Undecidability and Irreducibility Conditions for Open-Ended Evolution and Emergence

Santiago Hernández-Orozco****
Francisco Hernández-Quiroz**
Universidad Nacional Autónoma
de México

Hector Zenil[†]
University of Oxford
Karolinska Institute
LABORES

Abstract   Is undecidability a requirement for open-ended evolution (OEE)? Using methods derived from algorithmic complexity theory, we propose robust computational definitions of open-ended evolution and the adaptability of computable dynamical systems. Within this framework, we show that decidability imposes absolute limits on the stable growth of complexity in computable dynamical systems. Conversely, systems that exhibit (strong) open-ended evolution must be undecidable, establishing undecidability as a requirement for such systems. Complexity is assessed in terms of three measures: sophistication, coarse sophistication, and busy beaver logical depth. These three complexity measures assign low complexity values to random (incompressible) objects. As time grows, the stated complexity measures allow for the existence of complex states during the evolution of a computable dynamical system. We show, however, that finding these states involves undecidable computations. We conjecture that for similar complexity measures that assign low complexity values, decidability imposes comparable limits on the stable growth of complexity, and that such behavior is necessary for nontrivial evolutionary systems. We show that the undecidability of adapted states imposes novel and unpredictable behavior on the individuals or populations being modeled. Such behavior is irreducible. Finally, we offer an example of a system, first proposed by Chaitin, that exhibits strong OEE.

1   Introduction

An in-depth review of the preliminaries, as well as more information on selected topics, can be found in the extended version of this article [22].

Broadly speaking, a dynamical system is one that changes over time. Prediction of the future behavior of dynamical systems is a fundamental concern of science generally. Scientific theories are tested upon the accuracy of their predictions, and establishing invariable properties through the evolution of a system is an important goal. Limits to this predictability are known in science. For instance, chaos theory establishes the existence of systems in which small deficits in the information of the initial states make accurate predictions of future states unattainable. However, in this

---

 * Contact author.

** Department of Mathematics, Universidad Nacional Autónoma de México, Ciudad Universitaria, Ciudad de México, México 04510. Posgrado en Ciencias e Ingeniería de la Computación, Universidad Nacional Autónoma de México. E-mail: hosant@ciencias.unam.mx (S.H.-O.), fhq@ciencias.unam.mx (F.H.-Q.)

 † Algorithmic Dynamics Lab, Unit of Computational Medicine, SciLifeLab, Karolinska Institute, Karolinska Hospital L8:05, SE-171 76, Stockholm, Sweden. E-mail: hector.zenil@algorithmicnaturelab.org

article we focus on systems for which we have unambiguous, finite (as to size and time), and complete descriptions of initial states and behavior: computable dynamical systems.

Since their formalization by Church and Turing, the class of computable systems has shown that, even without information deficits (i.e., with complete descriptions), there are future states that cannot be predicted, in particular the state known as the *halting state* [34]. We will use this result and others from algorithmic information theory to show how predictability imposes limits on the growth of complexity during the evolution of computable systems. In particular, we will show that random (incompressible) times tightly bound the complexity of the associated states.

The relationship between dynamical systems and computability has been studied before by Bournez [10], Blondel [9], Moore [29], and Fredkin, Margolus, and Toffoli [20, 28], among others. That emergence is a consequence of incomputability has been proposed by Cooper [17]. Complexity as a source of undecidability has been observed in logic by Calude and Jurgensen [11]. Delvenne, Kurka, and Blondel [19] have proposed robust definitions of computable (effective) dynamical systems and universality, generalizing Turing's halting states, while also setting forth the conditions and implications for universality and decidability and their relationship with chaos. The definitions and general approach used in this paper differ from those in the sources cited above, but are ultimately related.

We will denote by $K(x|y)$ the algorithmic descriptive complexity of the string x with respect to the string y. The dynamical systems we are considering are deterministic, and each state must contain all the information needed to compute successive states. We are assuming an infinity of possible states for non-cyclical systems. Mechanisms and requirements for open-ended evolution in systems with a finite number of states (resource-bounded) have been studied by Adams et al. [3].

1.1 Open-Ended Evolution in Computable Dynamical Systems

Informally, *open-ended evolution* (OEE) has been characterized as "evolutionary dynamics in which new, surprising, and sometimes more complex organisms and interactions continue to appear" [34, p. 1]. Establishing and defining the properties required for a system to exhibit OEE is considered an open question [7, 31, 32], and OEE has been proposed as a required property of evolutionary systems capable of producing life [30]. This has been implicitly verified by various experiments *in silico* [27, 2, 26, 5].

One line of thought posits that open-ended evolutionary systems tend to produce families of objects of increasing *complexity* [6, 5]. Furthermore, for a number of complexity measures, it can be shown that the objects belonging to a given level of complexity are finite in number (for instance $K(x)$). Therefore an increase of complexity is a requirement for the continued production of new objects. A related observation, proposed by Chaitin [16, 15], associates evolution with the search for *mathematical creativity*, which implies an increase of complexity, as more complex mathematical operations are needed in order to solve interesting problems, which are required to drive evolution.

Following the aforementioned lines of thought, we have chosen to characterize OEE in computable dynamical systems as a process that has the property of producing families of objects of increasing complexity. Formally, given a *complexity measure C*, we say that a computable dynamical system $S(M_0, t)$, where $M_t = S(M_0, t) = S(M_{t-1}, 1) = S(M_t, 0)$ is the state of the system at the time t, exhibits *open-ended evolution* with respect to C if for every time t there exists a time t' such that the complexity of the system at time t' is greater than the complexity at time t, that is, $C(S(M_0, t)) < C(S(M_0, t'))$, where a complexity measure is a (not necessarily computable) function that goes from the state space to a positive numerical space.

The existence of such systems is trivial for complexity measures on which any infinite set of natural numbers (not necessarily computable) contains a subset where the measure grows strictly:

Lemma 1.1. *Let C be a complexity measure such that any infinite set of natural numbers has a subset where C grows strictly. Then a computable system $S(M_0, t)$ is a system that produces an infinite number of different states if and only if it exhibits OEE for C.*

Proof. Let $S(M_0, t)$ be a system that does not exhibit OEE, and C a complexity measure as described. Then there exists a time t such that for any other time t' we have $C(M_{t'}) \leq C(M_t)$, which holds true for any subset of states of the system. It follows that the set of states must be finite. Conversely, if the system exhibits OEE, then there exists an infinite subset of states on which S grows strictly, hence an infinity of different states.   □

Given the previous lemma, a trivial computable system that simply produces all the strings in order exhibits OEE on a class of complexity measures that includes algorithmic description complexity. However, we intuitively conjecture that such systems have a much simpler behavior than that observed in the natural world and the artificial life systems referenced. To avoid some of these issues we propose a stronger version of OEE.

Definition 1.2. A sequence $n_0, n_1, \ldots, n_i, \ldots$ exhibits *strong open-ended evolution* (strong OEE) with respect to a complexity measure C if for every index i there exists an index i' such that $C(n_i) < C(n_{i'})$, and the sequence of complexities $C(n_0), C(n_1), \ldots, C(n_i), \ldots$ does not drop *significantly*, that is, there exists a γ such that $i \leq j$ implies $C(n_i) \leq C(n_j) + \gamma(j)$ where $\gamma(j)$ is a positive function that does not grow *significantly*.

It is important to note that while the definition of OEE allows for significant drops in complexity during the evolution of a system, strong OEE requires that the complexity of the system not decrease *significantly* during its evolution. In particular we will require that the complexity drops as measured by γ not grow as fast as the complexity itself. Formally $C(n_j) - \gamma(j)$ should not be upper-bounded for any infinite subsequence for the smallest γ where the strong OEE inequality holds.

We will construe the concept of *speed* of growth of complexity in a comparative way: Given two sequences of natural numbers n_i and m_i, n_i *grows faster* than m_i if for every infinite subsequence and natural number N, there exists j such that $n_i - m_j \geq N$. Conversely, a subsequence of indexes denoted by i grows faster than a subsequence of indexes denoted by j if for every natural N, there exists i with $i < j$ such that $n_i - n_j \geq N$.

If a complexity measure is sophisticated enough to depend on more than just the size of an object, significant drops in complexity are a feature that can be observed in trivial sequences such as the ones produced by enumeration machines. Whether this is also true for *nontrivial* sequences is open to debate. However, if we classify random strings as low-complexity objects and posit that nontrivial sequences must contain a limited number of random objects, then a nontrivial sequence must observe bounded drops in complexity in order to be capable of showing nontrivial OEE. This is the intuition behind the definition of strong OEE.

Now, in the literature on dynamical systems, random objects are often considered simple ([1, p. 1]), with complexity being taken to lie between regularity and randomness. Various complexity measures have been proposed that assign low complexity to random or incompressible natural numbers. Two examples of such measures are logical depth [8] and sophistication [25]. Classifying random naturals as low-complexity objects is a requirement for the results shown in Section 3.

2   A Computational Model for Adaptation

Let's start by describing the evolution of an organism or a population by a computable dynamical system. It has been argued that in order for adaptation and survival to be possible, an organism must contain an effective representation of the environment, so that, given a reading of the environment, the organism can choose a behavior accordingly [35]. The more faithful this representation, the better the adaptation. If the organism is computable, this information can be codified by a computable structure. We will denote this structure by M_t, where t stands for the time corresponding to each of the stages of the evolution of the organism. This information is then processed following a finitely

specified unambiguous set of rules that, in finite time, will determine the adapted behavior of the organism according to the information codified by M_t. We will denote this behavior (or a theory explaining it) using the program p_t. An adapted system is one that produces an acceptable approximation of its environment. An environment can also be represented by a computable structure E. In other words, the system is adapted if $p_t(M_t)$ produces E. Based on this idea, we propose a robust, formal characterization for adaptation, which presents a necessary (but not sufficient) condition:

Definition 2.1. Let K be the prefix-free descriptive complexity. We say that the system at the state M_t is *ϵ-adapted* to E if $K(E \,|\, S(M_0, E, t)) \leq \epsilon$.

The inequality states that the minimal amount of information that is needed to describe E from a complete description of M_t is ϵ or less. This information is provided in the form of a program p that produces E from the system at time t. We will define such a program p as the *adapted behavior* of the system. It is not required that p be unique.

The proposed structure for adapted systems is robust, since $K(E \,|\, S(M_0, E, t))$ is less than or equal to the number of characters needed to describe any computable method of describing E from the state of the system at time n, whether it be a computable theory for adaptation or a computable model for an organism that tries to predict E. It follows that any computable characterization of adaptation that can be described within ϵ bits meets the definition of ϵ-adapted, given a suitable choice of E, the *adaptation condition* for any given environment. It is important to note that, although inspired by a representationalist approach to adaptation, the proposed characterization of adaptation is not contingent on the organism's containing an actual codification of the environment, since any organism that can produce an adapted behavior that can be explained effectively (is computable in finite time) is ϵ-adapted for some ϵ.

As a simple example, we can think of an organism that must find food located at the coordinates (x, j) on a grid in order to survive. If the information in an organism is codified by a computable structure M (such as DNA), and there is a set of finitely specified, unambiguous rules that govern how this information is used (such as the ones specified by biochemistry and biological theories), codified by a program p, then we say that the organism finds the food if $p(M) = (j, k)$. If $|\langle p \rangle| \leq \epsilon$, then we say that the organism is adapted according to a behavior that can be described within ϵ characters. The proposed model for adaptation is not limited to such simple interactions. For a start, we can suppose that the organism *sees* a grid, denoted by g, of size $n \times m$ with food at the coordinates (j, k). The environment can be codified as a function E such that $E(g) = (j, k)$, and being ϵ-adapted implies that the organism defined by the genetic code M, which is interpreted by a theory or behavior written on ϵ bits, is capable of finding the food upon seeing g. Similarly, more complex computational structures and interactions imply ϵ-adaptation.

Now, describing an evolutionary system that (eventually) produces an *ϵ-adapted* system is trivial via an enumeration machine (the program that produces all the natural numbers in order), as it will eventually produce E itself. Moreover, we require the output of our process to remain adapted. Therefore we propose a stronger condition called *convergence*:

Definition 2.2. Given the description of a computable dynamical system $S(M_0, E, t)$, where $t \in \mathbb{N}$ is the variable of time, M_0 is an initial state, and E is an environment, we say that the system S *converges* towards E with degree ϵ if there exists δ such that $t \geq \delta$ implies $K(E \,|\, S(M_0, E, t)) \leq \epsilon$.

For a fixed initial state M_0 and environment E, it is easy to see that the descriptive complexity of a state of the system depends mostly on t: We can describe a program that, given full descriptions of S, E, M_0, and t, finds $S(M_0, E, t)$. Therefore

$$K(S(M_0, E, t)) \leq K(S) + K(E) + K(M_0) + K(t) + O(1), \tag{1}$$

where the constant term is the length of the program described. In other words, as the time t grows, *time becomes the main driver for the descriptive complexity within the system.*

2.1 Irreducibility of Descriptive Time Complexity

In the previous paragraph, it was established that time was the main factor in the descriptive complexity of the states within the evolution of a system. This result is expanded by the time complexity stability theorem (Theorem 2.3). This theorem establishes that, within an algorithmic descriptive complexity framework, similarly complex initial states must evolve into similarly complex future states over similarly complex time frames, effectively erasing the difference between the complexity of the state of the system and the complexity of the corresponding time, and establishing absolute limits to the reducibility of future states.

Let $F(t) = T(S(M_0, E, t))$ be the *real execution time* of the system at time t. Using our time counting machine U^H, it is easy to see that $F(t)$ is computable and, given the uniqueness of the successor state, F increases strictly with t, and hence is injective. Consequently, F has a computational inverse F^{-1} over its image. Therefore, we have it that (up to a small constant) $K(F(t)) \leq K(F) + K(t)$ and $K(t) \leq K(F^{-1}) + K(F(t))$. It follows that $K(t) = K(F(t)) + O(c)$, where c is an integer independent of t (but that can depend on S). In other words, for a fixed system S, the execution time and the system time are *equally complex up to a constant*. From here on we will not differentiate between the complexities of the two times. A generalization of the previous equation is given by the following theorem:

Theorem 2.3 (Time complexity stability). *Let S and S' be two computable systems, and t and t' the first time where the systems reach the states M_t and $M'_{t'}$, respectively. Then there exists c such that $|K(M_t) - K(t)| \leq c$ and $|K(M_t) - K(M'_{t'})| \leq c$.*

Proof. First, note that we can describe a program such that given S, M_0, and E, we have that $S(M_0, E, x)$ runs for each x until it finds t. Therefore $K(t) \leq K(S(M_0, E, t)) + K(S) + K(M_0) + K(E) + O(1)$. Similarly for t'. By the inequality (1) and the hypothesized equalities we obtain $K(t) - (K(S) + K(M_0) + K(E) + O(1)) \leq K(M_t) \leq K(t) + (K(S) + K(E) + K(M_0) + O(1))$. □

The slow growth of time is a possible objection to the assertion that in the descriptive complexity of systems time is the dominating parameter for predicting their evolution: The function $K(t)$ grows within $O(\log t)$, which is very slow and often considered insignificant in the information theory literature. However, we have to consider the scale of time we are using. For instance, one second of *real time* in the system we are modeling may mean an exponential number of discrete time steps for our computable model (for instance, if we are modeling a genetic machine with current computer technology), yielding a potential polynomial growth in their descriptive complexity. However, if this time conversion is computable, then $K(t)$ grows at most at a constant pace. This is an instance of *irreducibility*, as there exist infinite sequences of times that cannot be obtained by computable methods. In the upcoming subsections we will call such times *random times* and the sequences containing them will be deemed *irreducible*.

2.2 Non-randomness of Decidable Convergence Times

One of the most important tasks for science is predicting the future behavior of dynamical systems. The prediction we will focus on is about the first state of convergence (Definition 2.2): Will a system converge, and how long will it take? In this section we shall show the limit that decidability imposes on the complexity of the first convergent state. A consequence of this is the existence of undecidable adapted states.

Formally, for the convergence of a system S with degree ϵ to be decidable, there must exist an algorithm D_ϵ such that $D_\epsilon(S, M_0, E, \delta) = 1$ if the system is convergent at time δ and 0 otherwise.

Moreover, we can describe a machine P such that, given full descriptions of D_ϵ, S, and M_0, it runs D_ϵ with inputs S and M_0 while running over all the possible times t, returning the first t for which the system converges. Note that $\delta = P(\langle D_\epsilon \rangle \langle S \rangle \langle M_0 \rangle \langle E \rangle)$. Hence we have a short description of δ, and therefore δ *cannot be random*: If $S(M_0, E, t)$ is a convergent system, then

$$K(\delta) \leq K(D_\epsilon) + K(S) + K(E) + K(M_0) + O(1), \tag{2}$$

where δ is the first time at which convergence is reached. Note that all the variables are known at the initial state of the system. This result can summed up by the following lemma:

Lemma 2.4. *Let S be a system convergent at time δ. If δ is considerably more descriptively complex than the system and the environment—that is, if for every reasonably large natural number d we have that $K(\delta) > K(S) + K(E) + K(M_0) + d$—then δ cannot be found by an algorithm described within d characters.*

Proof. It is a direct consequence of the inequality (2). □

We call such times *random convergence times*, and the state of the system M_δ a *random state*. It is important to note that the descriptive complexity of a random state must also be high:

Lemma 2.5. *Let S be a convergent system with a complex state $S(M_0, E, \delta)$. For every reasonably large d we have*

$$K(S(M_0, E, \delta)) > K(S) + K(E) + K(M_0) + d.$$

Proof. Suppose the contrary to be true, that is, that there exists d small enough that $K(S(M_0, E, \delta)) \leq K(S) + K(E) + K(M_0) + d$. Let q be the program that, given S, E, M_0, and $S(M_0, E, \delta)$, runs $S(M_0, E, t)$ in order for each t and compares the result with $S(M_0, E, \delta)$, returning the first time where the equality is reached. Therefore, given that the succesor states are unique, we have $\delta = q(S, M_0, E, S(M_0, E, \delta))$ and $K(\delta) \leq K(S) + K(E) + K(M_0) + K(S(M_0, E, \delta)) + |q| \leq K(S) + K(E) + K(M_0) + (K(S) + K(E) + K(M_0) + d) + O(1)$, which gives us a small upper bound to the random convergence time δ. □

In other words, if δ has high descriptive complexity, then there does not exist a reasonable algorithm that finds it, even if we have a complete description of the system and its environment. It follows that the descriptive complexity of a computable convergent state cannot be much greater than the descriptive complexity of the system itself.

What a *reasonably large d* is has been handled so far with ambiguity, as it represents the descriptive complexity of any computable method D_ϵ. We may intend to find convergence times, which intuitively cannot be arbitrarily large. It is easy to "cheat" on the inequality (2) by including in the description of the program D_ϵ the full description of the convergence time δ, which is why we ask for *reasonable* descriptions.

Another question left to be answered is whether complex convergence times do exist for a given limit d, considering that the limits imposed by the inequality (2) loosen up in direct relation to the descriptive complexity of S, E, and M_0.

The next result answers both questions by proving the existence of complex convergence times for a broad characterization of the size of d:

Lemma 2.6 (Existence of random convergence times). *Let F be a total computable function. For any ϵ there exists a system $S(M_0, E, t)$ such that the convergence times are $F(S, M_0, E)$-random.*

Proof. Let E and s be two natural numbers such that $K(E|s) > \epsilon$. By reduction to the halting problem [34], it is easy to see the existence of $F(S, M_0, E)$-random convergence times: Let T be a Turing machine, and S_t the Turing machine that emulates T for t steps with input M_0 and returns E

for every time equal to or greater than the halting time, and s otherwise. Let us consider the system $S(M_0, E, t) = S_t(\langle T \rangle \langle M_0 \rangle \langle t \rangle \langle E \rangle)$.

If the convergence times are not $F(S, M_0, E)$-random, then there exists a constant c such that we can decide HP by running S for each t that meets the inequality $|t| + 2 \log|t| + c \leq |S| + |\langle T \rangle \langle M_0 \rangle \langle t \rangle \langle E \rangle| + F(S, M_0, E)$,[1] which cannot be done, since HP is undecidable. □

Let us focus on what the previous lemma is saying: F can be any computable function. It can be a polynomial or exponential function with respect to the length of a given description for M_0 and E. It can also be any computable theory that we might propose for setting an upper limit to the size of an algorithm that finds convergence times given descriptions of the system's behavior, environment, and initial state. In other words, for a class of dynamical systems, finding convergence times, and therefore convergent states, is not decidable, even with complete information about the system and its initial state. Finally, by the proof of the lemma, adapted states can be seen as a generalization of halting states.

2.3 Randomness of Convergence in Dynamic Environments

So far we have limited the discussion to fixed environments. However, as observed in the physical world, the environment itself can change over time. We call such environments *dynamic environments*. In this subsection we extend the previous results to cover environments that change depending on time as well as on the initial state of the system. We also propose a weaker convergence condition, called *weak convergence*, and propose a necessary (but not sufficient) condition for the computability of convergence times, called *descriptive differentiability*.

We can think of an environment E as a dynamic computable system, a moving target that also changes with time and depends on the initial state M_0. In order for the system to be convergent, we propose the same criterion—there must exist δ such that $n \geq \delta$ implies $K(E(M_0, n) | S(M_0, E(M_0, n), n)) \leq \epsilon$. A system with a dynamic environment also meets the inequality (2) and Lemmas 2.4 and 2.6, since we can describe a machine that runs both S and E for the same time t. Given that E is a moving target, it is convenient to consider an *adaptation period* for the new states of E:

Definition 2.7. We say that S *converges weakly* to E if there exist an infinity of times δ_i such that

$$K(E(M_0, \delta_i) \mid S(M_0, E(M_0, \delta_i), \delta_i)) \leq \epsilon. \tag{3}$$

As a direct consequence of the inequality (2) and Lemma 2.6 we have the following lemma:

Lemma 2.8. *Let $S(M_0, E(M_0, t), t)$ be a weakly converging system. Any decision algorithm $D_\epsilon(S, M_0, E, \delta_i)$ can only decide the first nonrandom time.*

As noted above, these results do not change when dynamic environments are considered. In fact, we can think of static environments as a special case of dynamic environments. However, with different targets of adaptability and convergence, it makes sense to generalize beyond the first convergence time. Also, it should be noted that specifying a convergence index adds additional information that a decision algorithm can potentially use.

Lemma 2.9. *Let $S(M_0, E(M_0, t), t)$ be a weakly converging system with an infinity of random times such that $k > j$ implies that $K(\delta_k) = K(\delta_j) + \Delta K_\delta(j, k)$, where ΔK_δ is a (not necessarily computable) function with a range confined to the positive integers. If the function $\Delta K_\delta(i, i + m)$ is unbounded with respect to i, then any decision algorithm $D_\epsilon(S, M_0, E, i)$, where i is the i-th convergence time, can only decide a finite number of i's.*

---

1 For any string s there exists a self-delimited program (by a "print") that takes a prefix-free input of the form $1^{\log|s|}0|s|s$.

Proof. Suppose that $D_\epsilon(S, M_0, E, i)$ can decide an infinite number of instances. Let us consider two times δ_i and δ_{i+m}. Note that we can describe a program that, by using D_ϵ, S, E, M_0, and i together with the distance m, finds δ_{i+m}. The next inequality follows: $K(\delta_{i+m}) \leq K(D_\epsilon) + K(i) + K(m) + O(1)$. Next, note that we can describe another program that, given δ_i and using D_ϵ, S, E, and M_0, finds i, from which $K(i) \leq K(D_\epsilon) + K(\delta_i) + O(1)$ and $-K(\delta_i) \leq K(D_\epsilon) - K(i) + O(1)$. Therefore $\Delta K_\delta(i, i+m) = K(\delta_{i+m}) - K(\delta_i) \leq 2K(D_\epsilon) + K(m) + O(1)$, and $\Delta K_\delta(i, i+m)$ is bounded with respect to i. □

We will say that a sequence of times $\delta_1, ..., \delta_i, ...$ is *non-descriptively differentiable* if ΔK_δ is not a total function, which, as a consequence of the previous lemma, implies noncomputability of the sequence.

3 Beyond Halting States: Open-Ended Evolution

The inequality (2) states that being able to predict or recognize adaptation imposes a limit on the descriptive complexity of the first adapted state. A particular case is the halting state, as shown in the proof of Lemma 2.6. In this section we extend the lemma to continuously evolving systems, showing that computability of adapted times limits the complexity of adapted states beyond the first, imposing a limit to open-ended evolution for three complexity measures: sophistication, coarse sophistication, and busy beaver logical depth.

For a system in constant evolution converging to a dynamic environment, Lemma 2.9 imposes a limit on the growth of the descriptive complexity of a system with computable adapted states: *If the growth of the descriptive complexity of a sequence of convergent times is unbounded in the sense of Lemma 2.9, then all but a finite number of times are undecidable.* The converse would be convenient; however, it is not always true. Moreover, the next series of results shows that imposing such a limit would impede strong OEE:

Theorem 3.1. *Let S be a non-cyclical computable system with initial state M_0, E a dynamic environment, and $\delta_1, ..., \delta_i, ...$ a sequence of times such that for each δ_i there exists a total function p_i such that $p_i(M_{\delta_i}) = E(\delta_i)$. If the function $p : i \mapsto p_i$ is computable, then the function $\delta : i \mapsto \delta_i$ is computable.*

Proof. Assume that p is computable. We can describe a program D_ϵ that, given S, M_0, δ_i, and E, runs $p_{\delta_i}(M_t)$ and $E(t)$ for each time t, returning 1 if the δ_ith t is such that $p_{\delta_i}(t) = E(t)$, and 0 otherwise. Therefore the sequence of δ_i's is computable. □

The last result can be applied naturally to weakly convergent systems (Definition 2.7): The way each adapted state approaches E is unpredictable; in other words, its *behavior* changes unpredictably over different stages. Formally:

Corollary 3.2. *Let $S(M_0, E, t)$ be a weakly converging system, with adapted states $M_{\delta_1}, ..., M_{\delta_i}, ...$ and $p_1, ..., p_i, ...$ its respective adapted behaviors. If the mapping $\delta : i \mapsto \delta_i$ is not computable, then the function $p : i \mapsto p_i$ is also not computable.*

Proof. It is a direct consequence of applying Theorem 3.1 to the definition of weakly converging systems. □

While asking for totality might look like an arbitrary limitation at first glance, the reader should recall that in weakly convergent systems the program p_i represents an organism, theory, or other computable system that uses M_{δ_i}'s information to predict the behavior of $E(\delta_i)$, and if this prediction does not process its environment in a sensible time frame, then it is hard to argue that it represents an adapted system or a useful theory.

The intuition behind classifying descriptively differentiable adapted time sequences as *less complex* is better explained by borrowing ideas developed by Bennett and Koppel, within the framework of

logical depth [8] and sophistication [25], respectively. Their argument states that random strings are as simple as very regular strings, given that there is no complex underlying structure in their minimal descriptions. The intuition that random objects contain no useful information leads us to the same conclusion. And, given Theorem 2.3, the states must retain a high degree of randomness for random times.

Sophistication is a measure of useful information within a string. Proposed by Koppel, the underlying approach consists in dividing the description of a string x into two parts: the program that represents the *underlying structure* of the object, and the input, which is the random or *structureless* component of the object. This function is denoted by $\mathrm{soph}_c(x)$, where c is a natural number representing the significance level.

Definition 3.3. The *sophistication* of a natural number x at the significance level c, $c \in \mathbb{N}$, is defined as

$$\mathrm{soph}_c(x) = \min\{|\langle p\rangle| : p \text{ is a total function and } \exists y . p(y) = x \text{ and } |\langle p\rangle| + |y| \leq K(x) + c\}.$$

Now, the images of a mapping $\delta : i \mapsto \delta_i$ already have the form $\delta(i)$, where δ and i represent the structure and the random component respectively. Random strings should bind this structure strongly up to a logarithmic error, which is proven in the next lemma.

Lemma 3.4. *Let $\delta_1, \ldots, \delta_i, \ldots$ be a sequence of different natural numbers, and r a natural number. If the function $\delta : i \mapsto \delta_i$ is computable, then there exists an infinite subsequence where the sophistication is bounded up to a logarithm of a logarithmic term of their indexes.*

Proof. Let δ be a computable function. Note that since δ is computable and the sequence is composed of different naturals, its inverse function δ^{-1} can be computed by a program m that, given a description of δ and δ_i, finds the first i that produces δ_i and returns it; therefore $K(i) \leq K(\delta_i) + |\langle m\rangle| + |\langle\delta\rangle|$ and $K(\delta) + K(i) \leq K(\delta_i) + |\langle m\rangle| + 2|\langle\delta\rangle|$. Now, if i is an r-random natural where the inequality holds tightly, we have that $(K(\delta) + O(\log|i|)) + |i| - r \leq K(\delta_i) + |\langle m\rangle| + 2|\langle\delta\rangle|$, which implies that, since δ is a total function, $\mathrm{soph}_{|\langle m\rangle|+2|\langle\delta\rangle|+r}(\delta_i) \leq K(\delta) + O(\log\log i)$. Therefore, the sophistication is bounded up to a logarithm of a logarithmic term for a constant significance level for an infinite subsequence. □

Small changes in the significance level of sophistication can have a large impact on the sophistication of a given string. Another possible problem is that the constant $|\langle m\rangle| + 2|\langle\delta\rangle| + r$ implied in Lemma 3.4 could appear to be large at first (but it becomes comparatively smaller as i grows). A *robust* variation of sophistication, called coarse sophistication [4], incorporates the significance level as a penalty. The definition presented here differs slightly from theirs in order to maintain congruence with the chosen prefix-free universal machine and to avoid negative values. This measure is denoted by $\mathrm{csoph}(x)$.

Definition 3.5. The *coarse sophistication* of a natural number x is defined as

$$\mathrm{csoph}(x) = \min\{2|\langle p\rangle| + |\langle y\rangle| - K(x) : p(y) = x \text{ and } p \text{ is total}\},$$

where $|\langle y\rangle|$ is a computable unambiguous codification of y.

With a similar argument to the one used to prove Lemma 3.4, it is easy to show that coarse sophistication is similarly bounded up to an algorithm of a logarithmic term.

Lemma 3.6. *Let $\delta_1, \ldots, \delta_i, \ldots$ be a sequence of different natural numbers, and r a natural number. If the function $\delta : i \mapsto \delta_i$ is computable, then there exists an infinite subsequence where the coarse sophistication is bounded up to a logarithm of a logarithmic term.*

Proof. If δ is computable and i is r-random, then by definition of csoph and the inequalities presented in the proof of Lemma 3.4, we have that

$$
\begin{aligned}
\mathrm{csoph}(\delta_i) &\leq 2K(\delta) + (|i| + 2\log|i| + 1) - K(\delta_i) \\
&\leq 2K(\delta) + (|i| + 2\log|i| + 1) - K(i) + |\langle M \rangle| + |\langle \delta \rangle| \\
&\leq 2K(\delta) + |\langle M \rangle| + |\langle \delta \rangle| + (|i| + 2\log|i| + 1) - |i| + r \\
&= 2K(\delta) + |\langle M \rangle| + |\langle \delta \rangle| + r + 1 + O(\log\log(i)). \qquad \square
\end{aligned}
$$

Another proposed measure of complexity is Bennett's logical depth [8], which measures the minimum computational time required to compute an object from a nearly minimal description. Logical depth works under the assumption that complex, or *deep*, natural numbers take a long time to compute from near-minimal descriptions. Conversely, random or incompressible strings are *shallow*, since their minimal descriptions must contain the full description verbatim. For the next result we will use a related measure called busy beaver logical depth, denoted by $\mathrm{depth}_{\mathrm{bb}}(x)$.

Definition 3.7. The *busy beaver logical depth* of the description of a natural x, denoted by $\mathrm{depth}_{\mathrm{bb}}(x)$, is defined as

$$
\mathrm{depth}_{\mathrm{bb}}(x) = \min\{|p| - K(x) + j : U(p) = x \text{ and } T(p) \leq \mathrm{BB}(j)\},
$$

where $T(P)$ is the halting time of the program p, and $\mathrm{BB}(j)$, known as the busy beaver function, is the halting time of the slowest program that can be described within j bits [18].

The next result follows from a theorem formulated by Antunes and Fortnow [4] and from Lemma 3.6.

Corollary 3.8. *Let $\delta_1, \ldots, \delta_i, \ldots$ be a sequence of different natural numbers, and r a natural number. If the function $\delta : i \mapsto \delta_i$ is computable, then there exists an infinite subsequence where the busy beaver logical depth is bounded up to an algorithm of a logarithmic term of their indexes.*

Proof. By Theorem 5.2 in [4], for any i we have $|\mathrm{csoph}(\delta_i) - \mathrm{depth}_{\mathrm{bb}}(\delta_i)| \leq O(\log|\delta_i|)$. By Lemma 3.6 and Theorem 2.3 the result follows. $\qquad \square$

Let us focus on the consequence of Lemmas 3.4 and 3.6 and Corollary 3.8. Given the relationship established between descriptive time complexity and the corresponding state of a system (Theorem 2.3), these last results imply that either the complexity of the adapted states of a system (using any of the three complexity measures) grows very slowly for an infinite subsequence of times (becoming increasingly common up to a probability limit of 1 [13]), or the subsequence of adapted times is undecidable.

Theorem 3.9. *If $S(M_0, E(t), t)$ is a weakly converging system with adaptation times $\delta_1, \ldots, \delta_i, \ldots$ that exhibits strong OEE with respect to csoph and $\mathrm{depth}_{\mathrm{bb}}$, then the mapping $\delta : i \mapsto \delta_i$ is not computable. Also, there exists a constant c such that the result applies to soph_c.*

Proof. We can see the sequence of adapted states as a function $M_{\delta_i} : i \mapsto M_{\delta_i}$. By Lemmas 3.4 and 3.6 and Corollary 3.8, for the three stated measures of complexity, there exists an infinite subsequence where the respective complexity is upper-bounded by $O(\log\log i)$. It follows that if the complexity grows faster than $O(\log\log i)$ for an infinite subsequence, then there must exist an infinity of indexes j in the bounded succession where $\gamma(j)$ grows faster than $C(M_j)$. Therefore there exists an infinity of indexes j where $C(M_j) - \gamma(j)$ is upper bounded. Finally, note that if a

computable mapping $\delta : i \mapsto \delta_i$ allowed growth on the order of $O(\log \log i)$, then the computable function $\delta' : i \mapsto \delta_{2^i}$ would grow faster than the stated bound. □

Now, in the absence of absolute solutions to the problem of finding adapted states in the presence of strong OEE, one might cast about for a partial solution or approximation that decides most (or at least some) of the adapted states. The following corollary shows that the problem is not even semicomputable: *Any algorithm one might propose can only decide a bounded number of adapted states.*

Corollary 3.10. *If $S(M_0, E, t)$ is a weakly converging system with adapted states $M_1, \ldots, M_i, \ldots$ that show strong OEE, then the mapping $\delta : i \mapsto \delta_i$ is not even semicomputable.*

Proof. Note that for any subsequence of adaptation times $\delta_{j_1}, \ldots, \delta_{j_k}, \ldots$, the system must show strong OEE. Therefore, by Theorem 3.9, any subsequence must also not be computable. It follows that there cannot exist an algorithm that produces an infinity of elements of the sequence, since such an algorithm would allow the creation of a computable subsequence of adaptation times. □

In short, Theorem 3.9 imposes undecidability on strong OEE, and, according to Theorem 3.2, the behavior and interpretation of the system evolves in an unpredictable way, establishing one path for emergence: *a set of rules for future states that cannot be reduced to an initial set of rules.* Recall that for a given weakly converging dynamical system, the sequence of programs p_i represents the behavior or interpretation of each adapted state M_i. If a system exhibits strong OEE with respect to the complexity measure $soph_c$, $csoph$, or $depth_{bb}$, then by Corollary 3.2 and Theorem 3.9 the sequence of behaviors is uncomputable, and therefore irreducible to any function of the form $p : i \mapsto p_i$, even when possessing complete descriptions for the behavior of the system, its environment, and its initial state. In other words, *the behavior of iterative adapted states cannot be obtained from the initial set of rules.* Furthermore, we conjecture that the results hold for all adequate measures of complexity:

Conjecture 3.11. *Computability bounds the growing complexity rate to that of an order of the slowest-growing infinite subsequence with respect to any* adequate *complexity measure C.*

One way to understand Conjecture 3.11 is that *the information of future states of a system is either contained in the initial state—hence their complexity is bounded by that initial state—or is undecidable.* This should be a consequence, given that, for any computable dynamical system, the randomness induced by time cannot be avoided.

3.1 A System Exhibiting OEE

With the aim of providing mathematical evidence for the adequacy of Darwinian evolution, Chaitin developed a mathematical model that converges to its environment significantly faster than exhaustive search, being fairly close to an *intelligent* solution to a mathematical problem that requires *maximal creativity* [16, 15].

One of the solutions Chaitin proposes is to find digital organisms that approximate the busy beaver function $BB(n) = \max\{T(U(p)) : |p| \leq n\}$, which is equivalent (up to a constant) to asking for the largest natural number that can be named within n bits and for the first n bits of Chaitin's constant, which is defined as $\Omega_U = \Sigma_{T \in HP} 2^{-|T|}$, where HP is the set of all halting Turing machines for the universal machine U. We will omit the subindex from Ω in the rest of this text.

Chaitin's evolutionary system searches nondeterministically through the space of Turing machines using a reference universal machine U' with the property that all strings are valid programs. This random walk starts with the empty string $M_0 = $ " ", and each new state is defined as the output of a Turing machine, called a *mutation*, with the previous state as an input. These mutations are chosen stochastically according to the universal distribution [24]. If these mutations help to

more accurately approximate the digits of Ω, then this program becomes the new state M_{t+1}; otherwise we keep searching for new organisms. Chaitin demonstrates that the system approaches Ω *efficiently* (with quadratic overhead), arguing that this is evidence of the adequacy of Darwinian evolution [14].

Given that Ω can be used to compute BB(n) [21], a deterministic version of Chaitin's system is the following:

$$M_0 = 0,$$
$$M_t = p \cdot T(p) = \max\{T(U'(q)) : H(M_{t-1}, q) \leq w\},$$

where $H(M_{t-1}, p)$ is the *distance* between the programs, M_{t-1}, q is the quantification of the number of mutations needed to transform one string into the other, and w is a positive integer acting as an accumulator that resets to 1 whenever M_t increases in value, adding 1 otherwise.

Defining a computable environment or adaptation condition for this system is difficult, since the system seeks to approach an uncomputable function (BB) and the evolution rule itself is not computable, given the halting problem. The most direct way to define it is $E(t) = \mathrm{BB}(t)$ or, equivalently, as the first t bits of Chaitin's constant Ω. Another way to define the environment is by an encoding of the property *larger than* $U(M_{t-1})$ for each time t. Given that we can compute M_{t-1} and its relationship with M_t given a description of the latter and a constant amount of information (ϵ), we find adaptation at the times t where the busy beaver function grows.

It is easy to see that the sequence of programs $i \mapsto M_i$ is precisely what generates the busy beaver sequence $\eta_i = \mathrm{BB}(i)$. Given that $\mathrm{BB}(t)$ is not a computable function, the evolution of the system, along with the respective adaptation times, is not computable. Furthermore, this sequence is composed of programs that compute, in order, an element of a sequence that exhibits strong OEE with respect to depth$_\mathrm{bb}$: Let $\eta_i = \mathrm{BB}(i)$ be the sequence of all busy beaver values; by definition, if i is the first value for which $\mathrm{BB}(i)$ was obtained, then $\mathrm{depth}_\mathrm{bb}(\mathrm{BB}(i)) = \min\{|p_i| - K(\mathrm{BB}(i)) + i\}$, where $U(P_i) = K(\mathrm{BB}(i))$. It follows that $K(\mathrm{BB}(i)) = |p_i|$ and $\mathrm{depth}_\mathrm{bb}(\mathrm{BB}(i)) = i$; otherwise p_i would not be the minimal program.

Computing the system described requires a solution for the halting problem, and the system itself might also seem unnatural at first glance. However, we can think of the biosphere as a huge parallel computer that is constantly approximating solutions to the adaptation problem by means of survivability, and just as Ω has been approximated [12], we claim that *just as we generally cannot know whether a Turing machine will halt until it does, we may not know if an organism will keep adapting and survive in the future, but we can know when it failed to do so (extinction).*

One may consider biological evolution to be a very rough, but fundamental, analogue to Chaitin's system in the pursuit of the busy beaver values (longest-running TMs), equivalent to the search for the longest-surviving organisms. Within this framework, the periods of increase in complexity are a natural consequence of approximating BB. In this view, our assertion that adaptation is a generalization of the halting problem has a natural interpretation.

4 On Logical Depth

Although we conjecture that Theorem 3.9 must also hold for logical depth as defined by Bennett [8], extending the results to this measure is still a work in progress. Encompassing logical depth will require a deeper understanding of the internal structure of the relationship between system and computing time, beyond the time complexity stability (Theorem 2.3), and might be related to open fundamental problems in computer science and mathematics. For instance, finding a *low* upper bound to the growth of logical depth of all computable series of natural numbers would suggest a negative answer to the question of the existence of an efficient way of generating deep strings, which Bennett relates to the $P \neq \mathrm{PSPACE}$ problem.

Given that we intend to expand upon these questions in the future, it is important to address the fact that the diagonal algorithm that Bennett proposes for generating deep strings seemingly represents a contradiction to our conjecture: The *logical depth* of a natural x at the level of significance c is defined as

$$\text{depth}_c(x) = \min\{T(p) : |p| - K(x) < c \text{ and } U(p) = x\}.$$

The algorithm $\chi(n, T)$ produces strings of length n with depth T for a significance level $n - K(T) - O(\log n)$, where $K(T)$ must be smaller than n, and n must not be as large as (or larger than) T to avoid shallow strings. One possible difficulty with this algorithm is that the significance level is not computable, and we can expect it to vary greatly with respect to $K(T)$: For large T with small $K(T)$ (such as T^{T^T}), the significance level is nearly n, which suggests that, for a *steady* significance level with respect to times T with large $K(T)$, the growth in complexity might not be stable. This problem, along with an algorithm that consistently enumerates pairs of n and T such that $K(T) < n \ll T$ for growing T's, will be explored in future work, and its solution would require a formal definition of *adequate* complexity measures. The fact that χ presents a challenge to Conjecture 3.11 would suggest an important difference from the three complexity measures used in this article.

5   Conclusions

We have presented a formal and general mathematical model for adaptation within the framework of computable dynamical systems. This model exhibits universal properties for all computable dynamical systems, of which Turing machines are a subset. Among other results, we have given formal definitions of open-ended evolution (OEE) and strong open-ended evolution and supported the latter on the basis that it allows us to differentiate between trivial and nontrivial systems.

We have also shown that decidability imposes universal limits on the growth of complexity in computable systems, as measured by sophistication, coarse sophistication, and busy beaver logical depth. We show that as time dominates the descriptive algorithmic complexity of the states, the complexity of the evolution of a system tightly follows that of natural numbers, implying the existence of nontrivial states but the nonexistence of an algorithm for finding these states or any subsequence of them, which makes the computations for harnessing or identifying them undecidable.

Furthermore, as a direct implication of Corollary 3.2 and Theorem 3.9, the undecidability of adapted states and the unpredictability of the behavior of the system at each state is a requirement for a system to exhibit strong open-ended evolution with respect to the complexity measures known as sophistication, coarse sophistication, and busy beaver logical depth, providing rigorous proof that undecidability and irreducibility of future behavior is a requirement for the growth of complexity in the class of computable dynamical systems. We conjecture that these results can be extended to any adequate complexity measure that assigns low complexity to random objects.

Finally, we have provided an example of a (noncomputable) evolutionary system that exhibits strong OEE, and supplied arguments for its adequacy as a model of evolution, which we claim supports our characterization of strong OEE. Also, we assert that adaptation can be seen as analogous to the problem of finding the values of the busy beaver function, producing complexity as a byproduct.

One possible reading for the undecidability results exposed in this article is that the chase for strong OEE is a scientific dead end, as the sequence of adapted states are not even semicomputable. However, the beauty of an algorithmic theory of evolution, as briefly introduced in Section 3.1, already exposes a strong OEE system that opens some avenues for further research in artificial and biological evolution within an algorithmic probability framework. Some results in this direction can already be found in [23].

References

1. Adami, C. (2002). What is complexity? *BioEssays*, *24*(12), 1085–1094.

2. Adami, C., & Brown, C. T. (1994). Evolutionary learning in the 2D artificial life system 'Avida.' In R. A. Brooks & P. Maes (Eds.), *Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems* (*Vol. 1194*, pp. 377–381). Cambridge, MA: MIT Press.

3. Adams, A., Zenil, H., Davies, P. C. W., & Walker, S. I. (2017). Formal definitions of unbounded evolution and innovation reveal universal mechanisms for open-ended evolution in dynamical systems. *Scientific Reports*, *7*(1), [00810]. DOI: 10.1038/s41598-017-00810-8.

4. Antunes, L., & Fortnow, L. (2003). Sophistication revisited. *Theory of Computing Systems*, *45*(1), 150–161.

5. Auerbach, J. E., & Bongard, J. C. (2014). Environmental influence on the evolution of morphological complexity in machines. *PLoS Computational Biology*, *10*(1), e1003399.

6. Bedau, M. (1998). Four puzzles about life. *Artificial Life*, *4*(2), 125–140.

7. Bedau, M., McCaskill, J. S., Packard, N. H., Rasmussen, S., Adami, C., Green, D. G., Ikegami, T., Kaneko, K., & Ray, T. S. (2000). Open problems in artificial life. *Artificial Life*, *6*(4), 363–376.

8. Bennett, C. H. (1988). Logical depth and physical complexity. In R. Herken (Ed.), *The universal Turing machine: A half-century survey* (pp. 227–257). Oxford: Oxford University Press.

9. Blondel, V. D., Bournez, O., Koiran, P., & Tsitsiklis, J. N. (2000). The stability of saturated linear dynamical systems is undecidable. In H. R. S. Tison (Ed.), *Annual Symposium on Theoretical Aspects of Computer Science* (pp. 479–490). Berlin, Heidelberg: Springer.

10. Bournez, O., Graça, D. S., Pouly, A., & Zhong, N. (2013). Computability and computational complexity of the evolution of nonlinear dynamical systems. In P. Bonizzoni, V. Brattka, & B. Löwe (Eds.), *The nature of computation. Logic, algorithms, applications* (pp. 12–21). Berlin, Heidelberg: Springer.

11. Calude, C. S., & Jugensen, H. (2005). Is complexity a source of incompleteness? *Advances in Applied Mathematics*, *35*(1), 1–15.

12. Calude, C. S., Dinneen, M. J., & Shu, C. K. (2002). Computing a glimpse of randomness. *Experimental Mathematics*, *11*(3), 361–370.

13. Calude, C. S., & Stay, M. (2008). Most programs stop quickly or never halt. *Advances in Applied Mathematics*, *40*(3), 295–308.

14. Chaitin, G. J. (2012). Life as evolving software. In H. Zenil (Ed.), *A computable universe: Understanding and exploring nature as computation* (pp. 297–322). Singapore: World Scientific.

15. Chaitin, G. J. (2013). *Proving Darwin: Making biology mathematical*. New York: Vintage.

16. Chaitin, G. J. (2009). Evolution of mutating software. *Bulletin of the European Association for Theoretical Computer Science*, *97*, 157–164.

17. Cooper, S. B. (2009). Emergence as a computability-theoretic phenomenon. *Applied Mathematics and Computation*, *215*(4), 1351–1360.

18. Daley, R. (1982). Busy beaver sets: Characterizations and applications. *Information and Computation* (formerly *Information and Control*), *52*, 52–67.

19. Delvenne, J. C., Kurka, P., & Blondel, V. D. (2006). Decidability and universality in symbolic dynamical systems. *Fundamenta Informaticae*, *74*(4), 463–490.

20. Fredkin, E., & Toffoli, T. (1982). Conservative logic. *International Journal of Theoretical Physics*, *21*, 219–253.

21. Bennett, C. H., & Gardner, M. (1979). Mathematical games: The random number omega bids fair to hold the mysteries of the universe. *Scientific American*, *241*(5), 20–34.

22. Hernández-Orozco, S., Hernández-Quiroz, F., & Zenil, H. (2016). Undecidability and irreducibility conditions for open-ended evolution and emergence. arXiv:1606.01810.

23. Hernández-Orozco, S., Narsis, K., & Zenil, H. (2017). Algorithmically probable mutations reproduce aspects of evolution such as convergence rate, genetic memory, modularity, diversity explosions, and mass extinction. arXiv:1709.00268.

24. Kirchherr, W., Li, M., & Vitányi, P. (1997). The miraculous universal distribution. *The Mathematical Intelligencer*, *19*(4), 7–15.

25. Koppel, M. (1988). Structure. In R. Herken (Ed.), *The universal Turing machine: A half-century survey* (pp. 35–452). Oxford: Oxford University Press.

26. Lehman, J., & Stanley, K. O. (2008). Exploiting open-endedness to solve problems through the search for novelty. In S. Bullock, J. Noble, R. Watson, & M. A. Bedau (Eds.), *Artificial Life XI, Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Live Systems* (pp. 329–336). Cambridge, MA: MIT Press.

27. Lindgren, K. (1992). Evolutionary phenomena in simple dynamics. In C. G. Langton, C. Taylor, J. D. Farmer, & S. Rasmussen (Eds.), *Artificial Life II* (pp. 295–312). Redwood City, CA: Addison-Wesley.

28. Margolus, N. (1984). Physics-like models of computation. *Physica*, *10*, 81–95.

29. Moore, C. (1991). Generalized shifts: Unpredictability and undecidability in dynamical systems. *Nonlinearity*, *4*(2), 199–230.

30. Ruiz-Mirazo, K., Peretó, J., & Moreno, A. (2002). A universal definition of life: Autonomy and open-ended evolution. *Origins of Life and Evolution of the Biosphere*, *34*(3), 323–346.

31. Soros, L. B., & Stanley, K. O. (2014). Identifying necessary conditions for open-ended evolution through the artificial life world of chromaria. In H. Sayama, J. Rieffel, S. Risi, R. Doursat, & H. Lipson (Eds.), *Artificial Life 14* (pp. 793–800). Cambridge, MA: MIT Press.

32. Standish, R. K. (2003). Open-ended artificial evolution. *International Journal of Computational Intelligence and Applications*, *3*(2), 167–175.

33. Taylor, T. (2015). Requirements for open-ended evolution in natural and artificial systems. arXiv:1507.07403.

34. Turing, A. M. (1936). On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, *42*, 230–265.

35. Zenil, H., Gershenson, C., Marshall, J. A. R., & Rosenblueth, D. A. (2012). Life as thermodynamic evidence of algorithmic structure in natural environments. *Entropy*, *14*(11), 2173–2191.