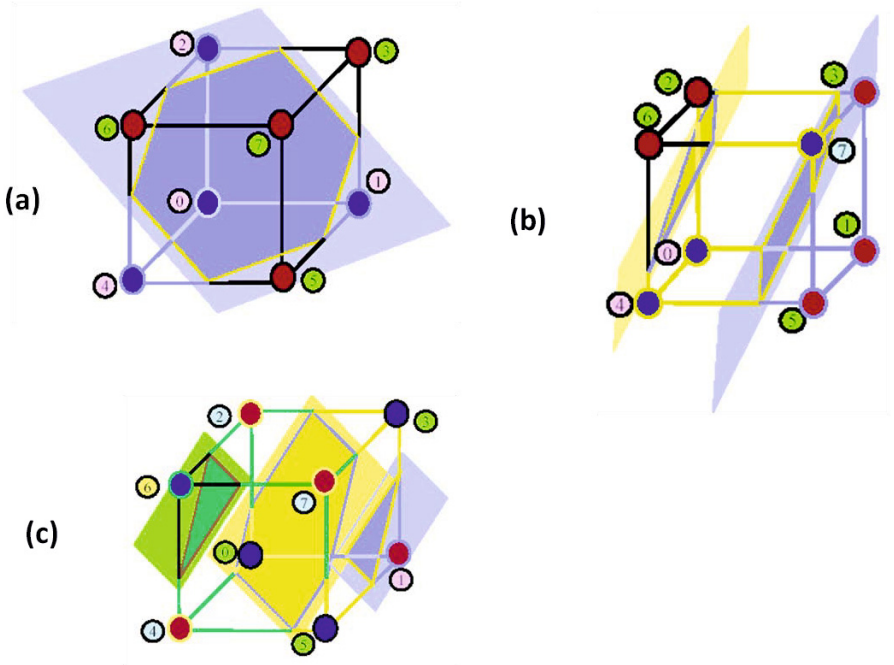

Fig. 6. Examples of complexity index $\kappa = 1, 2, 3$ with parallel planes separating all vertices having one color from those having a different color on the other side for rule 232 (**a**), rule 110 (**b**),and rule 150 (**c**)

2.2 Computational Complexity and Universal Computability

A motivation for the introduction of a complexity index is also *computational complexity*. The class of cellular automata with complexity index $\kappa = 2$ contains examples with universal computation (e.g., $N = 110$), but the local rules with complexity index $\kappa = 1$ are not capable of universal computation. It follows that $\kappa = 2$ also represents a threshold of computational complexity.

Universal computation is a remarkable concept of computational complexity which dates back to Alan Turing's universal machine [17]. Universal cellular automata are well-known since Conway's game of life [13]. A *universal Turing machine* can by definition simulate any Turing machine. According to the

Church-Turing thesis, any algorithm or effective procedure can be realized by a Turing machine. Now Turing's famous Halting problem comes in. Following his proof, there is no algorithm which can decide for an arbitrary computer program and initial condition if it will stop or not in the long run. (A computer program cannot stop if it must follow a closed loop.) Consequently, for a system with universal computation (in the sense of a universal Turing machine), we cannot predict if it will stop in the long run or not. Assume that we were able to do that. Then, in the case of a universal Turing machine, we could also decide whether any Turing machine (which can be simulated by the universal machine) would stop or not. That is obviously a contradiction to Turing's result of the *Halting problem*. Thus, systems with universal computation are unpredictable.

Unpredictability is obviously a high degree of complexity. It is absolutely surprising that systems with simple rules of behavior like cellular automata lead to complex dynamics which is no longer predictable. We will be very curious to discover examples of these, in principle, unpredictable automata in nature.

3 Symmetry in the Universe of Cellular Automata

A cursory inspection of the discrete time evolutions of the 256 local rules reveals some similarity and partial symmetry among various evolved patterns. It reminds us of more or less random observations in the natural sciences demanding for unifying mathematical explanations with fundamental laws. The unifying theory of physics is based on the assumption of fundamental mathematical symmetries [8, 9]. According to this view, the variety and complexity of natural phenomena have evolved from some few principles of symmetry. They are the "*Holy Grail*" of the Universe which is sought by prominent scientists and research groups all over the world. For the universe of cellular automata, we found the fundamental symmetries in the gallery of Boolean cubes [5]. Thus, at least in the toy world of cellular automata, the importance of symmetry laws can easily be imagined and understood.

3.1 Local Equivalence of Cellular Automata

But, even in the universe of cellular automata, the situation is sophisticated. The Boolean cubes of many different pairs of local rules seem to be related by some *symmetry transformations*, such as complementation of the vertex colors (e.g., rules 145 and 110). Yet, their evolved patterns are so different that it is impossible to relate them. How do we make sense of all these observations? In the case of rule 145 and 110, the associated Boolean cubes are related by a "red - blue vertex transformation". It is denoted as *local complementation operation* $\mathbf{T}^C$, because complementation is locally restricted. Intuitively, one might expect that their respective evolved patterns must also be related by a global complementation operation. But the intuition turns out to be wrong in general, upon comparing the two evolved patterns (Fig. 7). It is only true in a local sense with respect to special iterations. For example, starting from the same initial pattern (single red

center pixel) in the first row, we find the output (first iteration) of rule 145 is in fact the complement of that of rule 110; namely, two blue pixels for 145 and two red pixels for 110 at corresponding locations to the left of center. All other pixels at corresponding locations are also complements of each other.

However, the next iteration (row 3) under rules 145 and 110 in Fig. 7 are not complements of each other. The reason is that unlike the initial input u_i^0, $i = 0, 1, 2, \ldots, n$, which are the same for both 145 and 110, the next input u_i^1, $i = 0, 1, 2, \ldots, n$ (for $t = 1$ in row 2) needed to find the next iteration (row 3) are different and there is no reason for the output u_i^2 (for $t = 2$) at corresponding locations to be the complement of each other. In these cases, a pair of local rules is equivalent only in a local sense with respect to "*local in iteration time*", and not local in the usual sense of a spatial neighborhood.

In general, we define

Local Equivalence: Two local rules N and N' are said to be *locally equivalent* under a transformation $\mathbf{T} : N \to N'$ iff the output u_i^1 of N after one iteration of any initial input pattern u_i^0 can be found by applying the transformed input $\mathbf{T}(u_i^0)$ to rule N' and then followed by applying the *inverse transformation* $\mathbf{T}^{-1} : N' \to N$ to u_i^1.

3.2 Global Equivalence of Cellular Automata

Global aspects can be observed in the evolved patterns for the rules 110, 137 (Fig. 7), 124, and 193 (Fig. 8). Despite the fact that the respective Boolean cubes of these three rules do not seem to be related in an obvious way, their output patterns are so precisely related that one could predict the evolved pattern over all times t of each local rule 110, 124, 137, and 193. For example, the evolved output pattern of rule 124 can be obtained by a reflection of that of 110 about the center line, namely a bilateral transformation. The output of rule 193 can be obtained by applying the complement of u_i^0 (i.e. blue center pixels amidst a red background) to rule 110 and then taking the complement of the evolved pattern from 110. The output of rule 137 can be obtained by repeating the above algorithm for 193, and then followed further by a reflection about the center line. It can be proved that these algorithms remain valid for all initial input patterns. This result is most remarkable because it allows us to *predict* the evolved patterns from arbitrary initial configurations of three rules over all iterations and not just for one iteration as in the case of local equivalence.

In general, we define

Global Equivalence: Two local rules N and N' are said to be *globally equivalent* under a transformation $\mathbf{T} : N \to N'$ iff the output x_i^t of N can be found, for any t, by applying the transformed input $\mathbf{T}(x_i^0)$ to local rule N' and then followed by applying the *inverse transformation* $\mathbf{T}^{-1} : N' \to N$ to x_i^1, for any $t = 1, 2, \ldots$.

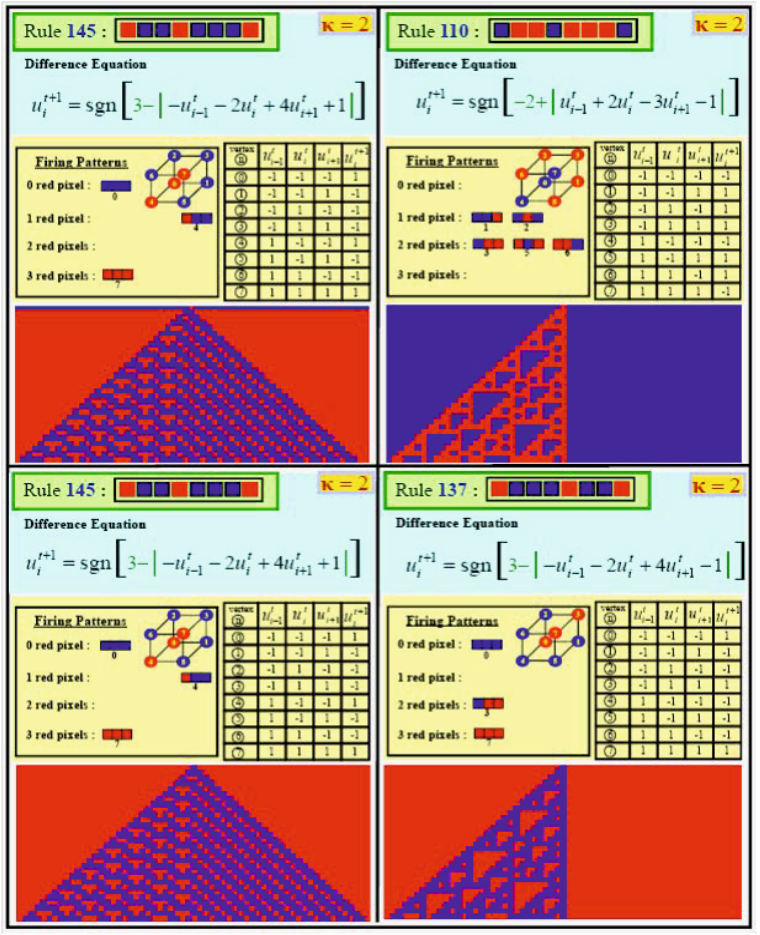

Fig. 7. The evolutions of rules 110 and 145 only reveal a local complement relationship in the first iteration, but 110 and 137 reveal global symmetrical relationship

Obviously, the four rules 110, 124, 137, and 193 are globally equivalent in the sense that the evolved patterns of any three members of this class can be trivially predicted from the fourth for all iterations. Therefore, these four rules have identical nonlinear dynamics for all initial input patterns and therefore they represent only one generic rule, henceforth called an *equivalence class*. This global property is not only true for four rules, but also for all rules, thereby allowing us to partition the 256 rules into only 88 global equivalence classes. It is convenient to identify these equivalence classes with the symbol ε_m^κ, where κ is the *complexity index* and m the *class number*. There are 38 cellular automata belonging to the equivalence classes ε_m^1 with complexity index $\kappa = 1$ and $m = 1, 2, \ldots, 38$. The equivalence classes ε_m^2 with complexity index $\kappa = 2$ are distinguished by $m = 1, 2, \ldots, 41$. Further on, there are nine global equivalence

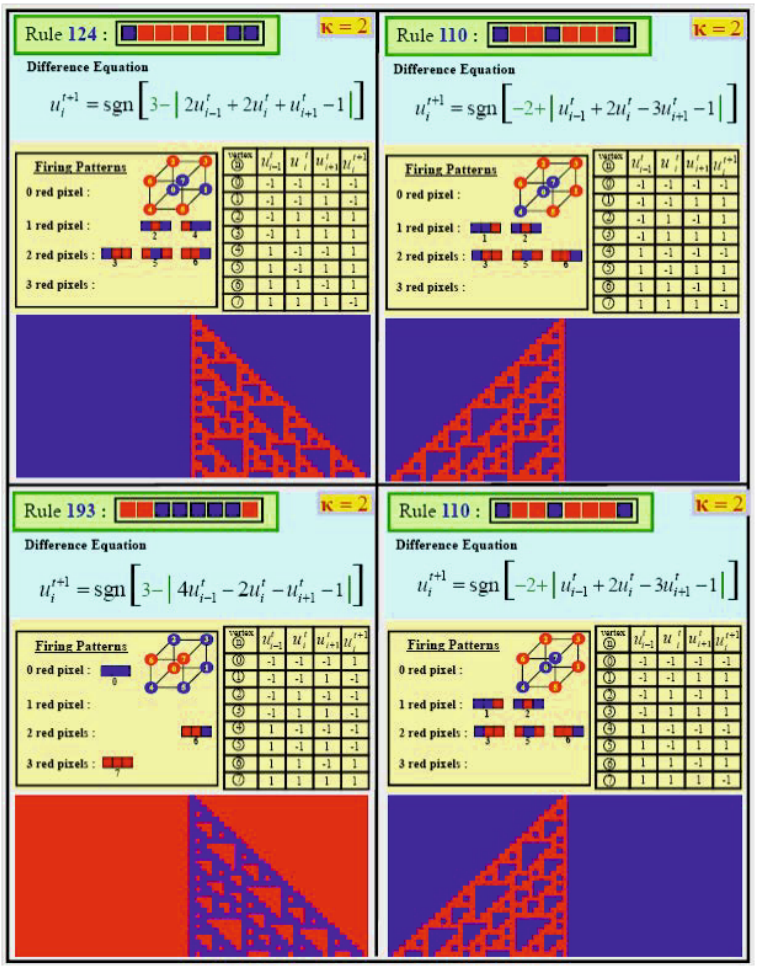

Fig. 8. The evolutions of rules 110, 124, 193 reveals global symmetrical relationships

classes with complexity index $\kappa = 3$. They are identified by ε_m^3 with $m = 1, 2, \ldots, 9$.

This result is significant because it asserts that one only needs to study in depth the dynamics and long-term behaviors of 88 representative local rules. Moreover, since 38 among these 88 dynamically distinct rules have complexity index $\kappa = 1$, and are therefore trivial, we are left with only 50 local rules (41 rules with $\kappa = 2$ and 9 rules with $\kappa = 3$) that justify further in-depth investigations.

3.3 Symmetry with Global Transformations

It can be proven that every local rule belongs to a global equivalence class determined by certain global transformations. There are three global transformations,

namely, *global complementation* $\overline{\mathbf{T}}$, *left-right complementation* $\mathbf{T}^{\star}$, and *left-right transformation* $\mathbf{T}^{\dagger}$ which are distinguished as *symmetry transformations* in the universe of cellular automata. The four rules 110, 124, 137, and 193 are globally equivalent to each other in the sense that their long term (as $t \to \infty$) dynamics are mathematically identical with respect to the three global transformations $\mathbf{T}^{\dagger}$, $\mathbf{T}^{\star}$, and $\overline{\mathbf{T}}$.

The intuitive meaning of these symmetry transformations can easily be seen in Fig. 9. In this picture, all four patterns of rules 110, 124, 137, and 193 have 60 rows corresponding to iterations numbers $t = 0, 1, 2, \ldots, 59$, and 61 columns, corresponding to 61 cells ($n = 60$). All patterns have a random initial condition ($t = 0$), or its reflection, complementation, or both. The two patterns 124 and 110 on top are generated by a left-right transformation $\mathbf{T}^{\dagger}$, and are related by a bilateral reflection about an imaginary vertical line situated midway between the two patterns. The two patterns 193 and 137 below are likewise related via $\mathbf{T}^{\dagger}$ and exhibit the same bilateral reflection symmetry. The two vertically situated local rules 137 and 110, as well as 193 and 124 are related by a global complementation $\overline{\mathbf{T}}$. The two diagonally-situated local rules 124 and 137, as well as 193 and 110 are related by a left-right complementation $\mathbf{T}^{\star}$.

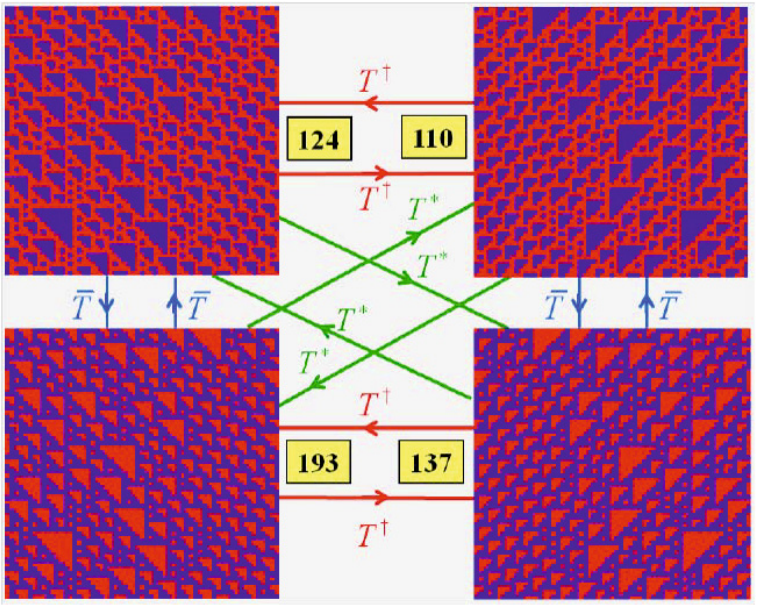

Fig. 9. Global equivalence of rules 110, 124, 137, and 193

The *geometrical definition* of these symmetry transformations is easy to understand and can even be imagined with help of our toy cubes of cellular automata. Mathematically, these transformations are defined by 3 x 3 *matrices* $\overline{\mathbf{T}}_{\mathbf{u}}$, $\mathbf{T}^{\star}_{\mathbf{u}}$, and $\mathbf{T}^{\dagger}_{\mathbf{u}}$. Each of the three matrices transforms the three axes (u_{i-1},

u_i, u_{i+1}), drawn through the center of the *Boolean cube* into a transformed set of axes (u'_{i-1}, u'_i, u'_{i+1}). These matrix representations also only need basic mathematics. An analytical definition is given in [12].

3.4 Global Symmetry of Klein's Vierergruppe V

The three global transformations $\mathbf{T}^\dagger$, $\mathbf{T}^\star$, and $\overline{\mathbf{T}}$ are generated from elements of the classic *noncyclic four-element Abelian group* $\mathbf{V}$, originally called the "*Vierergruppe*" by the German mathematician Felix Klein [16]. The four elements of $\mathbf{V}$ are denoted by the 3 x 3 matrices $\mathbf{T}_0$, $\overline{\mathbf{T}}_\mathbf{u}$, $\mathbf{T}^\star_\mathbf{u}$, and $\mathbf{T}^\dagger_\mathbf{u}$. The symbol $\mathbf{T}_0$ denotes the identity, or unit matrix, of any dimension. The actual transformations, however, that allow us to establish the long-term correlations among members of each of the 88 *global equivalence classes* of all 256 cellular automata are the 4 x 4 matrices $\mathbf{T}_0$, $\mathbf{T}^\dagger$, $\mathbf{T}^\star$, and $\overline{\mathbf{T}}$. Fig. 10 shows that they are related by the group multiplication table of Klein's Vierergruppe $\mathbf{V}$. This is the only abstract mathematical group which makes it possible to predict the *long-term correlations* among all members of the four remarkable rules 110, 124, 137, and 193.

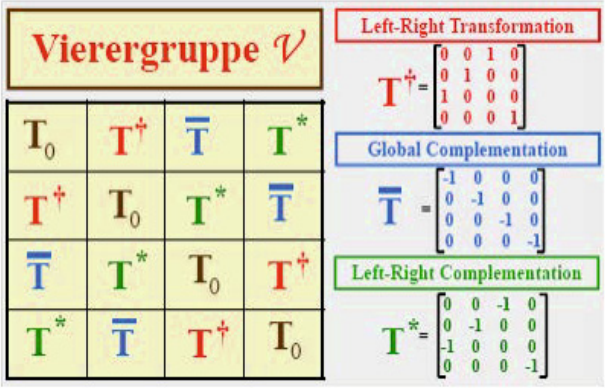

Fig. 10. Global Symmetry and Klein's Vierergruppe

These results are global in the sense of asymptotic time behavior as $t \to \infty$. It proves that even though there are 256 distinct local rules of 1-dimensional cellular automata, there are only 88 distinct global behaviors, a fundamental result predicted by the identification of 88 global equivalence classes ε^κ_m.

3.5 The Holy Grail of Symmetry and Computability

Since the local rule 110 has been proved to be capable of *universal computation*, it follows that all four local rules of the Vierergruppe $\mathbf{V}$ are *universal Turing machines*. The fundamental importance of the universality result was to exploit the *symmetry* of the Boolean cubes in order to identify equivalence classes among

the 256 rules. The discovery of the *Vierergruppe* **V** and the *rotation group* **R** had led to the major logical classifications of the 256 local rules into 88 *global equivalence classes* ε_m^κ and 30 *local equivalence classes* S_m^κ . The significance of the 88 global equivalence classes ε_m^κ is similar to the classification of computational algorithms into various complexity classes, for example, the *N*- or *NP*-classes, in the sense that any property that applies to one member of ε_m^κ applies to the other members in the same global equivalence class.

The *universality* of the four rules 110, 124, 137, and 193 and their identical long-term dynamic behaviors, with respect to the symmetry transformations of the Vierergruppe **V**, are encapsulated in the commutative diagram shown in Fig. 11. Thus, Klein's Vierergruppe represents the fundamental symmetry law of the 256 two-state one-dimensional cellular automata. It is the "*Holy Grail*" of a unified theory in the universe of these cellular automata, containing all information about their nonlinear dynamics.

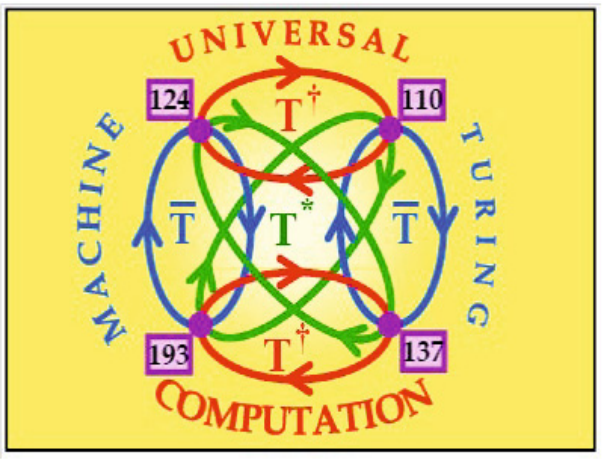

Fig. 11. Universal symmetry and computability in the universe of cellular automata

4 Outlook to a Computational Universe of Dynamical Systems

One-dimensional cellular automata with $L = I+1$ cells are *complex systems* with *nonlinear dynamics* [1, 11, 15] determined by one of the 256 local rules *N*. Their *state spaces* contain all distinct states of cellular rows $(x_0^t, \ldots, x_{I-1}^t, x_I^t)$ at step t of time (iteration or generation). An entire list of consecutive rows with no two rows identical and including the initial configuration is called an *orbit* in the state space of a cellular automaton. On that background, the well-known attractor dynamics of complex systems can also be studied in the theory of cellular automata [12].

Summing up all these insights, we are on the way to conceive the universe as an automaton and dynamical system. The success of this research program depends on the digitization of physics. The question "*Is the Universe a computer*" leads to the question: How far is it possible to map the laws of physics onto computational digital physics? [6] Digitization is not only exciting for answering philosophical questions of the universe. *Digitization* is the key paradigm of modern research and technology. Nearly all kind of research and technical innovation depend on computational modeling. The emerging complexity of nature and society cannot be handled without computers with increasing computational power and storage.

In order to make this complex computational world more understandable, cellular automata are an excellent tool. NKS and our analytical approach show that many basic principles of the expanding universe and the evolution of life and brain can be illustrated with cellular automata. The emergence of new structures and patterns depends on phase transitions of complex dynamical systems in the quantum, molecular, cellular, organic, ecological, and societal world [10]. Cellular automata are recognized as an intuitive modeling paradigm for complex systems with many useful applications [7]. In cellular automata, extremely *simple local interactions* of cells lead to the *emergence of complex global structures*. This *local principle of activity* is also true in the world of complex systems with elementary particles, atoms, molecules, cells, organs, organisms, populations, and societies [2]. Although local interactions generate a complex variety of being in the universe, they can be mathematically reduced to some fundamental laws of symmetry.

Symmetries play a key role in the physical world as well as in the universe of automata. In philosophy of science, they have been considered *universal principles* of *Platonic truth* and *beauty* [8]. The scientific search for symmetries reminds us of Parsifal's quest for the Holy Grail. The legend of Parsifal was written by the minnesinger Wolfram von Eschenbach (c. 1170–c. 1220). It may be a random accord of names that a "Wolfram" also wrote "A New Kind of Science" for cellular automata. In the 19^{th} century, Richard Wagner composed his famous opera based on Wolfram's legend of Parsifal. In Wagner's interpretation, it is the quest of the "poor fool" Parsifal for the Holy Grail. The question is still open whether the scientific search for a final symmetry or "world formula" will also be a "foolish" quest.

References

[1] Alligood, K.T., Sauer, T.D., Yorke, J.A.: Chaos: An Introduction to Dynamical Systems. Springer, New York (1996)
[2] Chua, L.O.: CNN: A Paradigm for Complexity. World Scientific, Singapore (1998)
[3] Chua, L.O., Yoon, S., Dogaru, R.: A nonlinear dynamics perspective of Wolfram's new kind of science. Part I: Threshold of complexity. International Journal of Bifurcation and Chaos (IJBC) 12(12), 2655–2766 (2002)
[4] Chua, L.O., Sbitnev, V.I., Yoon, S.: A nonlinear dynamics perspective of Wolfram's new kind of science. Part II: Universal neuron. International Journal of Bifurcation and Chaos (IJBC) 13(9), 2377–2491 (2003)

[5] Chua, L.O., Sbitnev, V.I., Yoon, S.: A nonlinear dynamics perspective of Wolfram's new kind of science. Part III: Predicting the unpredictable. International Journal of Bifurcation and Chaos (IJBC) 14, 3689–3820 (2004)
[6] Deutsch, D.: Quantum theory, the Church-Turing principle and the universal quantum computer. Proceedings of the Royal Society of London A 400, 97–117 (1985)
[7] Hoekstra, A.G., Kroc, J., Sloot, P.M.A. (eds.): Simulating Complex Systems by Cellular Automata. Springer, Berlin (2010)
[8] Mainzer, K.: Symmetries of Nature. De Gruyter, New York (1996) (German 1988: Symmetrien der Natur. De Gruyter, Berlin)
[9] Mainzer, K.: Symmetry and Complexity: The Spirit and Beauty of Nonlinear Science. World Scientific, Singapore (2005)
[10] Mainzer, K.: Thinking in Complexity. The Computational Dynamics of Matter, Mind, and Mankind, 5th edn. Springer, Berlin (2007)
[11] Mainzer, K. (ed.): Complexity. European Review (Academia Europaea) 17(2), 219–452 (2009)
[12] Mainzer, K., Chua, L.O.: The Universe as Automaton. From Simplicity and Symmetry to Complexity. Springer, Berlin (2011)
[13] Martin, B.: A universal cellular automaton in quasi-linear time and its S-m-n form. Theoretical Computer Science 123, 199–237 (1994)
[14] Rendell, P.: A Turing machine in Conway's Game of Life, extendable to a universal Turing machine. In: Adamatzky, A. (ed.) Collision-Based Computing. Springer, New York (2002)
[15] Shilnikov, L., Shilnikov, A., Turaev, D., Chua, L.: Methods of Qualitative Theory in Nonlinear Dynamics I-II. World Scientific, Singapore (1998-2001)
[16] Speiser, A.: Die Theorie der Gruppen von endlicher Ordnung, 4th edn. Birkhauser, Basel (1956)
[17] Turing, A.M.: On computable numbers with an application to the Entscheidungs problem. Proceeedings of the London Mathematical Society 2(42), corrections, ibid 43, 544–546 (1936-1937)
[18] Wolfram, S.: A New Kind of Science. Wolfram Media, Inc., Champaign Il (2002)

Figures (IJBC = International Journal of Bifurcation and Chaos)

Fig. 1: IJBC 2008 vol. 18, no. 9, p. 2490, Fig. 1a-d
Fig. 2: IJBC 2008 vol. 18, no. 9, p. 2496, Fig. 3a-d
Fig. 3: IJBC 2003 vol. 13, no. 9, p. 2378, Fig. 1b
Fig. 4: IJBC 2002 vol. 12, no. 12, Table 2, p. 2666 (rule 2), 2693 (rule 110), 2703 (rule 150), 2724 (rule 232)
Fig. 5: IJBC 2003 vol. 13, no. 9, p. 2417, Table 5 (rule 110)
Fig. 6a: IJBC 2002 vol. 12, no. 12, p. 2749, Fig. 14
Fig. 6b: IJBC 2002 vol. 12, no.12, p. 2742, Fig. 9
Fig. 6c: IJBC 2002 vol. 12, no. 12, p. 2746, Fig. 12
Fig. 7: IJBC 2004 vol. 14, no. 11, p. 3697, Fig. 5
Fig. 8 : IJBC 2004 vol. 14, no. 11, p. 3699, Fig. 6
Fig. 9 : IJBC 2004 vol. 14, no. 11, p. 3700, Fig. 7
Fig. 10: IJBC 2004 vol. 14, no. 11, p. 3818, Fig. 17a
Fig. 11: IJBC 2004 vol. 14, no. 11, p. 3818, Fig. 17b