

# Cardinality Leap for Open-Ended Evolution: Theoretical Consideration and Demonstration by “Hash Chemistry”

Hiroki Sayama<sup>1,2</sup>

<sup>1</sup> Center for Collective Dynamics of Complex Systems  
Binghamton University, State University of New York  
Binghamton, NY 13902-6000, USA

<sup>2</sup> Waseda University, Shinjuku, Tokyo, Japan  
sayama@binghamton.edu

## Abstract

Open-ended evolution requires unbounded possibilities that evolving entities can explore. The cardinality of those possibilities thus has a significant implication for the open-endedness of evolution. We propose that facilitating formation of higher-order entities is a generalizable, effective way to cause a “cardinality leap” in the set of possibilities that promotes open-endedness. We demonstrate this idea with a simple, proof-of-concept toy model called “Hash Chemistry” that uses a hash function as a fitness evaluator of evolving entities of any size/order. Simulation results showed that the number of individual entities involved in a single replication event gradually increased over time, indicating evolutionary appearance of higher-order entities. It was also observed that the cumulative number of unique replicating entities that appeared in evolution increased almost linearly along time without a bound, presenting a concrete example of open-endedness achieved by the cardinality leap.

## Introduction

Open-ended evolution (OEE) (Taylor et al., 2016) requires unbounded possibilities that evolving entities can explore. Such an infinite possibility space can be conceptualized mathematically as an infinite set of all possible types, on which the *landscape* of evolution is constructed. One can picture that evolving entities collectively search this landscape, over an indefinitely period of time, for locations (possibilities) where they can have a greater chance of continuous existence. It remains unclear, however, how one can effectively achieve such an infinite possibility space in artificial life (ALife) models, and how its cardinality affects evolutionary dynamics.

In this short paper, we revisit an important fact that the cardinality of possibilities has a significant implication for the open-endedness of evolution, and propose that facilitating formation of higher-order entities is a generalizable, effective way to cause a “cardinality leap” in the set of possibilities. We first provide a theoretical consideration based on mathematical concepts of cardinalities, and then demonstrate the idea using a simple, proof-of-concept toy model that we call “Hash Chemistry.”

## Theoretical Consideration

### Mathematical assumptions

Let  $S$  be the set of all possibilities of individual entities in an evolutionary system, such as all possible genotypes or evolving computer codes.  $S$  can be either finite or infinite. For evolution to be open-ended,  $S$  should be infinitely large, but in all practical implementations of living systems (including biological and artificial ones)  $S$  is technically finite because evolving entities cannot be bigger than the environment. In addition, if the entities are symbolically represented (as is the case in real biological evolution and in most of ALife models),  $S$  is made of discrete entities, and therefore it is countable.

Here we assume a very simplistic view of evolution that there is(are) an *effectively optimal* possibility(-ies) somewhere in  $S$ , whose “fitness”, i.e., chance of successful survival, cannot be surpassed virtually by any other entity. We do not call this just *optimal* because, theoretically, one can construct a landscape on which no optimal type exists, e.g., when the fitness is given by  $1 - 1/L$  where  $L$  is the length of the entity’s description. But even in such cases, the fitness resolution limit coming from limitations in the laws of the world (e.g., quantum limit) makes two types indistinguishable if their  $L$ s are very large. Those indistinguishable types are both *effectively optimal*.

Evolution can be visualized as a collective search process in  $S$  to find such an effectively optimal entity. Once the evolution finds one, that entity will dominate the system, and the open-endedness of evolution will no longer be exhibited because there will be no more adaptive types discovered by the evolutionary process. The question of our interest, then, is if an evolutionary system is destined to reach this final state, and if so, if there is any workaround available to help the system avoid reaching this destination so that it can remain open-ended.

### Hurdle to open-endedness

Assuming that  $S$  is countable, all the possibilities can be mapped onto a set of natural numbers (1, 2, 3, ...) in some arbitrary order. The effectively optimal entity is also located

somewhere in this set of numbers. Let  $n_o$  be the number that corresponds to that entity. If  $S$  is finite, it is obvious that there is no way evolution could produce an indefinitely long sequence of search history continuously producing novel types without hitting  $n_o$ , and therefore, the evolution cannot be open-ended. Even if  $S$  is infinite, the countability assumption of  $S$  implies that every single entity in  $S$  will be visited eventually at a finitely distant point in time in the future if the entities of  $S$  are enumerated. Since open-ended evolution is essentially a parallel process of enumerating novel possibilities over time, open-ended evolution will eventually hit  $n_o$  at some point in the future, which ironically makes the process not open-ended. These arguments indicate that evolution cannot be truly open-ended if  $S$  is finite or even countably infinite.

There are several different ways to respond to the somewhat daunting conclusion derived above. One is to accept the countability of  $S$  but argue that the size of  $S$  is very large or that the time until enumeration hits  $n_o$  is very long, so that evolution within such bounds can still look very rich and open-ended. This is a perfectly logical, reasonable view, and it is certainly possible that the actual evolution of life on Earth is simply in the middle of a finitely long evolutionary path.

Another way of response is to assume an uncountably infinite set for  $S$ . This approach is equivalent to adopting continuous-valued representations for descriptions of evolving entities. Since enumeration of such continuous values is not possible, true open-ended evolution is logically possible within this setting. This view is, however, not consistent with the mechanisms of biological evolution as we know it where entities are encoded in discrete symbols. It also has a risk to fall into a naive conclusion that any chaotic dynamical systems are open-ended evolutionary systems. While it may be true mathematically, such a simplistic conclusion may not be helpful in making advance in ALife research on OEE.

### Higher-order entities and cardinality leap

Here we take yet another approach to the above issue, which, we believe, is generalizable and more useful for research on OEE. Specifically, we consider a way to expand the cardinality of the possibility set by facilitating formation of *higher-order entities*, i.e., combinations/coalitions of multiple individual entities in  $S$ .

Mathematically, a higher-order entity can be defined as a multiset (a set in which multiple copies of an identical entry are allowed) of entities of  $S$ . For example, with  $S$  being a set of chemical elements, molecules like  $O_2$  and  $H_2O$  are higher-order entities, represented by multisets  $\{O, O\}$  and  $\{H, H, O\}$ , respectively. Other examples include a multiset of molecules contained within a micelle, a multiset of organelles contained within a eukaryotic cell, symbiosis of multiple organisms, etc. It can be argued that many, if not

all, of the major transitions in evolution (Maynard-Smith and Szathmary, 1995; Szathmary, 2015) can be described mathematically as formation of such higher-order entities (i.e., multisets of individual entities).

An important mathematical fact we want to point out is that the formation of higher-order entities naturally causes a “cardinality leap” in the possibility set. Let  $S^*$  be the possibility set of higher-order entities, i.e., the set of all multisets of individual entities in  $S$ . Each higher-order entity in  $S^*$  can be mapped onto a sequence of non-negative integers whose components represent numbers of individual entities of  $S$  that are contained in the multiset. For example, with  $S = \{H, He, Li, Be, B, C, N, O, F, Ne, \dots\}$ ,  $O_2$  and  $H_2O$  are mapped onto  $(0, 0, 0, 0, 0, 0, 0, 2, 0, 0, \dots)$  and  $(2, 0, 0, 0, 0, 0, 0, 1, 0, 0, \dots)$ , respectively. Given that the numbers of individual entities that appear in the multiset are theoretically unbounded, the size of  $S^*$  is infinite. If  $S$  is finite,  $S^*$  is the set of  $|S|$ -dimensional non-negative integer-valued vectors, and thus  $S^*$  is countably infinite. Or, if  $S$  is already countably infinite,  $S^*$  is the set of infinitely long sequences of non-negative integers, whose cardinality equals the cardinality of real numbers (i.e., uncountably infinite) by the diagonal argument (Simmons, 1993).

Note that in either case, the cardinality of possibility sets makes a fundamental leap (from finite to countably infinite, or from countably infinite to uncountably infinite). Such a cardinality leap would greatly promote the open-endedness of evolutionary processes. In particular, if the original possibility set  $S$  is already countably infinite, the formation of higher-order entities can achieve an uncountably infinite possibility space, creating a theoretical possibility of true open-ended evolution in it, even if the individual entities in  $S$  are represented in discrete symbols.

### Proof of Concept: Hash Chemistry

The theoretical consideration given in the previous section may give us some hope, but it does not provide concrete guidelines about how one can construct an artificial evolutionary system that has the potential to form higher-order entities and thereby exhibit OEE. In this section, we present a simple toy model to demonstrate how it could be done, at least in a very primitive form.

### General architecture

In creating artificial evolutionary systems that can facilitate formation of higher-order entities, one technical challenge is how to design a mechanistic, universal means that can evaluate the level of success of entities of *any arbitrary size*. The real laws of physics/chemistry/biology apparently have no issue on this, because they are fundamentally bottom-up and fully distributed. However, most of the existing evolutionary models assume a typical genotype-phenotype mapping, evaluating the fitness of each individual type separately. Some evolutionary models adopted more elabo-

rated methods such as context/environment dependence or multilevel selection, but the fitness evaluation mechanisms in those models are not flexible enough to be applied to  $S^*$  in the context of this study. Constructing a physically/chemically/biologically plausible mechanism for universal fitness evaluation of any higher-order entities would probably require a combinatorially large amount of design effort.

Because solving the above challenge is not part of our objectives, we circumvent the problem by throwing it at *Deus ex Machina* available in most computational environments, a.k.a. the *hash function*. A hash function takes any hashable data and returns a hash value that is deterministically assigned to the given data, which is perfect as a quick-and-dirty substitute of the universal fitness evaluator for lazy modelers (like the author of this paper).

With the power of this hash function, one can build a simple evolutionary model, which we call “Hash Chemistry,” in the following general architecture that follows a typical Artificial Chemistry framework (Dittrich et al., 2001; Banzhaf and Yamamoto, 2015):

1. Define a set of possibilities of individual entities ( $S$ ).
2. Define a spatial domain in which entities reside and interact. This domain can be a continuous Euclidean space, a discrete lattice or network, or any other space, as long as one can represent proximity of entities on it.
3. Initialize the system by placing some individual entities in the spatial domain.
4. Using any method of choice, extract a multiset of individual entities that are spatially close to each other (this could be just a set of one entity as well). This multiset is now a higher-order entity whose behavior is to be determined below.
5. Apply the hash function to the multiset extracted above (typically represented as a sorted list of the types of individual entities in it). The returned hash value should be normalized somehow to an interpretable fitness value  $f$ .
6. Based on  $f$ , do one of the following: (a) Add a copy of the multiset to the space (replication). (b) Remove the extracted multiset from the space (death). (c) Do neither.
7. Repeat 4–6.

In addition to the above steps, one may also include other factors such as spatial movement of entities, stochastic changes of entity types (mutation), and the carrying capacity of the space, to make the dynamics more plausible and/or practical.

## Specific implementation

We implemented a specific version of Hash Chemistry in Wolfram Research Mathematica. The following model settings were adopted:

- The possibility set of individual entities (types) are natural numbers ranging from 1 to 1,000, i.e.,  $S = \{1, 2, \dots, 1000\}$ .
- The space is a two-dimensional continuous unit square with cut-off boundaries, i.e., entities are not allowed to move out of the boundaries.
- The initial configuration is made of 10 individual entities of randomly chosen types, randomly distributed within the space.
- Each simulation is run for 2,000 iterations (time steps).

In each iteration, the following steps are taken to update the system’s state:

1. Move each of the individual entities randomly by adding to its spatial position a small randomly oriented vector whose length is sampled from a half-normal distribution with  $\sigma = 0.15$ .
2. For each of the positions of individual entities, do the following:
  - (a) Create a set  $N$  of individual entities that are of close distance (0.05 or less) from the focal position.
  - (b) Choose a random subset  $s$  of  $N$  by randomly selecting  $k$  entries from  $N$ , where  $k$  is a random integer in  $\{1, 2, \dots, |N|\}$ . This represents a multiset of entity types, i.e., a higher-order entity whose success in replication and/or survival is to be determined below.
  - (c) With probability  $1/|s|$  (this is to standardize the average probability of updating per entity per unit of time), do the following:
    - Create a sorted list of types of the individual entities in  $s$ .
    - Calculate the fitness  $f$  of  $s$  by applying Mathematica’s Hash function (Wolfram Research, 2018) to the above list. The output is mapped to a  $[0, 1)$  fitness range by computing  $(h \bmod m)/m$ , where  $h$  is the output of Hash ( $m = 100,000$  for the results shown here).
    - With probability  $1 - f$ , delete all individual entities in  $s$  from the space. (*death*)
    - With probability  $f(1 - |N|/d_{\max})$ , where  $d_{\max}$  is the maximum density of entities ( $d_{\max} = 100$  for the results shown here), add copies of all individual entities in  $s$  to the space. (*growth*)
3. For each of the individual entities, change its type to a type randomly sampled from  $S$  with probability 0.01.
4. Randomize the order of individual entities.

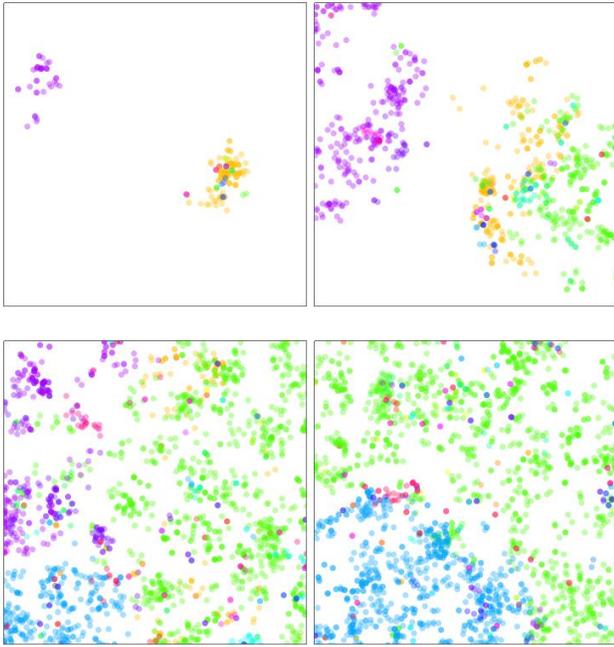


Figure 1: Sample simulation run of Hash Chemistry. Each frame shows a snapshot of the system at a certain time point (from top left to bottom right:  $t = 30, 100, 300, 1000$ ). Each individual entity is represented as a dot in the space, with a color showing its entity type. A movie of this simulation run is available on YouTube at <https://youtu.be/fVwUJ7pdPWY>.

## Results

This Hash Chemistry model was built only as a proof of concept, and a systematic evaluation of its dynamics is outside the scope of this short paper. Here we anecdotally present an illustrative example of a typical simulation run. We have run several independent simulations so far, and the observations indicate that the simulation results are qualitatively similar and fairly robust.

Figure 1 shows several snapshots of the system's state and its spatio-temporal development over time in one simulation run (a movie of this simulation run is available on YouTube at <https://youtu.be/fVwUJ7pdPWY>). Figure 2 presents time series of (i) maximum/average fitness values of individual entities that were successfully replicated, and (ii) the number of replicated individual entities. The former quickly increased and converged at steady state values, while the latter gradually increased over time. These are typical simulation results showing the evidence of adaptation, nothing particularly different from what have been reported with other evolutionary models in the literature.

However, Figure 3 tells a rather different story. In these plots, it is shown that the number of individual entities that were involved in a single replication event gradually increased over time. This means that the individual entities

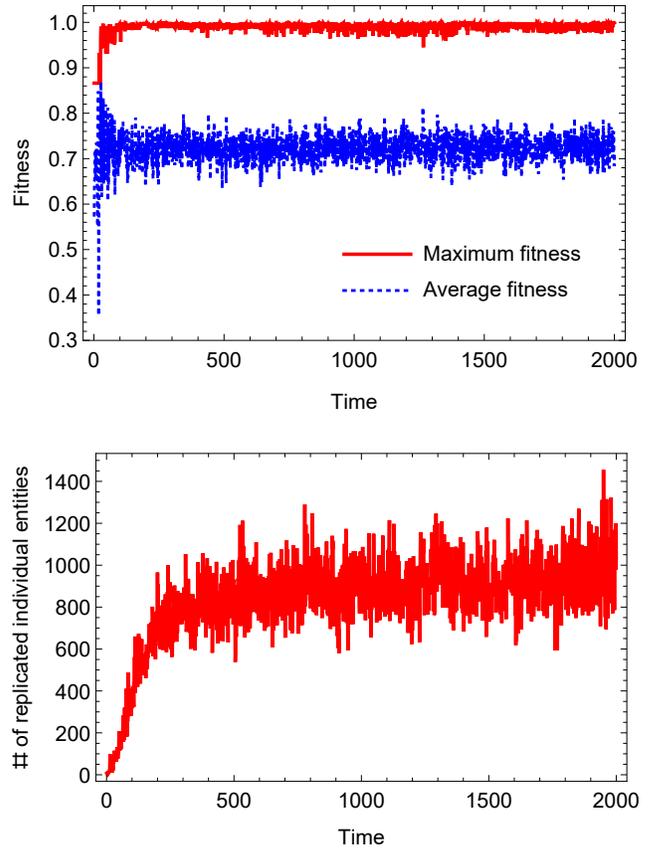


Figure 2: Evidence of adaptation going on in Hash Chemistry. Top: Maximum and average fitness values of replicated individual entities. Maximum fitness quickly increases and stays nearly at 1.0. Bottom: Number of individual entities replicated in each time step.

gradually became more and more replicated together with others, behaving as higher-order entities. This can be interpreted in that this evolutionary system continuously discovered slightly higher fitness values for increasingly higher-order entities in  $S^*$  on the nontrivial fitness landscape defined by the hash function. This kind of spontaneous increase in the order of evolving entities is quite unique to this Hash Chemistry model.

The most striking finding is shown in Figure 4, where the cumulative numbers of unique entity types that have ever appeared in the course of the simulation run are plotted. If only the types of individual entities are counted (Fig. 4 top), the number grows following a typical S-shaped growth curve, quickly exhausting all the possible types (1,000) in the early stage of evolution. This corresponds to the inevitable consequence of evolutionary enumeration of possibilities. In this view, open-ended evolution is not possible. However, if the types of higher-order entities are counted (Fig. 4 bottom), the number of possibilities continues increasing almost linearly along time, far above the number of possibilities of

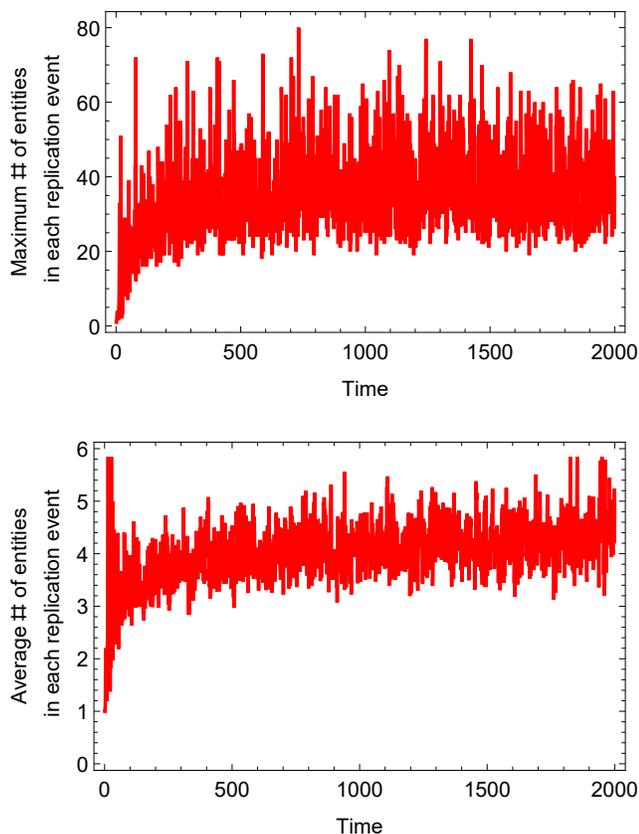


Figure 3: Maximum (top) and average (bottom) numbers of individual entities involved in each replication event.

individual entities. This is made possible by the cardinality leap caused by formation of higher-order entities. Namely, the cardinality leap results in an unbounded number of combinations of individual entities, and such higher-order entities can and do appear as the system explores and discovers them as more fit entities in the course of evolution.

## Conclusions

In this short paper, we emphasized the significance of formation of higher-order entities as a generalizable mechanism to induce a cardinality leap in the possibility set, naturally facilitating open-ended evolution. This idea was illustrated with a concrete computational model, Hash Chemistry, which adopted a general-purpose hash function as a means to evaluate fitness of evolving entities of any size or order. Numerical simulations successfully demonstrated evolutionary appearance of higher-order entities, and moreover, unbounded increase in the cumulative number of novel types produced in evolution, even if the possibilities of individual entity types were finite.

These results constitute a concrete, operationalized example of cardinality leaps through formation of higher-order entities, suggesting that it is indeed possible to achieve open-

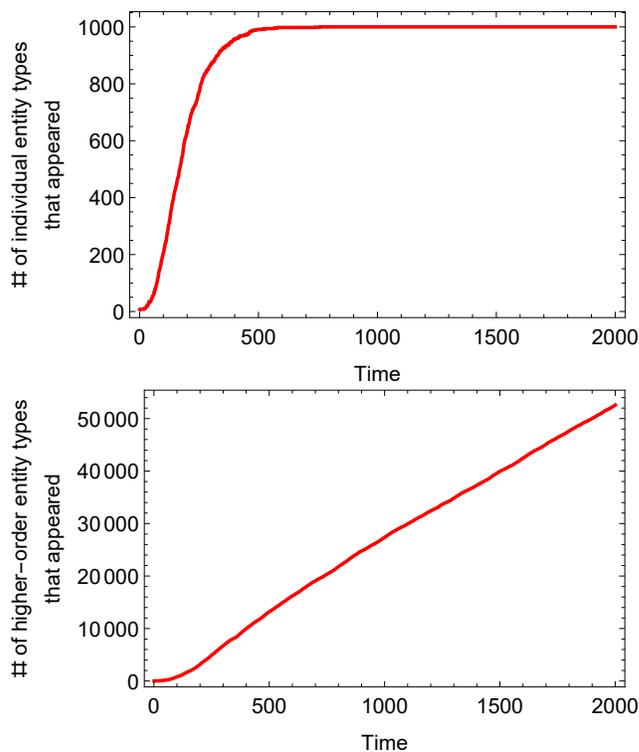


Figure 4: Cumulative numbers of unique entity types that have ever appeared in the course of simulation. Top: Number of individual entity types, which saturates at the maximum value (1,000). Bottom: Number of higher-order entity types, which continues to grow almost linearly along time.

ended evolution in a relatively simple ALife model framework. The present work remains at an anecdotal proof-of-concept level, and therefore a more systematic, quantitative evaluation of the model and its dynamics will be among our future studies. We also plan to explore how the idea of cardinality leaps could be introduced to other ALife models, including our recent evolutionary swarm models (Sayama, 2011, 2018).

## Acknowledgments

The author thanks Howard H. Pattee for having insightful discussions that provided many inspirations for this work.

## References

- Banzhaf, W. and Yamamoto, L. (2015). *Artificial Chemistries*. MIT Press.
- Dittrich, P., Ziegler, J., and Banzhaf, W. (2001). Artificial chemistries—a review. *Artificial Life*, 7(3):225–275.
- Maynard-Smith, J. and Szathmary, E. (1995). *The Major Transitions in Evolution*. Oxford University Press.
- Sayama, H. (2011). Seeking open-ended evolution in Swarm Chemistry. In *Proceedings of the 2011 IEEE Symposium on Artificial Life (IEEE ALIFE 2011)*, pages 186–193. IEEE.

- Sayama, H. (2018). Seeking open-ended evolution in Swarm Chemistry II: Analyzing long-term dynamics via automated object harvesting. In *Proceedings of the 2018 Artificial Life Conference*. MIT Press. In press.
- Simmons, K. (1993). *Universality and the Liar: An Essay on Truth and the Diagonal Argument*. Cambridge University Press.
- Szathmary, E. (2015). Toward major evolutionary transitions theory 2.0. *Proceedings of the National Academy of Sciences*, 112(33):10104–10111.
- Taylor, T., Bedau, M., Channon, A., Ackley, D., Banzhaf, W., Beslon, G., Dolson, E., Froese, T., Hickinbotham, S., Ikegami, T., et al. (2016). Open-ended evolution: perspectives from the oee workshop in york. *Artificial Life*, 22(3):408–423.
- Wolfram Research (2018). Wolfram Language & System Documentation Center – Hash. <http://reference.wolfram.com/language/ref/Hash.html>. Accessed: 2018-06-18.