

Review

# Emergent Information Processing: Observations, Experiments, and Future Directions

Jiří Kroc <sup>†</sup> 

Department of Informatics and Computers, Faculty of Science, University of Ostrava, 30. dubna 22, 70103 Ostrava, Czech Republic; dr.j.kroc@gmail.com

<sup>†</sup> Current address: Independent Researcher, 30614 Pilsen, Czech Republic.

**Abstract:** Science is currently becoming aware of the challenges in the understanding of the very root mechanisms of massively parallel computations that are observed in literally all scientific disciplines, ranging from cosmology to physics, chemistry, biochemistry, and biology. This leads us to the main motivation and simultaneously to the central thesis of this review: “Can we design artificial, massively parallel, self-organized, emergent, error-resilient computational environments?” The thesis is solely studied on cellular automata. Initially, an overview of the basic building blocks enabling us to reach this end goal is provided. Important information dealing with this topic is reviewed along with highly expressive animations generated by the open-source, Python, cellular automata software GoL-N24. A large number of simulations along with examples and counter-examples, finalized by a list of the future directions, are giving hints and partial answers to the main thesis. Together, these pose the crucial question of whether there is something deeper beyond the Turing machine theoretical description of massively parallel computing. The perspective, future directions, including applications in robotics and biology of this research, are discussed in the light of known information.

**Keywords:** emergent information processing; massively parallel computation; emergent logic; complex systems; cellular automaton; error-resilient; self-organization; biocomputing; turing machine equivalence



**Citation:** Kroc, J. Emergent

Information Processing: Observations, Experiments, and Future Directions.

*Software* **2024**, *3*, 81–106. <https://doi.org/10.3390/software3010005>

Academic Editors: Paulo Ferreira and Francisco José García-Peñalvo

Received: 11 August 2023

Revised: 31 December 2023

Accepted: 24 January 2024

Published: 5 March 2024



**Copyright:** © 2024 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Currently, science as a whole is facing an underlying challenge in the form of understanding emergent, error-resilient information-processing environments that have the potential to eventually become artificial analogs of natural phenomena observed within virtually all physical and biological systems. Such descriptions can become workhorses of future information processing devices.

This challenge should be decomposed into three distinct steps: the decoding, understanding, and design of artificial non-living and living systems based on massively parallel computations (MPCs), e.g., see documented examples in various systems and observed natural phenomena [1–4] and the following tables.

To describe the width and depth of the research field, the methods used there, and processes observed within complex systems, the following tables are presented: (i) Major classes of mathematical and computational methods, Table 1. (ii) More specialized huge databases and/or raw computational-power demanding methods, Table 2. (iii) The classification of major classes of processes operating within CSs; see Figure 1 and Table 3. (iv) Their biological applications, Table 4.

Due to the highly complicated and still poorly understood nature of emergence itself—and even more in its error-resilience against perturbations—this study is solely focused on cellular automata as the prototypical example and test-bed of all massively parallel computations. This enables us to focus on the foundations of emergence while simultaneously not being lost in a multitude of computational methods. The methods

are presented in Tables 1–4; together, they represent the road map of complex systems through all scientific disciplines. Additionally, tables provide links to other related research in physics, mathematics, computer science, biology, and medicine.



**Figure 1.** Complex systems contain immersed subsets that represent the following operational modes (going from the simplest): self-organization, emergence, replication, and self-replication. Life forms utilize them all within their different operational levels. Non-living systems utilize self-organization and emergence, too.

**Table 1.** A description of the major classes of mathematical and computational methods that are commonly used to study, quantify, and predict the behavior of complex systems.

Method Description	Examples and Links
Dynamical Systems (DSs) use ordinary (ODE) or partial differential equations (PDE) (continuous time) or difference equations (discrete time) to describe natural phenomena.	Recommended introduction [5], originally developed for the three-body problem by Poincaré [6–8]. Other very accessible introductory books requiring calculus [9–11]. Fractals are covered in [12].
Self-Organization (SO) and Emergence (Em) are found in the region between the regular/periodic behavior and chaos (i.e., at-the-edge of chaos) of many DS.	The SO regime is explained in synergetics from the physical point of view in [10,11]; SO-criticality is a subset of SO [13–15]. Emergence was initially coded manually [16–18].
Chaos is either deterministic or continuous and often arises from a predictably behaving system by a cascade of bifurcations.	Chaos [19] is often operating next to a self-organizing regime [10,11] and was discovered by Poincaré in 1890; see [6–8,20,21].
Cellular Automata (CAs) are defined above a lattice where each element evolves according to a local neighborhood in discrete steps.	The testbed for all massively parallel computations [22–24], recommended books [25,26], ‘Game of Life’ [27], and software [28,29].
Agent-Based Models (ABMs). A generalization of CAs in which independently deciding entities called agents freely move throughout the space while making decisions.	Examples of biologically observed agent systems: <i>D. discoideum</i> [30], stigmergy [31], and swarms [32]. Introduction [33,34], advanced [35,36], and agent-based software [34,37].
Complex Networks (CxN) are capturing observed dependencies and processes within all levels of living systems, from cells upwards.	CxN are divided into three major classes: random, small-world, and scale-free; introductory book [38], review [39], and books [40,41].

**Table 2.** A selection of more specialized methods, which require huge amounts of raw computer power, that are used to study complex systems. Those methods utilize either evolution (GA and GP) or huge databases (ANNs, ML, DL, and EnMEAs) to distill inherent hypotheses hidden within them. Their main deficiency is that they are not human-interpretable in the majority of cases (they are black boxes).

Method Description	Examples and Links
Machine Learning (ML) is based on the idea of automated hypothesis creation above data using analytical and statistical methods and other specialized algorithms.	Examples are regression, classification, decision trees, support vector machines, artificial neural networks, Bayesian networks, and Gaussian processes [42–45].
Deep Learning (DL) is a subset of ML and uses artificial neural networks with at least three layers of neurons that learn in unsupervised mode.	DL enables creation drive-less cars, design clever systems, etc. Two major learning methods exists: forward- and back-propagation; see [45–47] for details.
Genetic Algorithms (GAs) are evolutionary algorithms mimicking natural selection by using mutation and crossover of genotypes, leading to good solutions to a problem.	Used to global optimization problems, engineering, scheduling problems, antennae, aerodynamic bodies design, etc. [48,49]. Faces problems with premature convergence to local minima.
Genetic Programming (GP) uses evolutionary algorithms that search through the space of computer programs to solve a problem by utilizing the program population evolution.	Founded by Koza’s research on the evolution of algorithms [50–52]. Used to optimize (e.g., antennae), design electronic circuits, etc. Requires huge amounts of computational time for large problems.
Entropy Measures (EnMeas) represent a powerful tool for assessing the operational modes of CSs and MPCs by sampling their signals and features, which is not attainable by other means.	The concept was coined by Boltzmann in statistical physics [53,54], improved by Shannon on sending information [55], and advanced by [56–60]. Applications encompass [61–66].

**Table 3.** The classification of the major classes of CS processes, mathematical, and computational methods that are employed to study complex systems (citations included). The toolkit of every CS researcher should contain all of them, at least passively.

Processes Observed	Examples and Links
Massively Parallel Computations (MPCs) are indispensable in the description of physical, chemical, and biological systems. Their descriptive power is underestimated.	Cellular automata, agent-based models, dynamical systems, and dynamical networks (CAs are a subset of them); see Tables 1 and 2.
Self-Organization (SO). During SO, CSs are spontaneously evolving towards the preferred operational mode, irrespective of the initial conditions.	Non-Newtonian liquids [67], by genetic programming designed SO structures [18], SO-criticality [13–15], self-assembly [68], and stigmergy in insects (bees, wasps) [31].
Replication (Re) is the process that creates new entities within the given level of CSs. It possesses a unique place in the explanation of biochemical and cellular processes on many functional levels.	Replication was originally designed manually [16,17]. It later evolved using GP [18]. There are existing many replicators in CSs that are defined by CAs; see this review and [4].
Self-Replication (SRe) is the sub-class of replication that creates fully functional, identical copies of itself without external influences and interference, exactly like is observed living entities.	John von Neumann’s self-replicator (the seminal work) [16] and Langton’s minimal self-replicator [17] are the prominent examples. Self-replication in biology and chemistry [69–72].
Emergence (Em). A CS generates new entities along with its unique interaction rules at a higher-level of the system, utilizing the pre-existing, lower-level ones (emergent hierarchies observed). An example is the Game of Life.	Emergence is observed in ant-colonies [73], insect-colonies [31,74], flocks, cells [39–41], organs [75], bodies [2,76–78], societies, and ecosystems; see [4,74,79]. Emergent structures operating within MPC systems such as CAs, ABM, and CxN.

**Table 4.** Emergent information processing is abundant in biological systems, which simultaneously serve as the ultimate testbed for novel mathematical and computational approaches. The goal is to develop models describing the functioning of biological systems from the first principles.

Bio-System Process Description	Examples of Biological and Biology-Mimicking Systems
Morphology Growth leads to the end-goal system, which is defined by electric potential distribution on cell membranes.	Embriogenesis [80], morphological growth [75,81–83], rejuvenation and limb re-growth [71,80,82–84], and cancer treatment [85].
Bio-Chemistry. Abiotic compounds are assembled into biochemicals that produce other biochemicals.	Chemistry, bio-chemistry, self-replication [75,77], and light-induced bio-chemistry [86–88].
Replication is the process of novel species creation from those already existing: in biology and in silico.	It is utilized in the definition and simulation of logic gates, processing units [4,79,89,90], and ships breeding ships [79,90].
Self-replication represents a subset of replication where an entity creates identical copies, both in biology and in silico.	(Self-)Replication is utilized by all living entities (at their many scales simultaneously) to maintain their growth and long-term function [69–71,86].
Bio-Computing is an approach that deals with biological structures as the computing environment.	There are existing interpretations of life as the computing environment [76,81,82,84,91].
Emergent Information Processing is coined in this review as the concept describing multi-level information processing in living and non-living systems.	A hierarchy of bio-molecules, cellular building blocks, cells, tissues, organs, bodies, and societies are understood as emergents [this review] and [4,28,29,79,90].

### 1.1. Cellular Automata and Turing Machine

The initial motivation for the research directions presented in this review was provided by the capability of CAs to simulate the Turing machine; for details see forthcoming Section 1.8 (Table 6). Unexpectedly, in the past, it was demonstrated that a Turing machine (TM) can be embedded in a cellular automaton [92–96]. One example is provided by a well-known, ‘very primitive’, two-state CA called the ‘Game of Life’, which is proven to be capable of performing universal computations [97].

Hence, CAs are proven to be more general than TM; it is unclear how much more general CAs are and how much complicated systems can be simulated by them. Some of the so-far unknown properties of CA models with a focus on emergence are going to be uncovered.

### 1.2. Self-Organization and Emergence Observed in Natural Systems

Emergence is found in many, if not all, observed natural phenomena that operate on scales ranging from quantum mechanics; across atoms, molecules, and bio-molecules; to cells, organs, bodies, and ecosystems in biology. In non-living matter, it goes from quantum mechanics; across atoms, molecules, solids, and liquids; to rocks, celestial bodies, stars, galaxies, and the entire Universe.

All systems expressing emergence are part of self-organizing ones, and replicating systems are part of self-organizing ones; finally, self-replicating systems are a part of replicating ones; see Figure 1. For an introduction to the modeling and analysis of complex systems [5], the books on emergence [25,74] provide a valuable introduction to the research area of emergence, along with the software provided in this review.

An important subclass of SO emergent systems is represented by self-organized criticality (SOC)-expressing systems [13,14]. Simply put, they are fed continuously by the same amount of input energy, which is released in irregularly timed bursts of energy that lead to a power-law distribution of avalanches of released energy. Examples encompass

earthquakes, stock market crashes, multi-scale plasma instabilities, traffic flow jams [15], and laser dynamics [98].

A CA model of quantum mechanics based on underlying deterministic principles was proposed by G.'t-Hooft [99] and tested in [100]. When proven correct, it will join the group of other deterministic models that yield complex and chaotic behaviors [6,7,15,20].

Yet another important class is represented by agent-based models (ABM) that are applied to morphological development and embryogenesis [35,36], societal systems, stock markets, socio-economic systems, opinion polls, etc.

Emergence is behind the occurrence of phenomena observed within solids, metals, biological systems, societal opinions, electromagnetic properties of materials, or information processing in material and biological systems; see Tables 3 and 4.

Random, small-world, and scale-free networks are very important in studying self-organization and the emergence of irregular networks in biology, medicine, sociology, the economy, and related areas; see the review [39] and books [40,41]. Important note: CAs are possible to interpret as perfectly regular networks, unlike the above-mentioned ones.

### 1.3. Self-Organization and Emergence Observed in Biological Systems

A growing volume of research is confirming the possibility of manipulating morphology, regeneration, and even limb re-growth in lower-level species, e.g., flatworms and salamanders. Repeatedly, two-headed, two-tailed, or heads of different sub-species have been demonstrated to grow in flatworms through the manipulation of the electric potential of cell membranes [75] and salamanders. A more general understanding of body plan and morphology growth is covered in [81–84]. To give a brief insight, genes serve as logo bricks; their assembly is driven by the electric potentials on cell membranes and by the body's memories; see Table 4 for more examples.

Surprisingly, as confirmation of the above statement, Levin and coworkers observed [101] that flatworms can re-build their dissolved heads, including neurons in the head, when they get exposed to a neurotoxic agent, in this case  $BaCl_2$ . The flatworm is capable of reassembling a new head that is boron-resistant. This all happens despite the fact that, obviously, the worm has always operated above the identical genome [101]. This clearly shows that the genome is creating, as already said, building blocks like Lego bricks, and the electric environment is deciding the assembly of those building blocks into a certain living structure. Using the words of the authors, "Anatomical homeostasis results from dynamic interactions between gene expression, physiology, and the external environment [101]".

As already said, the morphological growth of cells, tissues, organs, and bodies is affected by changes in the electric potential of the cell membranes. One application in human medicine is the healing of breast tumors by specific changes in the electric potential of tumor cell membranes [85]. Such observations are far-reaching and direct us towards the development of an understanding of emergent information processing, which leads us directly towards the theoretical studies of flexibility and robustness/error-resilience of the living systems presented in this review (see Table 4).

### 1.4. Emergence in Theory of Computing

Physics, chemistry, biology, and the theory of computing are mutually interwoven research disciplines when observed from the computational point of view. Without physics, chemistry, and biology, many essential insights into the very principles of computing would be missing.

All of this started with the theoretical concept of Turing machines [102] simulating the functions of a computer in 1930s, which was followed by von Ulam's cellular automata definition [103] and the development of von Neuman's self-replicating automata [16] in the 50s. In parallel, Turing had been developing the theory of biological pattern formation on fur and other similar phenomena using reaction-diffusion systems [104], which was experimentally studied and confirmed in cyclic chemical systems studied by Belousov [105] and Zhabotinski [106].

Self-replication was later revisited by Langton in the design of the minimal self-replicator [17], which has a much lower number of states and lower complexity due to sacrificing its universality. Those studies lead to the idea of self-replicating structures using simple CA-like robots, utilizing stigmergy, that was developed by Chou and Reggia [18]. Genetic programming was applied [50] to find the local rules that lead to the recreation of the given end-goal structure.

CAs relationship to other mathematical methods like difference and differential equations can be found in [23,24]. The space of CA rules leading to emergence, replication, and self-replication often lies outside of the space of discretized differential equations. CA-rules space is richer when compared to discretized differential equations.

The definition of ‘Game of Life’ using eight nearest neighbors (N, NE, E, SE, S, SW, W, and NW) is shown in [27]. Many emergent structures are observed there; some examples follow. Additionally, we know that replication and self-replication are observed in biology and chemistry [69–71], which deserves greater attention. The above reasons motivated the study of generalized versions of GoL [28,29], with results presented in this review, see the text.

Importantly, sometimes it is necessary to discern relationships among a huge number of data to be capable of designing the evolution rules of CAs; this is where ML becomes useful, see introduction [42,43,45–47].

### 1.5. Distinguishing between Traditional and Massively Parallel Computing

A side-by-side comparison of two major computational methods known in the theory of computation: (a) the classical von Neumann architecture that is based on the sequential processing of information using logic gates, and (b) the emergent one [74] that is based on the utilization of massively parallel computations (MPC) (cellular automata, movable agents, atomic and molecular interactions, etc.), see Table 5.

**Table 5.** A side-by-side comparison of two major approaches known in the theory of computing: (a) the standard, classical, sequential von Neumann architecture; and (b) the future, massively parallel information processing observed in natural phenomena and living systems (as well-developed by S. Ulam and J. von Neumann [16,103] in the 1940s).

Computation Type	Classical	Emergent
Implementation method	artificial, designed, engineered, sequential, mechanical, liquid, and in silico	natural, self-assembling, automatic, massively parallel, in physical and biological systems
Constituting units	AND, OR, NOT, and XOR logic-gates	atoms, molecules, cells, agents, graph nodes ...
Evaluation methods	fixed wiring of logic-gates in fixed circuits	self-organized and emergent information processing circuits
Design	micro-mechanic, micro-flow, micro-electric elements, and connective paths	any medium enabling massive parallelism and definition of the local rules
Information flow	predominantly sequential or slightly parallel	inherently massively parallel
Output	single-state or value	everything between single value and multidimensional array
Error-resilience	virtually non-existent	quite high in a subset of generalized r-GoL

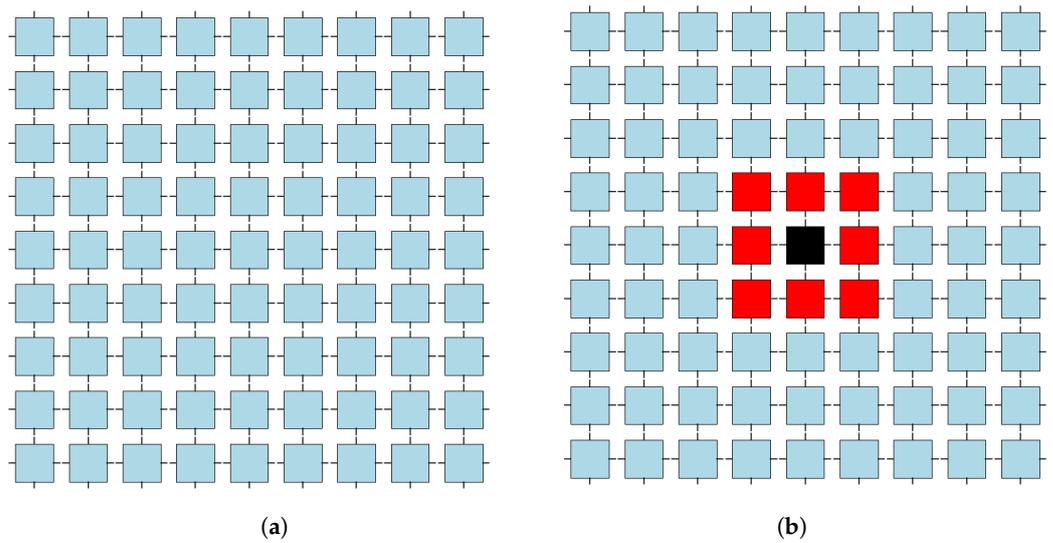
### 1.6. Brief Introduction to Cellular Automata and Complex Systems Simulations

Figure 2 depicts the very principle of massively parallel computation demonstrated on cellular automata: (a) a uniform lattice of square cells defining the computational medium; and (b) a uniform cell’s neighborhood identical to each cell.

The cellular space  $L$  is in this description defined by the two-dimensional lattice of square cells  $\square_{i,j}$  (see the more general definition in [26])

$$L = \{\square_{i,j} | \forall i \in \langle 1, N \rangle, \forall j \in \langle 1, N \rangle\}, \tag{1}$$

where each cell  $\square_{i,j}$  has the associated state  $s_{i,j}$ . The top and down, and left and right, edges of the lattice are connected together, which creates a toroid.



**Figure 2.** The principles of massively parallel information processing are described in the prototype of all such systems: a cellular automaton. (a) The lattice definition. (b) From the extended neighborhood of 24 cells, eight of them are chosen and assigned as neighbors to each cell (the neighborhood of just one cell is shown). Each cell evaluates its next state in the next step according to the states of itself and its defined neighbors. (a) An example of a cellular automaton lattice having a size of  $9 \times 9$ . The real lattices are substantially larger than neighborhoods. (b) One instance of neighborhood (selected from 735,471 possible within extended  $5 \times 5$  neighborhood); details are presented in forthcoming Section 3.4 (see simulations shown in Figure 7 there).

In our case, the state of each cell can attain only two values

$$s_{i,j} = \{0, 1\}, \tag{2}$$

where the 0 state (depicted as a white square,  $\square_{i,j}$ ) is often called the dead state and the 1 state (depicted as a black square,  $\blacksquare_{i,j}$ ) is called the alive one.

The uniform neighborhood, which is identical for each cell, is defined by eight neighbors  $N$ , which are selected from the extended neighborhood  $N_{radius} = \{\text{given } i, j | \forall i \in \langle i - 2, i + 2 \rangle, \forall j \in \langle j - 2, j + 2 \rangle\} = 5 \times 5 - 1 = 24$  cells; see Figure 2. This gives a total of  $\binom{24}{8} = 735,471$  possible neighborhoods:

$$N = \{n_{i,j} | \forall i, j \in \{8 \text{ pre-selected } \square_{i,j} \text{ from-within } N_{radius}\}\}. \tag{3}$$

The uniform evolution rule  $E$  (known as the transition function or local rule) defines the change of the state of each cell going from step  $N$  to step  $N + 1$  independently among all cells (the evaluation of the next cell's state is always centripetal; no change of the state of any neighboring cell is allowed):

$$s^{t+1} = E(\{s_{i,j}^t | \forall \text{ cells } \square_{i,j} \in N\}). \tag{4}$$

The global transition function, which describes the evolution of the whole system

$$G_{N+1} = E(G_N) = \{E(s_{i,j}) | \forall i \in \langle 1, N \rangle, \forall j \in \langle 1, N \rangle\} \tag{5}$$

has no closed analytic solution in the vast majority of cases.

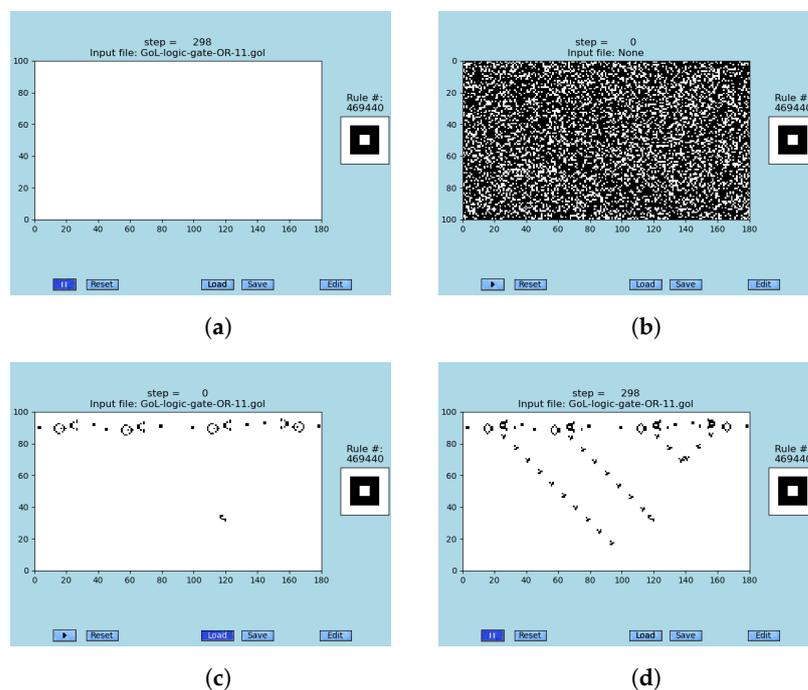
The principle of the program that simulates cellular automaton is described by the pseudo-code of the Algorithm 1; as examples, see simplified GoL software in [28] and full GoL-N24 software in [29]. The major stages of each simulation by GoL-N24 are depicted in Figure 3.

**Algorithm 1** The algorithm, describing GoL-N24, is performed simultaneously in all cells during one time-step within the function *UpdateCell*, which itself is called from the *main()* function. This algorithm was used to generate all simulations except r-GoL ones.

```

1: function NEIGHBORSALIVE(i, j)
2:   Count number of alive neighbors when  $state \geq AliveThreshold$ 
3: end function
4: function UPDATECELL(i, j)
5:   if NeighborsAlive == 3 then
6:     state = 9 “# Cell becomes alive”
7:   else if (NeighborsAlive == 2) and (cell >= AliveThreshold) and (cell <=
8:     DeadThreshold) then
9:     state = 9 “# Cell becomes alive”
10:  else
11:    state = 0 “# Cell dies or stay dead”
12:  end if
13: end function
14: function MAIN()
15:  for t = 0, t < End do
16:    for i = 0, i < M do
17:      for j = 0, j < N do
18:        UpdateCell(i, j)
19:      end for
20:    end for
21: end function

```



**Figure 3.** The principles of CA simulations and emergence are explained in the simulation of the cellular automaton ‘Game of Life’ [4,29]: (a) empty matrix, step #298; (b) random matrix, step #0; (c) predefined initial configuration, step #0; and (d) simulation of the case ‘c’, step #298. Those simulations are not error-resilient. (a) No output: an empty matrix without any initialized alive cells with an embedded local rule gives empty output. (b) Random input: step #0, a matrix with a random initial configuration; the local rule will lead to the evolution. (c) The logic-gate OR: the initial configuration at step #0; see [79] for animations. (d) The logic-gate OR: the configuration at step #298; see [79] for animations.

### 1.7. Brief Introduction into Error-Resilient GoL Called r-GoL

The CA r-GoL [4] is a variant of GoL where the local rule uses the redundancy of the cell  $state \in \langle 0, 9 \rangle$  to accommodate randomly injected errors into the evaluation process unlike the original GoL using  $state \in \{0, 1\}$ . The later fails to continue its evaluation with an even single injected error [4]; see Algorithm 2 for details (software link in [28]).

As already stated, unlike the original GoL, which has two states per cell (alive = 1, dead = 0), r-GoL has 10 states per cell. This requires defining when a cell is alive or dead in r-GoL. This is accomplished by utilizing two thresholds, *AliveThreshold* and *DeadThreshold*, where  $0 < AliveThreshold < DeadThreshold \leq 9$ . A cell is said to be alive in r-GoL when its state is above an *AliveThreshold* and below a *DeadThreshold*. Cells in the neighborhood are alive when their state is above the alive-threshold.

Two simulations are demonstrated in detail later within Section 3.3, see Figures 5 and 6 there; the first figure does not have randomly injected errors while the other one has injected 1% of them. The later simulation differs from the first one but is error-resilient.

---

**Algorithm 2** The algorithm, describing r-GoL, is performed simultaneously in all cells during one time-step. The main loop is omitted; it is same as in Algorithm 1. The algorithm used to generate the simulation in Figures 5 and 6. The simulation presented in Figure 5 has the same algorithm except for the random flipping of cells.

---

```

1: function NEIGHBORSALIVE(i, j)
2:   Count number of alive neighbors when  $state \geq AliveThreshold$ 
3: end function
4: if NeighborsAlive == 3 then
5:    $state = 9$  “# Cell becomes alive”
6: else if (NeighborsAlive == 2) and ( $cell \geq AliveThreshold$ ) and ( $cell \leq$ 
    $DeadThreshold$ ) then
7:    $state = 9$  “# Cell becomes alive”
8: else
9:   if  $rand \leq 0.01$  then
10:     $state = 1$  “# Cell becomes alive with a lower value”
11:   else
12:     $state = 0$  “# Cell dies or stay dead”
13:   end if
14: end if

```

---

### 1.8. Outline of the Review

The review has two major lines: error-resilient and non-resilient. Error-resilient is represented by r-GoL cellular automata [4], where the transition rule is changed, and the rest of the paper is represented by GoL-N24, where only the neighborhood varies.

The text is composed of the following parts: (a) The introduction contains a description of the difference between classical and massively parallel models and information processing, and it provides an overview of the most important methods used in the description of complex systems and methods in the form of concise tables, which provide the most important processes used to describe complex systems. All of this is simultaneously observed through the lenses of different scientific areas: mathematics, physics, biology, and computer science, among others. The introduction draws our attention towards the explanation of the main process studied in this review: emergence [74]. This is all finalized by a brief introduction to cellular automata. (b) The main thesis of the review. (c) The section dealing with examples and descriptions of various instances of massively parallel computations: ‘Game of Life’ (GoL), GoL-N24, logic-gates in GoL, error resilient emergents in r-GoL, gliders, ships, and emergents of the 2nd order. (d) The simulation of massively parallel systems using cellular automata. (e) Counter-examples to the proposed approach, (f) which are followed by a number of arguments or examples confirming it. (g) All is finalized with the section describing the perspective and future directions, which is followed by (h) conclusions.

For the purpose of easy navigation throughout the entire text and defining the four major types of emergence studied and reviewed in the entire text, the following Table 6 puts all four ways of achieving emergence together in one place.

**Table 6.** Summarizing the four major types of emergence studied and reviewed in the entire text. It is important to keep this information throughout the reading of the entire text and distinguish between those four different types.

Class of Emergence	Where It Occurs
Pure Emergence.	Sections 3 and 4 together with Algorithm 1.
Multi-Level Emergence.	Section 3.4.
Emergents' Error-Resilience Against Injected Evaluation Errors.	Section 3.3 together with Algorithm 2.
Emergents' Resilience Against Neighborhood Alternation.	Section 4.3.

## 2. Thesis: Existence of Emergent Information Processing

Thesis: "Can we design artificial, massively parallel, self-organized, emergent, error-resilient computational environments?"

The most wanted properties in such a design are emergent and error-resilient ones; when achieved, they will eventually lead to self-assembling and self-healing information-processing environments.

The main motivation of the thesis is delivered by biological systems as they have exactly the same properties. As shown in Tables 1–4 in Section 1, it is known that biological systems are running the same 'program', which is composed of many interdependent processes, on the top of always reconstructing, re-configuring, rebuilding, and fluid wetware. Is it possible to build such systems artificially?

Beware, this review is solely focused on the generic type of all massively parallel computational environments: cellular automata. Everyone can apply the approaches covered here to the respective fields of complex systems.

## 3. Simulations of Massively Parallel Computations Using Cellular Automata

A concise introduction to cellular automata (CA) [26], along with CA-books covering quite diverse topics and CA-examples [25,26], is provided to allow a quick start in the field. This learning stage is recommended to be followed and even accompanied by the following software, which together will assist in building a strong skill set in CA design and programming.

A recommended, easy-to-think-through starting point to understand the very principles of CA computations and programming is the following: less than one hundred-lines long Python [107,108] program simulating the John. H. Conway's 'Game of Life' [109] along with CA theory covered, e.g., in [26]. This is recommended to be followed by books [25,33,35], more advanced CA software [28,29], and agent-based software [34,37]. The combination of experimenting with CA-codes along with studying their applications represents the fundamental approach to reaching the cutting-edge of knowledge in this field quickly and efficiently. The open-source Python software GoL-N24 [29] was used to create all simulations (initial configurations are open-source) except the simulations of r-GoL presented in Section 3.3 (see Figures 5 and 6 there), which were simulated by [28].

### 3.1. Neighborhood Definition within GoL-N24 Cellular Automaton Implementation

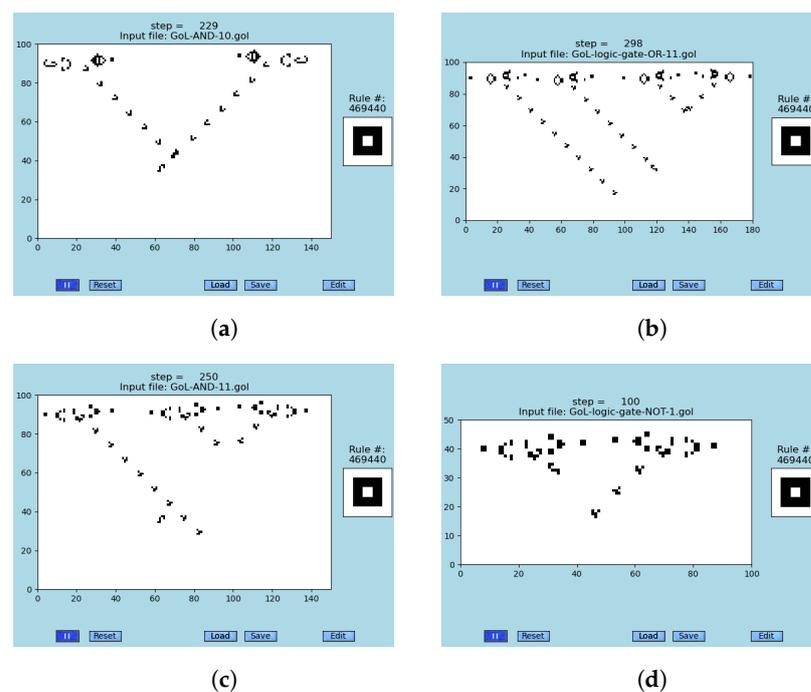
Each rule number is defined in the following way. The extended neighborhood is numbered as  $2^{x_1} + 2^{x_2} + 2^{x_3} + 2^{x_4} + 2^{x_5} + 2^{x_6} + 2^{x_7} + 2^{x_8}$ , where the numbers  $\{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$  represent the ordering number of the given cell in the neighborhood: counting

starts at the lower-left corners and proceeds in rows upwards to the upper-right corner (the left-lower corner has position 0 and contributes by the value of  $2^0 = 1$  and, finally, the upper-right corner has position 24 and contributes by the value of  $2^{24} = 16,777,216$ ). The central cell is not part of the neighborhood.

### 3.2. Implementation of Logic-Gates in the ‘Game of Life’ Cellular Automaton

Several independent streams of emergents that are generated by a predefined number of glider-guns—when designed in precisely aligned geometrical configurations—can carry on logic operations by utilizing collisions of glider-streams. This is demonstrated in Figure 4, which was created by program [29], with details covered in [4] and animations provided in [79]. In other words, glider streams with their mutual collisions define the specific logic-gate type: the given simulated operator emerges through and from all emergent processes of all of the involved glider-guns. Each logic-gate must be precisely initiated through glider-guns and different blockers. A single error present within any element of the given operator leads to its malfunction and disappearance.

Each glider-gun is composed of two counter-wise moving parts, which together emit gliders in regular moments that propagate further independently. The specific topology of glider streams, which can be switched on or off using various glider-stream blockers, defines the actual output of the logic gate under consideration; see Figure 4.

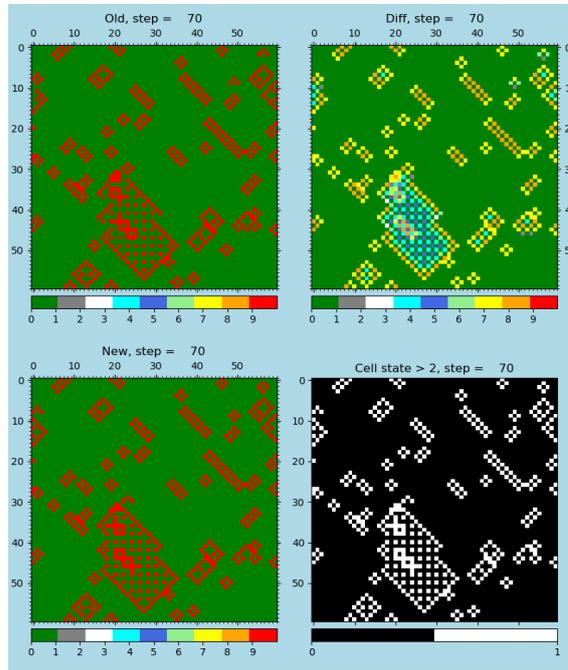


**Figure 4.** The Turing machine equivalence of the GoL (rule #469440) is demonstrated on snapshots of a glider-gun generating gliders of logic-gates AND, OR, and NOT; see Tables S4–S6, respectively. Those gates enable one to construct emergent logic circuits and memory within the original cellular automaton ‘Game of Life’ [89,97], simulated by [4,29]. These simulations are not error-resilient. (a) A snapshot of two colliding glider streams that are emitted by two glider guns; see animations [79]. (b) A snapshot of the emergent logic-gate OR, where inputs are two central GGs, step #298; see animations [79]. (c) A snapshot of the emergent logic-gate AND, where inputs are two left GGs, step #250; see animations [79]. (d) A snapshot of the emergent logic-gate NOT, where the input of zero is the left GGs, step #100; see [79].

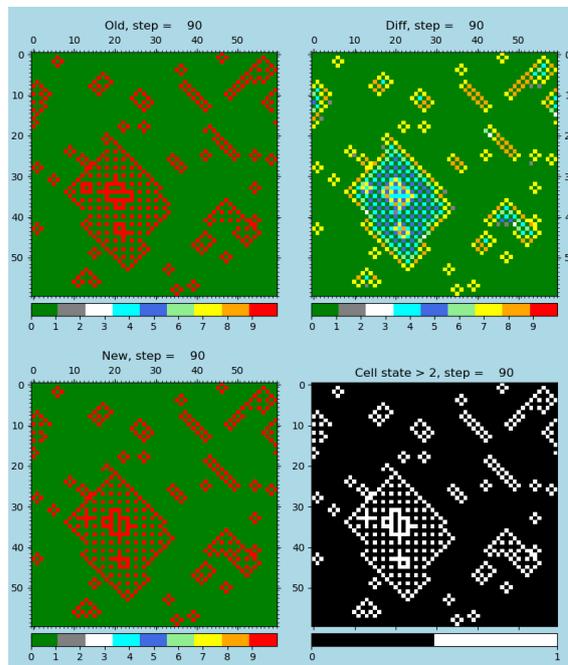
### 3.3. Error-Resilient Emergents Observed in Cellular Automata

The hypothesis of the existence of error-resilient emergents, which utilize generalized local rules, was tested and confirmed in the CA, named r-GoL [4], using software [28]; animations are available in [79] and Table S2.

It was observed in the modified GoL, called r-GoL, that, contrary to the GoL rule, some emergents express resilience against the injection of 1% of errors in the evaluation process of the rule; see details in Figures 5 and 6.



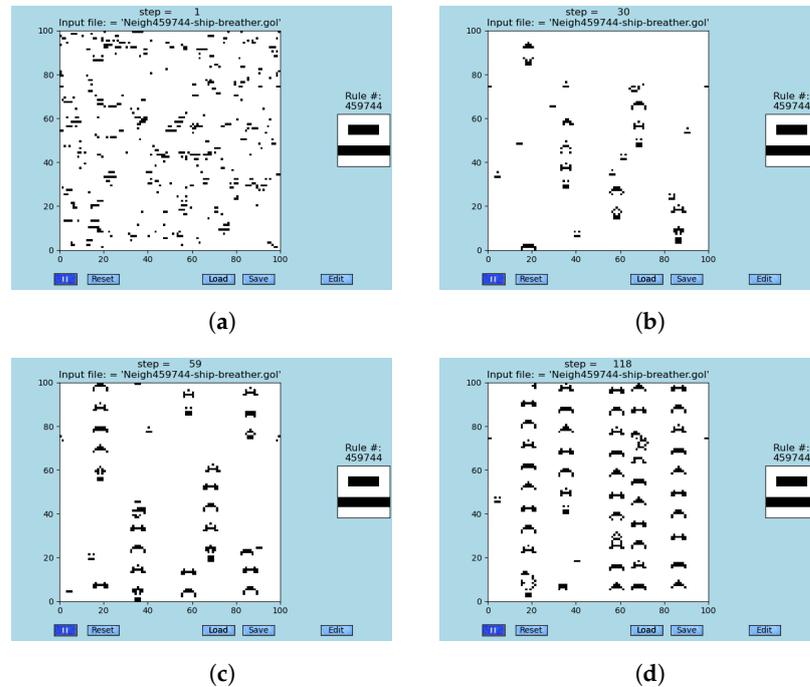
**Figure 5.** A demonstration of the effect of the changed rule (r-GoL) and the identical neighborhood. A snapshot of error-resistant (resilient) emergents operating within a rule variant of the ‘Game of Life’, r-GoL, is shown: an undisturbed error-resilient rule; see the publication [4], software [28], animations [79], and Table S2. The old, diffusion, new, and state > 2 sub-figures at step 70 are shown.



**Figure 6.** The same error-resilient rule and simulation as in Figure 5, with the only difference being that injected 1% of faulty evaluations are injected; see the publication [4], software [28], animations [79], and Table S2 for details. In both figures showing r-GoL simulations, old and new steps demonstrate the existence of alternating states.

### 3.4. Ship Breathing Ships Found in GoL-N24: Second-Order Emergence Confirmed

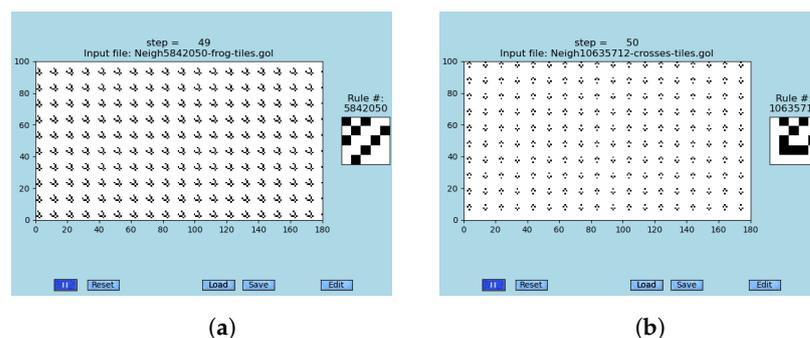
Unexpectedly, important components of emergent information processing were already found; see Figure 7, animations [79], and Table S7. The first-order emergents that generate a stream of second-order emergents are observed. This opens a window to the exploration of unconventional self-assembling circuitry design; see other examples in [4], which is a highly challenging task requiring novel computational approaches.



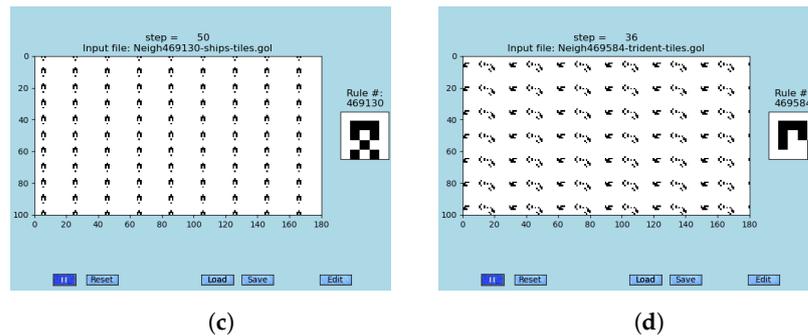
**Figure 7.** Four snapshots of ships breathing ships that are observed within the modified ‘Game of Life’, GoL-N24, rule #459744; see animations [79] and Table S7. An important observation is that almost any simulation breeds those two-level emergent structures from the random initial conditions. (a) Almost any random initial condition gives birth to at least one breathing ship later in a simulation, step #1. (b) The first four breathing ships (a rectangle with two dots above it) occur, e.g., see coordinates (30,30), step #30. (c) Trails of secondary emergents are already partially built, step #59. (d) Trails of secondary emergents almost fully build, step #118. Collisions occur.

### 3.5. Gliders Observed in Different GoL-N24 Neighborhoods

An important observation within simulations that were carried out on a whole range of different neighborhoods in GoL-N24 is the fact that important components of emergent information processing were found in many of those tested neighborhoods: gliders emerge from random conditions; see Figure 8 for selected examples. Hence, one of the most critical components of the emergent computation proves to be abundant in GoL-N24.



**Figure 8.** Cont.



**Figure 8.** The effect of different neighborhoods on the identical rule (GoL) is demonstrated by various types of gliders that were originally generated from the random initial conditions. (a–c) propagates in the speed of one cell per one time step. (a) The glider called the frog, rule #5842050; see video [79] and ‘\*.gol’ ini-file [90]. (b) The glider called the opposing crosses, rule #10635712; see ‘\*.gol’ [90]. (c) The glider called the rowboat, rule #469130; see ‘\*.gol’ ini-file [90]. (d) The glider called the trident, rule #469584; see ‘\*.gol’ ini-file [90].

#### 4. Counter-Examples

A number of possible counter-examples to the hypothesis—where it is initially assumed that all are true; each of them is followed by supporting evidence and subsequently by its refutation—are provided and discussed in this section.

##### 4.1. Genes Are Encoding All Processes Observed within Bodies of All Living Entities

It is well known from a huge number of research publications dealing with genetics that genes encode all processes occurring and operating within living organisms.

**Evidence.** The functions of cells, tissues, organs, and bodies are defined and solely driven by the expression of genetic material and nothing else as disruption of the genetic code leads to disruption of cellular and bodily functions.

**Refute.** Gradually, it was demonstrated that epigenetics plays an indispensable role in the health and disease of living organisms. On top of that, it is experimentally proven that physiology, cellular regulation, tissues, organs, and the body plan are defined by the value and distribution of the electric potentials on cellular membranes [2,77,101].

##### 4.2. Genes Are Encoding All Body Plans Observed within Bodies of Living Entities

It is a widely accepted idea that morphological development and the resulting body plans are encoded locally by genes.

**Evidence.** Defects in gene-encoding lead to dysfunctions, as is known from and proven by experiments with knock-out genes, which in turn lead to malformations within tissues, organs, and body growth.

**Refute.** The tissues’, organs’, and bodies’ growth and morphological development can be manipulated by the electric potential changes on the surface of cells in larger volumes. Additionally, it had been proven that an arbitrary number of legs, heads, tails, two-heads, or two-tails in planarian worms and lizards can be arbitrary engineered by mere changes in the electric potential on cell surfaces [2,77]. The body plan is defined by the electric potential on cell membranes within the tissue of developing living entities.

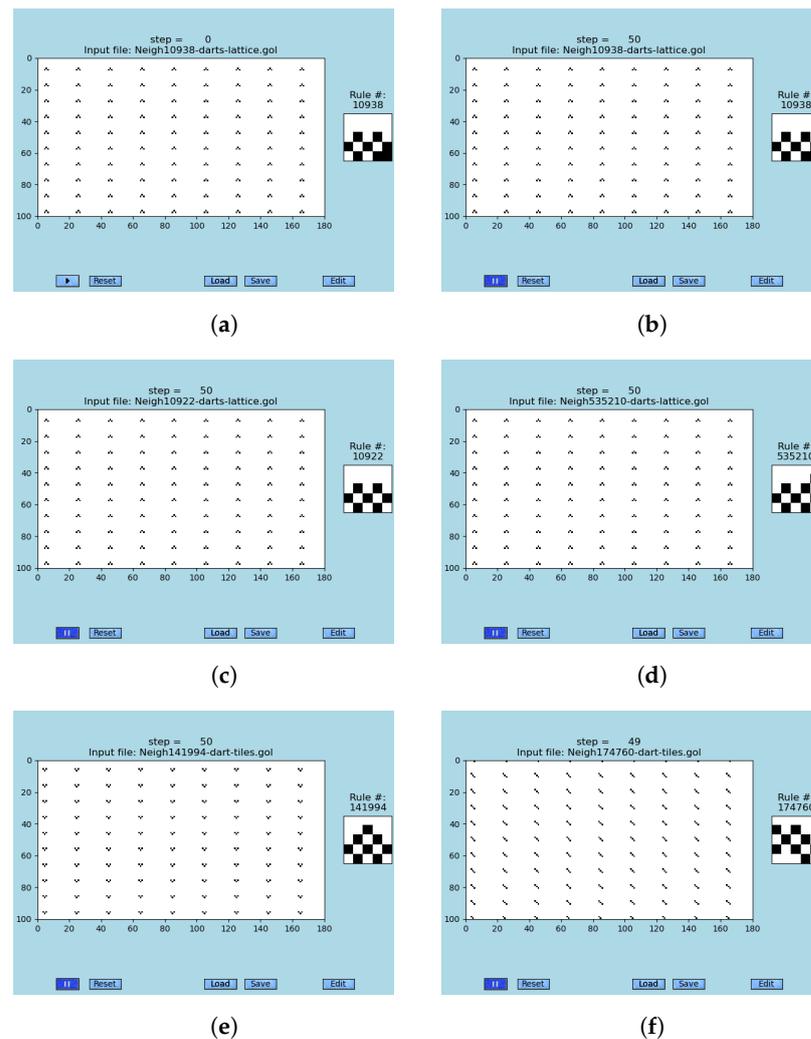
The morphological growth is the direct result of the interplay among gene expression, epigenetics regulation, and electric potentials on cell membranes.

##### 4.3. Resilience of Emergents against Perturbation of Neighborhood

The problem with emergence is that emergents are not resilient when the neighborhood is perturbed.

Evidence. Emergents cannot maintain their shape even after even the slightest change in the GoL neighborhood; see logic gate simulations [4] (an example of OR logic-gate in Figure 4) that work with emergent glider-guns and gliders.

Refute. As the counter-example, observe the resilience of emergents, which were demonstrated to persist even when the underlying neighborhood was gradually changed Figure 9. Additionally, it was demonstrated in [4] that the special design of the evolution function enables emergents to resist the injection of 1% of evaluation errors into the simulation. Hence, it has been demonstrated that a class of emergents exists that are resistant to perturbations of either neighborhood or evolution function.



**Figure 9.** Resilience of emergents is demonstrated in three different neighborhoods, which generate the very same emergent arrow-glider from identical initial conditions (a). The computations in sub-figures (e,f) are initiated by the arrow pointing down, where in (f) the symmetry gets broken and the evolution alternates between the initial arrow and a diagonal line. The computation in figure (c) has only seven neighbors; see GoL-N24 [29] and animations [79]. (a) Cross-glider with neighborhood #N10938, step #0. (b) Cross-glider with neighborhood #N10938, step #50. (c) Cross-glider with neighborhood #N10922, step #50. (d) Cross-glider with neighborhood #N535210, step #50. (e) Cross-glider with neighborhood #N141994, step #50. (f) A glider with neighborhood #N174760, step #50.

## 5. Arguments/Examples

The hypothesis is supported by the following observations and phenomena observed across the whole spectrum of scientific fields, including biology, where it should be tested in depth.

### 5.1. Emergents Operating within GoL Carrying on Computations

It has been known for a long time that AND, OR, and NOT logic gates can be constructed within GoL using glider guns, gliders, and stopping structures; see Figure 4 and Tables S4–S6, respectively. Promising attempts to design emergent processing units and memories in this way within GoL are in progress [89].

Supporting examples. GoL was proven to be Turing-machine-equivalent [97], which creates a solid basis for its future applications in emergent computations.

### 5.2. Gliders Are Observed in Modified GoL-N24 with Extended Neighborhood

From the proven Turing equivalence of GoL [97], there is a hypothesis that the presence of gliders in modified GoL (see Figure 8) indicates that there is a high probability of the discovery or design of glider-guns. Once those emergents are found, any type of classic logic can be designed.

Supporting examples. Many randomly tested neighborhoods within GoL-N24 contain gliders and ships; in some cases, even more complicated emergent structures are observed (e.g., second-order emergents in the neighborhood #459744, see Figure 7).

### 5.3. Emergent Information Processing Produces Morphology

MPCs have the proven potential [1,30–32] to process information in so-far poorly understood, distributed forms, which are computationally richer and simultaneously very robust when compared to classical logic gates used in designed computing devices.

Supporting examples. The proof is provided by many known natural systems, including biological systems. The morphological growth of living organisms represents the best example. It is known that *Dictyostelium discoideum* [1,30] (slime mold) solves complicated spatio-temporal tasks while searching for food; the same goes for ants and other social insects [32]. Additionally, *D. discoideum* is capable of assembling single cells into a fruiting body using simple signaling techniques, which serves as an example of a primitive anatomy plan.

Some organisms and even cells use the external environment as a kind of external memory, e.g., ant pheromone-trails and honey-comb structure—they make local, independent decisions accordingly to the interactions with the surrounding environment, which leads to self-organization and emergence [31] without the presence of the internal memory. The flocking behavior of birds expresses similar features.

### 5.4. EIP Produce Tissues, Organs, and Bodies

It is known that bodies, organs, and tissues exist throughout the entire life span of each living organism, while its constituting elements are continuously repaired and replaced. In humans, on average, this might take about 1 year to replace the majority of cells in the whole body: some are replaced within days like square cells in the colon, and others may take many years like long sensory and motor nerve connections. Persistence of function and maintaining the identity of the whole while the constituting parts are continuously replaced represents an excellent example of emergence, which expresses error-resilience and self-repair.

Supporting examples. Stem and senescent cells' replacement and cellular regeneration are insufficient to perform tissues', organs', and bodies' regeneration as morphology is not fully defined by them. Those questions have already been addressed in biomedical research for over two decades by trying to uncover the role of cell-membrane potentials in the genesis and maintenance of living organisms, e.g., [2,77]. Well-defined changes in cell-membrane potentials are capable of initiating the regrowth of lost limbs in lower-level vertebrae and even healing breast cancer in humans.

There is an increasing number of researchers of consciousness, e.g., [110,111], who have explored the origins of all living and even non-living forms using various ways combining quantum approaches, emergence, and consciousness.

### 5.5. Physical Laws Are Emergent

Contrary to our beliefs based on daily experiences, physical laws themselves are emergent and not fixed. For example, at the microscopic level, a solid material is an emergent arising through the interactions of its constituting components, called atoms. These interactions result in a specific crystal lattice. Every element has a specific internal structure that results in a specific crystallographic lattice. Very complicated dependencies cannot be explained in any other way than by the application of massively parallel interactions; examples are non-Newtonian liquids [67] and dynamic recrystallization [3]. Atoms themselves emerge from lower-level interactions. Interestingly, according to recent research, laws governing quantum mechanics might lead to emergent, higher-level phenomena [99,100].

Supporting examples. For example, strength, stiffness, lattice type, and temperature are emerging properties of solids. Non-Newtonian liquids have a very different response to physical loading during slow deformations (liquid-like) and fast deformations (solid-like). This type of behavior is reached in liquid materials containing long strings like starches or polymers, which smoothly slide on each other during slow deformation speeds but mutually interlock during fast deformation speeds.

Other examples can be found in statistical physics, Hook's law, DRX [3], and many other natural phenomena that express self-organizing and emergent properties.

### 5.6. Artificial Morphology Growth and Embryo-Genesis as Proxy to Uncover Principles of Emergent Design of Computation

Artificial morphology represents a means to recreate the processes that are observed during the embryonic growth of living organisms, which in turn enriches our understanding of phenomena observed in health and disease because it can be compared with real biology. As a bonus, it will improve our theory of EIP and MPCs.

Supporting examples. Morphogenesis [77,78], swarm behavior [32], and other observed phenomena operating within biological systems give us the means to validate the tested models.

The understanding of this area of biological phenomena is still emerging. Top-down models [2] dealing with pattern formation, growth, and embryo-genetic regulation are complementing bottom-up developmental models that are discussed in this paper and, e.g., [4], in mathematical description of biological systems.

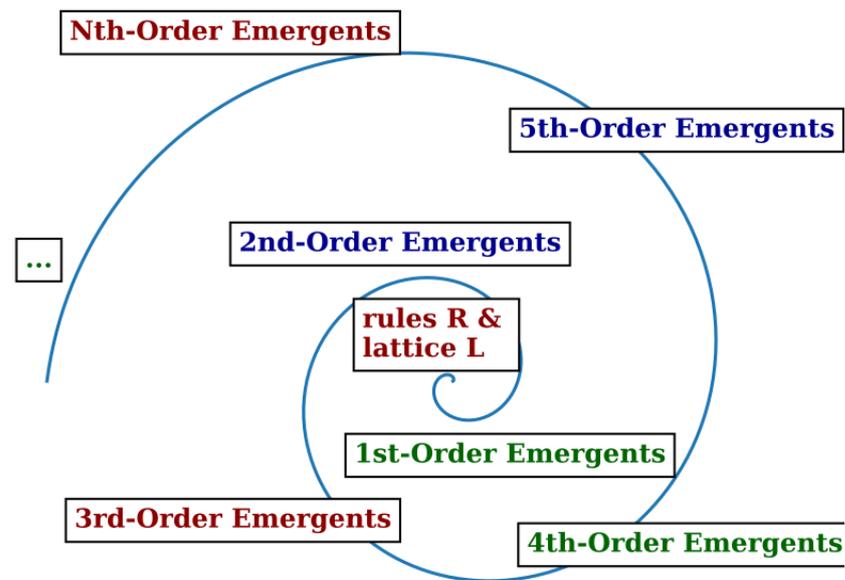
### 5.7. Error-Resilient Emergent Information Processing as Alternative to VLSI Technology

There is a rising chance of the emergence of the following very important application. Contrary to VLSI chips that are artificially designed by humans, emergent computations and information processing are not prone to the occurrence of local errors due to the existence of their error-resilience within the local rules and neighborhoods (partially proven here and in [4]).

Supporting examples. The morphological growth of living entities that maintain emergents at many hierarchical levels. Did anyone ever observe a biological entity, e.g., a vertebrae species, which after the failure of a single neuron cease to function? It is just the opposite with our current VLSI technology: a failure of a single-chip constituting element is often detrimental to the performance of the whole processor, or it can become even fatal.

## 6. Future Directions

As has already been shown, emergent, error-resilient structures exist within relatively simple CAs; see Figure 10. This is opening the doors towards the development of novel descriptive tools capable of morphological-development-like descriptions of biological systems; such approaches can be equally well applied to the description of simpler chemical and physical systems.



**Figure 10.** A spiral depicting the emergence of the hierarchy of emergents, as observed within various emergent levels of the generalized ‘Game of Life’, GoL-N24, and other complex systems. The emergence results from the core rules and processes without any additional design. In this paper, we documented the first- and second-order emergents only; some of these are artificially designed, and others are pure emergents.

#### 6.1. Realization Hypothesis: Description of Emergent Information Processing from the First Principles

What would be the line of attack along which it is possible to resolve the Existence of the EIP Thesis? As already demonstrated, there are known examples of such systems (see this text), but there is no theory enabling us to design them systematically. All of the presented emergents were discovered by the following procedure—called the forward task: randomly choosing a rule  $\rightarrow$  applying it to the matrix  $\rightarrow$  observing occurrence of emergents and their operation.

The opposite direction—called the reverse task—where the desired emergent structure initiates the search for local rules that are generating those emergents is currently unavailable. It requires deep future theoretical studies due to its intractability using brutal force approaches.

#### 6.2. Emergent-Logic Hypothesis: Beyond Gödel’s Theorem of Incompleteness

From the simulations of logic-gates (see Figure 4), significant implicit and not immediately obvious consequences arise.

What kind of reasoning allow us to understand the fact that logic can be simulated within complex systems—see the examples of logic-gates AND, OR, NOT, and XOR simulated using GoL?

We just proved with an example that massively parallel interactions are capable of generating standard logic operations [4,89,97]. This leads to the conclusion that logic arises from and through a medium of low-level, massively parallel interactions. The big question is: How is it realized? The design sequence of logic gates within GoL is: a rule  $\rightarrow$  projection into matrix  $\rightarrow$  logic.

It seems to be that our mathematical logic is coarse-grained and human-readable, whereas MPC logic is fine-grained and beyond standard human perception due to its inherent massive parallelism. The search for rules governing MPC logic is one of the

cornerstones in theory and design of future models of living systems and error-resilient information processing devices, including computers.

6.3. Emergent-Logic Hypothesis: What Are Means of Decision-Making in CSs?

From all of the results so far presented and as an extension of the previous subsection, it is proven that logic can be simulated by MPC logic gates in GoL (see, e.g., [89,97]). This implies the fact that there must exist a richer, elusive, massively parallel ‘logic system’, which is going to be called emergent logic (=MPC-logic), that in some special cases collapses into already known mathematical logic (in our case, it collapses into emergent logic gates, see Figure 4). Currently, it is only possible to speculate on the character and theory of this emergent, massively parallel emergent-logic.

From the work of Kurt Gödel, it is known that there are existing logical statements within any axiomatic system containing Peano arithmetic, which are impossible to evaluate. A very important question is: “Is the underlying, massively parallel, emergent-logic complete or incomplete?” A quite important check in depth is the possibility that all living systems are operating at the level of emergent-logic. When this is true, we can look at logic operations without even noticing them due to their massively parallel nature. The author’s personal hypothesis is that massively parallel emergent-logic is error-resilient and, hence, the issue of completeness/incompleteness is avoided in this way.

The central question: Are all living systems operating and performing decisions in a highly distributed manner, utilizing massively parallel, emergent-logic decisions? In such cases, emergent logic is virtually invisible to the naked eye.

6.4. Emergence of Natural Laws’ Hypothesis: Are Physical, Biological, and Other Observed Natural Laws Just Emergents Originating in Lower-Level Processes?

Statistical physics is the first example of a theory where collective emergent properties (temperature, pressure, information, etc.) arise from the micro-states of atoms or molecules without the necessity of tracing the energy and position of each single atom. As we know, Ludwig Boltzmann [53,54] proved that it is not necessary to trace every micro-state of the system; instead, emergent properties are derived from collective statistical properties based on micro-states.

Following the above example, when we assume the emergence of natural laws’ hypothesis to be true, it leads us to the conclusion that a much larger set of emergent behaviors originating in MPCs have to exist. Instances of MPC contained within this set are not describable by standard analytical and computational approaches used to define natural laws in the closed analytical form we are used to (see Tables 7 and 8).

**Table 7.** The abstract concept of the hierarchy of emergents states that processes operating in the parent medium (emergents of the  $(N - 1)$ th order) creates emergents of the  $N$ th order in the child medium. Each next-level child-medium and child-process is based on principles different from its parent-medium.

Process	Matrix	Emerging Phenomenon
Cellular automaton rules	periodic lattice of identical elements	glider guns, gliders (1st-level emergents)
Rules of movable agents	continuous space	collisions, crowds (1st-level emergents)
...		
Interactions of 1st-level emergents	1st-level emergent medium	2nd-level emergents
Interactions of 2nd-level emergents	2nd-level emergent medium	3rd-level emergents
...	.	.
Interactions of $N - 1$ th-level emergents	$N - 1$ th-level emergent medium	$N$ th-level emergents

**Table 8.** The hierarchy of emergents is explained by well-known natural phenomena observed within physics, biochemistry, and biology that express emergent behavior.

Process	Matrix	Emerging Phenomenon
Atomic bonds	space	metallic lattice
Atomic bonds	space	chemical compounds
Bonds among chemicals	chemicals and catalysts	biochemicals
mRNA	ribosomes	proteins
Proteins	cellular functional-units	cells
Cells	cellular interior and exterior	tissues
Tissues	cellular ensembles	organs
Organs	organ systems and physiology	bodies
...	.	.

An MPC example can be observed within the well-documented DRX model [3], where only the MPC model—despite centuries of trying to derive a heuristic, probabilistic, or analytic model—is capable of describing observed stress-strain curves expressing oscillations or a single-peak response to the imposed deformation under varying deformation conditions and elevated temperatures. Centuries of failures in creation heuristics, statistics, and analytic models have been successfully replaced by MPC models.

#### 6.5. *By-No-Law-Describable Natural Phenomena Hypothesis: Are There Existing Processes Contradicting Causality and Determinism?*

To extend the Emergence of the Natural Laws' Hypothesis further, a question is given: "Is nature allowing by-no-law-describable interactions?" Does it mean that the standard notion of event sequences known as causality does not always hold? This opens doors to novel approaches in the description of observed natural phenomena that seem too 'messy' to be described by the standard, closed-form approaches used by our contemporary mathematical descriptions within science; the book [112] describes the paradigm shifts in science. Currently, there are two known proxies of such descriptions: massively parallel distributed causality and those descriptions that are potentially contradicting causality (this is just a hypothesis, which deserves a deeper study on its own). QM experiments are an area where some phenomena can contradict causality [99].

As proven in the case of DRX [3], the broader, by-no-law-describable natural phenomena hypothesis is very probable because, for example, MPCs allow for descriptions impossible to achieve by standard analytic and computational descriptions. Hence, MPC environments are richer and more expressive due to their inherent flexibility in the definition of local interactions (each single computational element makes a decision that is independent from the rest; see [23,24] for details). ODEs and PDEs lead to difference equations through numerical discretization. Difference equations are just a fraction of all possible MPEs. The existence of non-causal natural phenomena should be addressed in serious studies specially designed for this purpose.

From simulations of GoL-N24, it is known that microstate evolution is defined by the neighborhood topology along with fixed interactions within all simulations. This can result in basically two situations: either emergents arise or not. In the later case, the outcome of the simulation is silent, oscillating, or chaotic. Emergents can be compared to the by-law-describable phenomenon at the macroscale, and the rest cannot. In other words, laws are observed only in some cases. Yet, even a chaotic system is describable from the micro-state (deterministic) evolution, similarly to [6,7,20,53,54], yet no law exists at the macroscale.

A question for future studies is: "Are processes and matrix arising from realms of the ocean of the unknown, which are indescribable and beyond the reach of our common senses and detecting devices?" The quantum-mechanical description using wave function,

delocalization, entanglement, and localization with delocalization points towards the possibility of such a description [99,100]. All is open.

### 6.6. Hypothesis Creation

The very roots of the scientific method are founded on the process of falsifiable hypothesis creation. -Karl Popper [113].

The scope of the standard description of the observed phenomena is limited [112]; this has been, is, and always will be true in science. Hence, in the recent decade, scientists have started to use novel methods to create hypotheses using Machine Learning, AI, Deep Learning, data mining, and other methods of inference of relationships among data. The quantum-mechanical description of natural phenomena faces similar issues. The problem with AI, ML, and DM methods is that they are typically utilizing black-box methods that are not human-understandable.

EIP provides us with the means to develop deeper methods of description of natural phenomena, i.e., get closer to the primary causes, that go beyond by-the-equation-based ones. Brain functioning can become one such area of research where from-the-first-sight random neuronal activity patterns can have some underlying mechanism exploiting EIP. The first step would be to find some of those emergent information processing configurations—in the ideal case, error-resilient ones.

## 7. Conclusions

The main purpose of this publication is to review the current knowledge about capabilities to reproduce, means of defining the input data and local processes, means of solution, and future lines of attack to find “artificial, massively parallel, self-organized, emergent, error-resilient computational environments”. All of this is solely explained in cellular automata, as the prototypical massively parallel computational environment. Links to other research areas will be the subject of future research.

Two main streams of research along these lines demonstrate the existence of error-resilient emergents and the resilience of emergents against variations in the neighborhood. That is all, besides other features such as the proof of the existence of second-order emergents. Reviewed approaches provide proofs of possibility and partially even means of achieving the goal of this type of research. The above is built upon the top of and broadening of the known proof of Turing equivalence of the ‘Game of Life’ cellular automaton.

After providing examples and counter-examples, the possible future research paths along with a number of hypotheses are given in order to sufficiently cover as many scientific disciplines as possible. This allows everyone to apply these ideas in their respective field of research. Some of those hypotheses go beyond what is currently known and understood.

**Supplementary Materials:** The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/software3010005/s1>, Table S2: Error-resilient ‘GoL’ software variants; Table S4: Logic gate AND; Table S5: Logic gate OR; Table S6: Logic gate NOT; Table S7: Emergents expressing interesting properties [28,29,90,109,114–130].

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** All data and software created and used in this review (the open-source, Python software GoL-N24 [29] & r-Gol [28]) can be found on the RG profile of the author: <https://www.researchgate.net/profile/Jiri-Kroc> (accessed on 21 February 2024).

**Acknowledgments:** This work is entirely dedicated to the people of the Czech Republic through Department of Informatics and Computers, University of Ostrava, Faculty of Science and is entirely

the result of independent research that was fully sponsored by the author himself without any kind of external support.

**Conflicts of Interest:** The author declares no conflicts of interest.

### Abbreviations

The following abbreviations are used in this manuscript:

CA	Cellular automata
GoL	'Game of Life'
SOC	Self-organized criticality
CSs	Complex systems
EIP	Emergent information processing
ABM	Agent-based models
AND	AND logic-gate
OR	OR logic-gate
NOT	NOT logic-gate
r-GoL	Resilient GoL software
GoL-N24	GoL software using extended neighborhoods of 24 cells

### References

1. Adamatzky, A. A brief history of liquid computers. *Philos. Trans. R. Soc. Biol. Sci.* **2019**, *374*, 20180372. [CrossRef] [PubMed]
2. Pezzulo, G.; Levin, M. Top-down models in biology: Explanation and control of complex living systems above the molecular level. *J. R. Soc. Interface* **2016**, *13*, 20160555. [CrossRef] [PubMed]
3. Kroc, J. Application of Cellular Automata to Modeling of Dynamic Recrystallization. In *Computational Science—ICCS 2002*; Sloot, P., Hoekstra, A., Tan, C.J.K.C., Dongarra, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2002; Volume 2329, pp. 773–782. [CrossRef]
4. Kroc, J. Robust massive parallel information processing environments in biology and medicine: Case study. *J. Probl. Inf. Soc.* **2022**, *13*, 12–22. [CrossRef]
5. Sayama, H. *Introduction to the Modeling and Analysis of Complex Systems*; Open SUNNY Textbooks, Milne Library; State University of New York: Geneseo, NY, USA, 2015; p. 498.
6. Poincaré, H. Sur le problème des trois corps et les équations de la dynamique. *Acta Mat.* **1890**, *13*, 1–271. [CrossRef]
7. Poincaré, H. *The Three-Body Problem and the Equations of Dynamics: Poincaré's Foundational Work on Dynamical Systems Theory*; Springer: Cham, Switzerland, 2017; p. 248.
8. Arnold, V. *Geometrical Methods in the Theory of Ordinary Differential Equations*; Grundlehren der Mathematischen Wissenschaften; Springer: New York, NY, USA, 1988; p. 351. [CrossRef]
9. Strogatz, S. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering*, 2nd ed.; CRC Press: Boca Raton, FL, USA, 2015; p. 532. [CrossRef]
10. Haken, H. *Synergetics; An Introduction Nonequilibrium Phase Transitions and Self-Organization in Physics, Chemistry and Biology*; Springer Series in Synergetics; Springer: Berlin/Heidelberg, Germany, 1978; p. 358. [CrossRef]
11. Haken, H. *Advanced Synergetics: Instability Hierarchies of Self-Organizing Systems and Devices*; Springer Series in Synergetics; Springer: Berlin/Heidelberg, Germany, 1983; p. 356. [CrossRef]
12. Mandelbrot, B. *The Fractal Geometry of Nature*; W.H. Freeman & Co.: San Francisco, CA, USA, 1982; p. 461.
13. Bak, P.; Tang, C.; Wiesenfeld, K. Self-organized criticality: An explanation of the  $1/f$ -noise. *Phys. Rev. Lett.* **1987**, *59*, 381–384. [CrossRef] [PubMed]
14. Bak, P. *How Nature Works: The Science of Self-Organized Criticality*; Copernicus: New York, NY, USA, 1996; p. 212. [CrossRef]
15. Nagel, K.; Shreckenberg, M. A cellular automaton model for freeway traffic. *J. Phys. I* **1992**, *2*, 2221–2229. [CrossRef]
16. von Neumann, J. *Theory of Self-Reproducing Automata*; University of Illinois Press: Urbana, IL, USA; London, UK, 1966; p. 288.
17. Langton, C. Self-reproduction in cellular automata. *Phys. D Nonlinear Phenom.* **1984**, *10*, 135–144. [CrossRef]
18. Chou, H.H.; Reggia, J. Emergence of self-replicating structures in a cellular automata space. *Phys. D Nonlinear Phenom.* **1997**, *110*, 252–276. [CrossRef]
19. Layek, G. *An Introduction to Dynamical Systems and Chaos*; Springer: New Delhi, India, 2015; p. 622. [CrossRef]
20. Lorenz, E. Deterministic nonperiodic flow. *J. Atmos. Sci.* **1963**, *20*, 130–141. [CrossRef]
21. Milosavljevic, M.; Blakely, J.; Beal, A.; Corron, N. Analytic solutions throughout a period doubling route to chaos. *Phys. Rev. E* **2017**, *95*, 062223. [CrossRef]
22. Kroc, J.; Sloot, P.; Hoekstra, A. *Simulating Complex Systems by Cellular Automata*; Understanding Complex Systems; Chapter Introduction to Modeling of Complex Systems Using Cellular Automata; Springer: Berlin/Heidelberg, Germany, 2010. Available online: <https://link.springer.com/book/10.1007/978-3-642-12203-3> (accessed on 5 May 2023).
23. Vichniac, G. Simulating physics with cellular automata. *Phys. D Nonlinear Phenom.* **1984**, *10*, 96–116. [CrossRef]

24. Toffoli, T. Cellular automata as an alternative to (rather than and approximation of) differential equations in modeling physics. *Phys. D Nonlinear Phenom.* **1984**, *10*, 117–127. [CrossRef]
25. Illachinski, A. *Cellular Automata: A Discrete Universe*; World Scientific: Ithaca, NY, USA, 2001; p. 808. Available online: <https://www.worldscientific.com/worldscibooks/10.1142/4702#> (accessed on 5 May 2023).
26. Kroc, J.; Sloat, P.; Hoekstra, A. *Simulating Complex Systems by Cellular Automata*; Understanding Complex Systems; Springer: Berlin/Heidelberg, Germany, 2010; p. 384.
27. Gardner, M. The fantastic combinations of John Conway’s new solitaire game “life”. *Sci. Am. Mag.* **1970**, *223*, 120–123. Available online: <https://www.scientificamerican.com/article/mathematical-games-1970-10/> (accessed on 23 April 2023). [CrossRef]
28. Kroc, J. Python Program Simulating Cellular Automaton r-GoL that Represents Robust Generalization of ‘Game of Life’. 2022. Available online: <https://www.researchgate.net/publication/358445347> (accessed on 25 March 2023).
29. Kroc, J. Exploring Emergence: Python Program GoL-N24 Simulating the ‘Game of Life’ Using 8 Neighbors from 24 Possible. Available online: <https://www.researchgate.net/publication/365477118> (accessed on 25 March 2023).
30. Pears, C.; Gross, J. Microbe Profile: Dictyostelium discoideum: Model system for development, chemotaxis, and biomedical research. *Microbiology* **2021**, *167*. [CrossRef] [PubMed]
31. Heylighen, F. Stigmergy as a universal coordination mechanism I: Definition and components. *Cogn. Syst. Res.* **2016**, *38*, 4–13. [CrossRef]
32. Ahmed, H.; Glasgow, J. Swarm Intelligence: Concepts, Models and Applications. In Proceedings of the Conference Queens University, Technical Report 2012-585, Montreal, ON, Canada, 16 February 2012. [CrossRef]
33. Resnick, M. *Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds*; Complex Adaptive Systems; MIT Press: New York, NY, USA, 1997.
34. Wilensky, U. *NetLogo*; Center for Connected Learning and Computer-Based Modeling, Northwestern University: Evanston, IL, USA, 1999. Available online: <https://ccl.northwestern.edu/netlogo/> (accessed on 11 May 2023).
35. Illachinski, A. *Artificial War: Multiagent-Based Simulation of Combat*; World Scientific: Singapore, 2004; p. 787. [CrossRef]
36. Glen, C.; Kemp, M.; Voit, E. Agent-based modeling of morphogenetic systems: Advantages and challenges. *PLoS Comput. Biol.* **2019**, *15*, e1006577. [CrossRef] [PubMed]
37. Kazil, J.; Masad, D.; Crooks, R. Utilizing Python for Agent-Based Modeling: The Mesa Framework. In Proceedings of the Social, Cultural, and Behavioral Modeling, Washington, DC, USA, 18–21 October 2020; Thomson, R., Bisgin, H., Dancy, C., Hyder, A., Hussain, M., Eds.; Springer: Cham, Switzerland, 2020; pp. 308–317. ISBN 978-3-030-61255-9.
38. Barabási, A.L. *Linked: How Everything Is Connected To Everything Else and What It Means for Business, Science, and Everyday Life*; Basic Books: New York, NY, USA, 2014; p. 304.
39. Albert, R.; Barabási, A.L. Statistical mechanics of complex networks. *Rev. Mod. Phys.* **2002**, *74*, 47. [CrossRef]
40. Loscalzo, J.; Barabási, A.L.; Silverman, E.K. (Eds.) *Network Medicine: Complex Systems in Human Disease and Therapeutics*; Harvard University Press: Cambridge, MA, USA, 2017; p. 436.
41. Barabási, A.L.; PÁ3sfai, M. *Network Science*; Cambridge University Press: Cambridge, UK, 2016; p. 456.
42. Mahesh, B. Machine Learning—A Review. *Int. J. Sci. Res.* **2020**, *9*, 381–386.
43. Jhaveri, R.; Revathi, A.; Ramana, K.; Raut, R.; Dhanaraj, R. A Review on Machine Learning Strategies for Real-World Engineering Applications. *Mob. Inf. Syst.* **2022**, *2022*, 1833507. [CrossRef]
44. Cleophas, T.; Zwinderman, A. *Machine Learning in Medicine—A Complete Overview*; Springer: Cham, Switzerland, 2020; p. 667.
45. Pichler, M.; Hartig, F. Machine Learning and Deep Learning—A review for ecologists. *Methods Ecol. Evol.* **2023**, *14*, 994–1016. [CrossRef]
46. Patterson, J.; Gibson, A. *Deep Learning*; O’Reilly Media, Inc.: Sebastopol, CA, USA, 2017.
47. Shresta, A.; Mahmood, A. Review of Deep Learning Algorithms and Architectures. *IEEE Access* **2019**, *7*, 53040–53065. [CrossRef]
48. Katoch, S.; Chauhan, S.; Kumar, V. A review on genetic algorithm: Past, present, and future. *Multimed. Tools Appl.* **2021**, *80*, 8091–8126. [CrossRef]
49. Lambora, A.; Gupta, K.; Chopra, K. Genetic Algorithm—A Literature Review. In Proceedings of the 2019 International Conference on Machine Learning, Big Data, Cloud, and Parallel Computing (COMITCon), Faridabad, India, 14–16 February 2019; pp. 380–384. [CrossRef]
50. Koza, J. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*; MIT Press: Cambridge, MA, USA, 1992; p. 836.
51. Poli, R.; Koza, J. *Search Methodologies*; Chapter Genetic Programming; Springer: Boston, MA, USA, 2014; pp. 143–185. [CrossRef]
52. Espejo, P.; Ventura, S.; Herrera, F. A Survey on the Application of Genetic Programming to Classification. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **2010**, *40*, 121–144. [CrossRef]
53. Boltzmann, L. *Vorlesungen über Gastheorie*; J.A. Barth: Leipzig, Germany, 1896; Volume I.
54. Boltzmann, L. *Vorlesungen über Gastheorie*; J.A. Barth: Leipzig, Germany, 1896; Volume II.
55. Shannon, C. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [CrossRef]
56. Pincus, S. Approx entropy as a measure of system complexity. *Proc. Natl. Acad. Sci. USA* **1991**, *88*, 2297–2301. [CrossRef]
57. Pincus, S. Approx entropy (ApEn) as a measure of system complexity. *Chaos Iterdiscip. J. Nonlinear Sci.* **1995**, *5*, 110–117. [CrossRef] [PubMed]

58. Ben-Naim, A. *Farewell to Entropy: Statistical Thermodynamics Based on Information*; World Scientific: Singapore, 2008; p. 412. [CrossRef]
59. Ben-Naim, A. *Information, Entropy, Life and the Universe: What We Know and What We Do Not Know*; World Scientific: Singapore, 2015; p. 492. [CrossRef]
60. Styer, D. Entropy as Disorder: History of Misconception. *Phys. Teach.* **2019**, *57*, 454–458. [CrossRef]
61. Costa, M.; Goldberger, A.; Peng, C.K. Multiscale Entropy Analysis of Complex Physiologic Time Series. *Phys. Rev. Lett.* **2002**, *89*, 068102. [CrossRef] [PubMed]
62. Costa, M.; Peng, C.K.; Goldberger, A.; Hausdorff, J. Multiscale entropy analysis of human gait dynamics. *Physica A* **2003**, *330*, 53–60. [CrossRef]
63. Olofsen, E.; Sleight, J.; Dahan, A. Permutation entropy of the electroencephalogram: A measure of anaesthetic drug effects. *Br. J. Anaesth.* **2008**, *101*, 810–821. [CrossRef]
64. Humeau-Heurtier, A. Multiscale Entropy Algorithm and Its Variants: A Review. *Entropy* **2015**, *17*, 3110–3123. [CrossRef]
65. Liang, Z.; Wang, Y.; Sun, X.; Li, D.; Voss, L.; Sleight, J.; Higihira, S.; Li, X. EEG entropy measures in anesthesia. *Front. Comput. Sci.* **2015**, *9*, 16. [CrossRef]
66. Vargas, B.; Varela, M.; Ruiz-Esteban, R.; Cuesta-Frau, D.; Cirugeda-Roldan, E. What Can Biosignal Entropy Tell Us About Health and Disease? Applications in Some Clinical Fields. *Nonlinear Dyn. Psychol. Life Sci.* **2015**, *19*, 419–436.
67. Zhou, J.; Papautsky, I. Viscoelastic microfluids: Progress and challenges. *Microsyst. Nanoeng.* **2020**, *6*, 113. [CrossRef]
68. Cook, M.; Rothmund, P.; Winfree, E. Self-Assembled Circuit Patterns. In *DNA Computing: DNA 2003*; Chen, J., Reif, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2004; Volume 2943, pp. 91–107. [CrossRef]
69. Adamski, P.; Eleveld, M.; Sood, A.; Kun, A.; Szilágyi, A.; Czárán, T.; Szathmáry, E.; Otto, S. From self-replication to replicator systems en route to de novo life. *Nat. Rev. Chem.* **2020**, *4*, 386–403. [CrossRef]
70. Duim, H.; Sijbren, O. Towards open-ended evolution in self-replicating molecular systems. *Bellstein J. Org. Chem.* **2017**, *13*, 1189–1203. [CrossRef] [PubMed]
71. Brakmann, S. Origin of Life, Theories of. In *Encyclopedia of Biodiversity*; Levin, S., Ed.; Elsevier: New York, NY, USA, 2001; pp. 439–450. [CrossRef]
72. De Capitani, J.; Mutschler, H. The Long Road to a Synthetic Self-Replicating Central Dogma. *Biochemistry* **2023**, *62*, 1221–1232. [CrossRef] [PubMed]
73. Hölldobler, B.; Wilson, E. *Journey to the Ants: A story of Scientific Exploration*; Harvard University Press: Cambridge, MA, USA, 1998; p. 240.
74. Johnson, S. *Emergence: The Connected Lives of Ants, Brains, Cities and Software*; Penguin Books: New Delhi, India, 2001; p. 288.
75. Emmons-Bell, M.; Durant, F.; Hammelman, J.; Bessonov, N.; Volpert, V.; Morokuma, J.; Pinet, K.; Adams, D.; Peitak, A.; Lobo, D.; et al. Gap Junctional Blockade Stochastically Induces Different Species-Specific Head Anatomies in Genetically Wild-Type *Girardia dorotocephala* Flatworms. *Int. J. Mol. Sci.* **2015**, *16*, 27865–27896. [CrossRef]
76. Levin, M. *Evolution “On Purpose”: Teleonomy in Living Systems*; The MIT Press: Cambridge, MA, USA, 2023. [CrossRef]
77. Whited, J.; Levin, M. Bioelectrical controls of morphogenesis: From ancient mechanisms of cell coordination to biomedical opportunities. *Curr. Opin. Genet. Dev.* **2019**, *57*, 61–69. [CrossRef]
78. Doursat, R.; Sayama, H.; Mitchel, O. A review of morphogenetic engineering. *Nat. Comput.* **2013**, *12*, 517–535. [CrossRef]
79. Kroc, J. Exploring Emergence: Video-Database of Emergents Found in Advanced Cellular Automaton ‘Game of Life’ Using GoL-N24 Software. Available online: <https://www.researchgate.net/publication/373806519> (accessed on 9 October 2023).
80. Fields, C.; Levin, M. Does evolution have a target morphology? *Organisms. J. Biol. Sci.* **2020**, *4*, 57–76. [CrossRef]
81. Cervera, J.; Levin, M.; Mafe, S. Bioelectricity of non-excitable cells and multicellular memories: Biophysical modeling. *Phys. Rep.* **2023**, *1004*, 1–31. [CrossRef]
82. Lagasse, E.; Levin, M. Future medicine: From molecular pathways to the collective intelligence of the body. *Trends Mol. Med.* **2023**, *29*, 687–710. [CrossRef] [PubMed]
83. Watson, R.; Levin, M. The collective intelligence of evolution and development. *Collect. Intell.* **2023**, *2*. [CrossRef]
84. Bongard, J.; Levin, M. There’s Plenty of Room Right Here: Biological Systems as Evolved, Overloaded, Multi-Scale Machines. *Biomimetics* **2023**, *8*, 110. [CrossRef] [PubMed]
85. Cernet, B.; Adams, D.; Lobikin, M.; Levin, M. Use of genetically encoded, light-gated ion translocators to control tumorigenesis. *Oncotarget* **2016**, *7*, 19575–19588. [CrossRef] [PubMed]
86. Larson, R.; Malek, A. The Transformation by Catalysis of Prebiotic Chemical Systems to Useful Biochemicals: A Perspective Based on IR Spectroscopy of the Primary Chemicals: Solid-Phase and Water-Soluble Catalysts. *Appl. Sci.* **2021**, *11*, 10125. [CrossRef]
87. Larson, R.; Malek, A. The Transformation by Catalysis of Prebiotic Chemical Systems to Useful Biochemicals: A Perspective Based on IR Spectroscopy of the Primary Chemicals I. The Synthesis of Peptides by the Condensation of Amino Acids. *Appl. Sci.* **2020**, *10*, 928. [CrossRef]
88. Larson, R.; Malek, A.; Odenbrand, I. The Transformation by Catalysis of Prebiotic Chemical Systems to Useful Biochemicals: A Perspective Based on IR Spectroscopy of the Primary Chemicals II. Catalysis and the Building of RNA. *Appl. Sci.* **2020**, *10*, 4712. [CrossRef]
89. Carlini, N. Multiplexing Circuits on the Game of Life—Part 5. Available online: <https://nicholas.carlini.com/writing/2022/multiplexing-circuits-game-of-life.html> (accessed on 4 January 2023).

90. Kroc, J. Exploring Emergence: Database of Initial Files of Advanced Cellular Automaton 'Game of Life' Using GoL-N24 Software. Available online: <https://www.researchgate.net/publication/372187009> (accessed on 9 October 2023).
91. Blackiston, D.; Kriegman, S.; Bongard, J.; Levin, M. Biological Robots: Perspectives on an Emerging Interdisciplinary Field. *Soft Robot.* **2023**, *10*, 674–686. [[CrossRef](#)]
92. Lindgren, K.; Nordahl, M. Universal Computation in Simple One-Dimensional Cellular Automaton. *Complex Syst.* **1990**, *4*, 299–318.
93. Martin, B. A universal cellular automaton in quasi-linear time and its S–m–n form. *Theor. Comput. Sci.* **1994**, *123*, 199–237. [[CrossRef](#)]
94. Banks, E. Universality in cellular automata. In Proceedings of the 11th Annual IEEE Symposium on Switching and Automata Theory (SWAT 1970), Santa Monica, CA, USA, 28–30 October 1970; pp. 194–215. [[CrossRef](#)]
95. Cook, M. Universality in Elementary Cellular Automata. *Complex Syst.* **2004**, *15*, 1–40.
96. Durand-Lose, J. *Cellular Automata. Encyclopedia of Complexity and Systems Science*, 2nd ed.; Chapter Universality of Cellular Automata; Springer: New York, NY, USA, 2009. [[CrossRef](#)]
97. Wainwright, R. Life is Universal. In Proceedings of the Winter Simulation Conference: Proceedings of the 7th Conference on Winter Simulations, Washington, DC, USA, 14–16 January 1974; Volume 2, pp. 449–459. [[CrossRef](#)]
98. Guisado, J.; Jiménez-Morales, F.; Guerra, J. Cellular automaton model for the simulation of laser dynamics. *Phys. Rev. E* **2003**, *67*, 066708. [[CrossRef](#)] [[PubMed](#)]
99. 't Hooft, G. *The Cellular Automaton Interpretation of Quantum Mechanics; Physics and Astronomy*; Springer: Cham, Switzerland, 2016; p. 298. [[CrossRef](#)]
100. Dolce, D.; Perali, A. Testing cellular automata interpretation of quantum mechanics in carbon nanotubes and superconductivity. *J. Phys. Conf. Ser.* **2015**, *626*, 012062. [[CrossRef](#)]
101. Emmons-Bells, M.; Durant, F.; Tung, A.; Pietak, A.; Miller, K.; Kane, A.; Martyniuk, C.; Davidian, D.; Morokuma, J.; Levin, M.; et al. Regenerative Adaptation to Electrochemical Perturbation in Planaria: A Molecular Analysis of Physiological Plasticity. *iScience* **2019**, *22*, 147–165. [[CrossRef](#)] [[PubMed](#)]
102. Turing, A. On Computable Numbers, with an Application to the Entscheidungsproblem. *Proc. Lond. Math. Soc.* **1937**, *s2-42*, 230–265; Erratum in *Commun. Proc. Lond. Math. Soc.* **1938**, *s2-43*, 544–546. [[CrossRef](#)]
103. Ulam, S. Random Processes and Transformations. In Proceedings of the International Congress of Mathematicians 1950, Cambridge, MA, USA, 30 August–6 September 1950; Volume 2, pp. 264–275.
104. Turing, A. The chemical basis of morphogenesis. *Philos. Trans. R. Soc. B* **1952**, *237*, 37–72. [[CrossRef](#)]
105. Belousov, B. Periodically acting reaction and its mechanism. In *Sbornik Referatov po Radiacionnoj Medicine (Collection of Reports on Radiation Medicine)*; Publishing House of the State University: New York, NY, USA, 1958; pp. 145–147.
106. Zhabotinskii, A. [Periodic course of the oxidation of malonic acid in a solution (Studies on the kinetics of Belousov's reaction)]. *Biofizika* **1964**, *9*, 306–311.
107. Langtangen, H. *A Primer on Scientific Programming with Python; Texts in Computational Science and Engineering*; Springer: Berlin/Heidelberg, Germany, 2016; p. 922. [[CrossRef](#)]
108. Sundnes, J. *Introduction to Scientific Programming with Python; Simula SpringerBriefs on Computing*; Springer: Cham, Switzerland, 2020; p. 148. [[CrossRef](#)]
109. Kroc, J. The Simplest Python Program Simulating a Cellular Automaton Model of a Complex System: The 'Game of Life'. Available online: <https://www.researchgate.net/publication/355043921> (accessed on 25 March 2023).
110. Theise, N.; Kafatos, M. Sentience Everywhere: Complexity Theory, Panpsychism & the Role of Science in Self-Organization of the Universe. *J. Conscious. Explor. Res.* **2013**, *4*, 378–390.
111. Li, T.; Tang, H.; Zhu, J.; Zhang, J. The finer scale of consciousness: Quantum theory. *Ann. Transl. Med.* **2019**, *7*, 585. [[CrossRef](#)] [[PubMed](#)]
112. Kuhn, T.; Hacking, I. *The Structure of Scientific Revolutions*, 4th ed.; University of Chicago Press: Chicago, IL, USA, 2012; p. 212.
113. Popper, K. *The Logic of Scientific Discovery*; Routledge Classics, Taylor & Francis: New York, NY, USA, 2005; p. 544.
114. Kroc, J. Complex Systems and Their Use in Medicine: Concepts, Methods and Bio-Medical Applications. Preprint on ResearchGate. Available online: [www.researchgate.net/publication/330546521](http://www.researchgate.net/publication/330546521) (accessed on 25 March 2023).
115. Kroc, J. Python Program Simulating Cellular Automaton r-GoL That Represents Robust Generalization of 'Game of Life': Sample Runs. 2022. Available online: <https://www.researchgate.net/publication/357285926> (accessed on 25 March 2023).
116. Kroc, J. Robust Emergents within GoL Resisting Perturbation of Neighborhood. 2023. Available online: <https://www.researchgate.net/publication/372364849/> (accessed on 14 July 2023).
117. Kroc, J. Emergent Computations: Emergents Are Breeding Emergents as Demonstrated on Ships Breeding Trains of Ships Occuring in Modified GoL Using Program GoL-N24. 2023. Available online: <https://www.researchgate.net/publication/368635079/> (accessed on 1 April 2023).
118. Kroc, J. Emergent Computations: Simulations of Logic-Gate OR Using Cellular Automaton GoL-N24 Implemented in Python. 2023. Available online: <https://www.researchgate.net/publication/367380336/> (accessed on 1 March 2023).
119. Kroc, J. Emergent Computations: Simulations of Logic-Gate AND Using Cellular Automaton GoL-N24 Implemented in Python. 2023. Available online: <https://www.researchgate.net/publication/368300518/> (accessed on 25 March 2023).

120. Kroc, J. Emergent Computations: Simulations of Logic-Gate NOT Using Cellular Automaton GoL-N24 Implemented in Python. 2023. Available online: <https://www.researchgate.net/publication/368300097/> (accessed on 25 March 2023).
121. Kroc, J. Logic-Gate OR: Inputs 00. Open file GoL-logic-gate-OR-00-anim.png. 2023. Available online: <https://www.researchgate.net/publication/367380336/> (accessed on 15 September 2023).
122. Kroc, J. Logic-Gate OR: Inputs 01. Open file GoL-logic-gate-OR-01-anim.png. 2023. Available online: <https://www.researchgate.net/publication/367380336/> (accessed on 15 September 2023).
123. Kroc, J. Logic-Gate OR: Inputs 10. Open file GoL-logic-gate-OR-01-anim.png. 2023. Available online: <https://www.researchgate.net/publication/367380336/> (accessed on 15 September 2023).
124. Kroc, J. Logic-Gate OR: Inputs 11. Open file GoL-logic-gate-OR-11-anim.png. 2023. Available online: <https://www.researchgate.net/publication/367380336/> (accessed on 15 September 2023).
125. Kroc, J. Logic-Gate AND: Inputs 00. Open file GoL-AND-00-anim.png. 2023. Available online: <https://www.researchgate.net/publication/368300518/> (accessed on 15 September 2023).
126. Kroc, J. Logic-Gate AND: Inputs 01. Open file GoL-AND-01-anim.png. 2023. Available online: <https://www.researchgate.net/publication/368300518/> (accessed on 15 September 2023).
127. Kroc, J. Logic-Gate AND: Inputs 10. Open file GoL-AND-10-anim.png. 2023. Available online: <https://www.researchgate.net/publication/368300518/> (accessed on 15 September 2023).
128. Kroc, J. Logic-Gate AND: Inputs 11. Open file GoL-AND-11-anim.png. 2023. Available online: <https://www.researchgate.net/publication/368300518/> (accessed on 15 September 2023).
129. Kroc, J. Logic-Gate NOT: Inputs 00. Open file GoL-logic-gate-NOT-0-anim.png. 2023. Available online: <https://www.researchgate.net/publication/368300097/> (accessed on 15 September 2023).
130. Kroc, J. Logic-Gate NOT: Inputs 11. Open file GoL-logic-gate-NOT-1-anim.png. 2023. Available online: <https://www.researchgate.net/publication/368300097/> (accessed on 15 September 2023).

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.