Novel Properties Generated by Interacting Computational Systems

Fabio Boschetti

CSIRO Marine and Atmospheric Research, Australia
and
School of Earth and Environment
The University of Western Australia

John Symons

University of Texas at El Paso
El Paso, Texas 79968, USA
jsymons@utep.edu

After giving definitions for novelty and causality for computational systems, we describe a simple computational system that is able to produce genuine novelty. We explain how novelty can arise in systems of simple interacting components and describe what it would mean for such emergent properties to have causal powers.

1. Introduction

In this work we address two questions: first, what is the smallest number of components a computational system needs in order to display genuine novelty? And second, can the novel features of such systems also exhibit novel causal powers?

The possibility of generating novelty via computational processes has been an ongoing topic of investigation and debate in artificial intelligence research for several decades. More recently, topics such as self-organization and emergence have been discussed in computational terms within complex systems science. Researchers in complexity have worked to understand the possibility of genuine novelty in computational systems in order to understand the significance of computational models of putatively emergent properties. Putatively emergent properties such as the flocking behavior of birds (Reynolds 1988), the adaptive features of the immune systems (Hofmeyer et al. 2000), and the characteristic patterns of traffic flow (Schreckenberg 1995) have been given computational models.

Since “novelty” and “causation” carry different meanings in different scientific contexts, it is important to specify which definitions we use in this work. In this context, a system can be thought to generate genuine novelty given the appearance of processes or behaviors not explicitly coded in the generating algorithm [1]. We roughly follow the spirit of the definition of emergence in computational models given by Stephanie Forrest and others. However, we do not wish to rest a definition of novelty on the explicit intentions of the authors of the code. Instead, given an automaton, we define novel structures as those which cannot arise solely from the code that determines the properties of the automaton. By "causality," we mean the possibility of intervention or control in terms of idealized agency [2–4]. Our view of causation is explained in more detail in Section 3. Both definitions provide constraints that we believe set an appropriately high bar for discussions of novel properties in computational systems. In our view, if more relaxed accounts of causation and novelty were adopted, many of the more ambitious goals implicit in the scientific study of complexity would become trivial.

One underlying problem in both artificial intelligence and in complex systems science involves determining whether novel features of systems are, in fact, endowed with causal power independent of the causal powers of their components. Interest in novel causal agency is evident, for example, in the desire among computer scientists to develop agents that exhibit genuinely autonomous interaction with a changing environment. In a somewhat different context, researchers in complexity science hope to provide models allowing us to study the causal characteristics of emergent properties in complex systems. Emergent properties of complex systems are interesting precisely insofar as they are not merely epiphenomenal, but instead result in some genuinely new agency in the system under consideration.

In a number of papers, we (along with co-authors) have stressed that within computational systems, genuinely novel causal powers, as described, cannot occur. Our argument is based on the acknowledgment that traditional (non-interactive) computational systems are closed systems, whose dynamics are fully predetermined by the initial conditions and a fixed set of rules; we refer the reader to our previous work for more details [5, 6]. According to this view, emergent patterns arising from computational systems may appear novel to an observer with incomplete knowledge of the system or to an agent who does not possess logical omniscience. By contrast, patterns resulting from non-interacting computational processes would be logically deducible, and thus not genuinely novel, given full knowledge of the system's code (and logical omniscience). Thus, while weakly emergent features of computational models of the kind discussed by Mark Bedau and others [7, 8] are objective features of those systems, they would not count as genuinely new from our viewpoint. (See [9] for further discussion.) Following a similar argument, in our view, patterns arising from non-interacting computational systems do not possess unique causal powers independent of their components, since whatever causal agency the system exhibits is restricted to the rules written in the algorithm [10, 11]. Emergent patterns can convey information to an agent who is not logically omniscient and who does not have complete knowledge of the rules of the system, but they carry no novel information that has relevant consequences within the processes in operation within the system [10].

The idea that the generation of genuine novelty and independent causal processes require a system to be open (and thus interaction with at least one external system) has been discussed within different contexts [12–15]. In the existing literature, notions of interaction and openness tend to be imprecise. As a step toward clarifying these concepts, we aim to develop a system of minimal size and complexity capable of generating novel patterns. Here, the patterns considered are strings of symbols. The system we propose requires three components: (*a*) two initial automata, (*b*) interaction, and (*c*) the ability of at least one automaton to process strings not predefined in its alphabet. For our purposes here, the latter would permit the capacity to engage in genuine interaction. Of course, there are a variety of ways that systems could interact and we call this general feature *interaction openness*. In accounting for the interactive character of a system, it is important to understand the nature of a system's interaction openness.

Because of the close relation between the generation of novelty and the occurrence of genuine causation [10, 11], we also aim to discuss whether this basic system allows for the appearance of patterns with novel causal powers. We suggest that even in such simple systems, the judgment of whether genuinely novel structures and genuine causation occur depends crucially on the problem of determining system boundaries.

The problem of how one individuates systems poses a profound challenge for metaphysics and is beyond the scope of the present paper. While we do not rule out the possibility that oncologists may provide a strategy for individuating systems in a non-arbitrary way, judgments with respect to causal power in the contexts we discuss here will obviously be relative to the determination of those boundaries. (See [16].)

2. The System

2.1 The Automata

We first define two automata as follows:

- **Machine A:** Alphabet=[0, 1]; Initial state=[0 0]; Transition rules=[00→1, 01→1, 10→0, 11→0]
- **Machine B:** Alphabet=[0, 1]; Initial state=[0 0]; Transition rules=[00→1, 01→0, 10→1, 11→0]

Each automaton generates periodic strings according to the defined transitions. In particular,

Machine A	0011001100110011001100 11…
Machine B	0010010010010010010010 01….

We can use the causal state splitting reconstruction (CSSR) algorithm [17] to reconstruct the ε-machine from the output strings and to estimate the statistical complexity and the entropy rate of the strings. Since both machines generate periodic output, their entropy rate is zero. The statistical complexity is 2 and 1.58 for Machines A and B, respectively, since Machine A has a period 4 and Machine B has a period 3 (see Appendix A for a brief description of the CSSR algorithm).

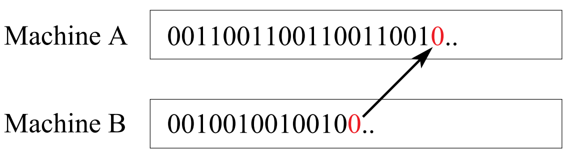
Now suppose that an external observer sees the combined output of the two machines. For example, the observer may not be able to discern that a system is made up of the two machines, but may see the output of a pair of symbols [[0 0], [0 0], [1 1], [1 0], …], which is obtained by combining the symbols emitted at the same time. The outcome is another periodic time series with period 12 and consequently of zero entropy rate and a statistical complexity of 3.58.

Note that so far the machines are not interacting; their output is simply combined and appears more complex to an external observer.

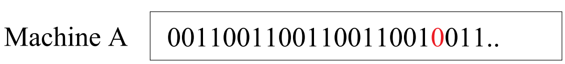
2.2 Interaction

Now we include in the system an interactive identity machine (IIM) as defined in [13]; this performs a "unit of interaction" by taking an input and emitting it unaltered as output.

The IIM enables Machines A and B to interact. In particular, the IIM takes a symbol from Machine B and copies it unaltered to a certain location of Machine A's tape.



After the interaction has occurred, Machine A proceeds in the computation by following its transition table:



As a result of the interaction, Machine A has performed a transition [01→0] that was not specified in the original transition rules, and consequently the output contains a word [010] that was not present in the tape before the interaction. To an external observer the state "01" now appears to have a non-determinist transition rule, since it has been observed transiting to either "1" or "0". As a result, both the statistical complexity and the entropy rate of the machine increase; the exact size of the increase depends on how often the interaction occurs and is not relevant to our discussion.

We now define Machine C.

- **Machine C:** Alphabet=[22, 23, 32, 33]; Initial state=[22]; Transition rules=[22→3, 23→2, 32→2, 33→2]

Machine C is equivalent to Machine B, with the symbols 2 and 3 replacing 0 and 1, respectively. As a result, Machine C has a statistical complexity equal to 1.58 and zero entropy rate.

Then, we apply an IIM to Machines A and C:

Machine A 001100110011001100113

Machine C 2232223222322232223...

Unlike before, now Machine A is not able to proceed since the word "13" is not in its alphabet and no transition rule is available to process it. As a result, Machine A halts.

Finally, we introduce Machine D by modifying Machine A.

- **Machine D:** Initial state=[0011]; Transition rules=[go back 4 steps along the tape; copy the next 3 symbols; go forward 4 steps along the tape and paste the 3 symbols]

Machine D 001100110011001100110011.....

Machine D generates the same output as Machine A, which results in a statistical complexity equal to 2 and zero entropy rate.

We now apply an IIM to Machines D and C.

Machine D 0011001100110011001103

Machine C 2232223222322232223...

Despite the word "03" not appearing in Machine D's alphabet, Machine D is able to process it by simply copying and pasting the values along the tape. This is because its transition rule is a function of the tape position, not of the machine's initial alphabet:

Machine D 001100110100110011031103110311031103 ...

In its processing, now Machine D also generates the word "31". Note that the words "03" and "31" are new not only to Machine D, but also to Machine C and consequently also to the larger system Machine D ⋃ Machine C.

An observer studying the output of Machine D would now see an entropy rate larger than zero, as for Machine A after it interacted with Machine B. This is due to the uncertainty generated by the sudden appearance of symbol 3 and the resulting unexpected transition. Also the observer will see eight states, with a statistical complexity close to 3 (the exact value depends on when the interaction happens and is not relevant to our discussion).

Finally, an observer external to the system Machine D $\bigcup$ Machine C, analyzing the combined output of the two machines [[0 2], [0 2], [1 3], [1 2], …], would see a time series with 24 states (12 before the interaction and 12 afterwards) and an entropy rate larger than zero. To this observer, this combined output appears the most complex among those described so far.

3. Causation

In this paper, we adopt an interventionist view of causation, according to which causation implies control in terms of an idealized agency that can produce an effect by altering the course of a process [3, 18]. As Menzies and Price put it [19]: "an event A is a cause of a distinct event B just in case bringing about the occurrence of A would be an effective means by which a free agent could bring about the occurrence of B."

Hausman [2] defines causal control as a *relation* between processes. This allows us to think of causation in terms of intervention, free from any anthropocentric interpretations. This is achieved by replacing the intuitive idea of human intervention with the abstract concept of an idealized agent able to carry out an intervention. To clarify, imagine the relation between (*a*) a human actor A, (*b*) a cause C the human actor can manipulate, and (*c*) the resulting effect E. In all situations in which the relation between processes P1, P2, and P3 is analogous to the relation between A, C, and E, we call P1 a *generalized agent*, we call *intervention* the action of P1 on P2 [2, 19], and we call *effect* the (potential) impact on P3. As a result, neither intervention nor agency implies human intervention, while they nonetheless satisfy the anthropocentric need for explanation.

Within the dynamical evolution of a system, it is important to stress the difference between the transitions carried out by a computational process on the one hand and the interventionist view of causation on the other. What differs is the relation between the states of the computational process (purely sequential and logically inevitable) and the idealized processes P1, P2, and P3 mentioned earlier (inherently parallel and dependent on the nature of the intervention).

This difference becomes important when we ask where the experimenter should intervene in order to change the behavior of a process. In a closed, computational process the dynamics can be inferred from the initial conditions and transition rules. Consequently, the only way to interfere with the system's behavior (to intervene in the system) is to modify either the input or the algorithm. Obviously, this would apply to the individual machines in isolation. When two machines interact, the experiment can also intervene by affecting the machine interaction itself. In Section 2.2, we have seen that the interaction (and thus the relation) between Machines C and D allows for the generation of novel structures. In the following section, we discuss whether the relation that generates these novel patterns also permits novel causal powers to arise.

4. System Boundaries

In the previous section, we have seen that novel strings can be generated by allowing an automaton (Machine D) to interact with an outside process (Machine C and IIM) and by defining its transition rule in terms of memory positions rather than a predefined alphabet. Nevertheless, if both the automaton and the outside process (Machine D, Machine C, and IIM) are coded in the same program, the entire system would be closed and, by applying the same reasoning discussed in Section 1, we would need to claim that genuine novelty cannot be generated by the system.

This problem reduces to defining the system boundary, an issue regularly encountered in complex systems science. Let us assume an observer can detect only the string output by Machine D. With the help of Figure 1, we can distinguish these four cases.

1. Machine D is the system and everything else constitutes its environment; this leads to a number of observations. First, the words "03" and "31" are novel to the system. Second, an external observer notices an abrupt change in system behavior after the interaction occurs. This occurrence is defined by the novel transition "10"→"3". What follows this transition appears to be a novel behavior characterized both by new words and new transitions. For example, by applying a machine reconstruction algorithm like the CSSR, what happens before the interaction would appear as a transient process and the word "01" as a transient state. Whether such a transient is a feature internal to Machine D dynamics or external to it is something we would not be able to discern without extra information. Elsewhere [16] we have argued that the possibility of transients simply cannot be excluded a priori. This provides a basic challenge to judgments concerning system boundaries in contexts where the boundaries cannot be stipulated by fiat. Obviously, an algorithm can be stipulated by fiat and the boundaries of the system behavior are a direct consequence of that stipulation. However, such stipulated boundaries have a highly idealized status requiring all contextual and environmental factors to be bracketed. By contrast, in natural science and in the modeling of complex systems, responsiveness to empirical considerations is directly relevant.

2. Machine D $\bigcup$ Machine C are the system and the IIM is external to it. The words "03" and "31" are still novel to the system, since they would not be generated without the IIM. Before the interaction occurs,

to an observer the behavior of the system appears similar to the one including only Machine D, since Machine C does not interfere with Machine D dynamics. All considerations, including the possible cause of the behavior change, would be as given earlier.

3. Machine D $\bigcup$ Machine C $\bigcup$ IIM constitutes the system. This brings about the crucial question of what triggers the IIM action. If the trigger rule is not coded in either machine, then IIM must respond to an external trigger; this external trigger then represents the idealized agent discussed in Section 3 and, within the interventionist view, constitutes the "cause" of the change in system behavior.

4. Machine D $\bigcup$ Machine C $\bigcup$ IIM constitutes the system and the trigger for the IIM action is coded within Machine C. This reduces the overall system to a closed system (a self-containing algorithm) in which, by definition, no causation can arise.

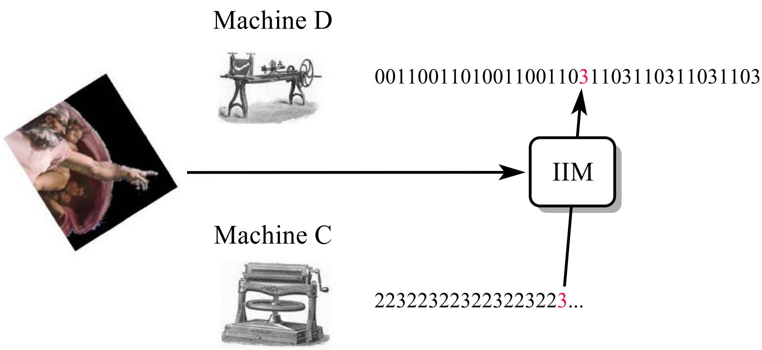


Figure 1. Relation between the two machines, the IIM, and the trigger to the interaction.

5. Interaction Openness

While the role of interaction and system boundaries is often discussed in the literature, to our knowledge less attention has been given to the importance of what we characterized earlier as interaction openness. In our previous discussion, an essential requirement for the generation of novel strings is Machine D's ability to process symbols in terms of tape position, not of transition rules; this feature allows Machine D to process a large set of incoming symbols, without any need for such symbols to be predefined. Loosely speaking, Machines A, B, and C are similar to traditional engineering systems, designed to interact with a predetermined class of processes, while Machine D is akin to a more "natural" system, able to interact with a larger class of processes thereby, allegedly, generating genuine novelty.

It is important to note that this flexibility comes at a cost to the observer. While it is in principle possible for an observer to determine the natural behavior of Machines A, B, and C, it may not be possible to do so for Machine D. It is so for two related reasons. First, after Machine D interacts with Machine C, a number of transient states become inaccessible, as will be discussed in detail in [16]. The possibility of transients ensures that any state or system could be the result of a process of emergence via interaction. Second, because a system's behavior is determined, at least partly, by the type of interaction it undergoes and the nature of the interacting process, the full range of possible system behaviors may itself not be closed, and consequently it may not be possible to fully determine. This is at odds not only with current practice of computational experimentation in complex systems science (in which behavior of the agents is usually fully predetermined) but also with the assumption that the behavior of the basic constituents of Nature is known and what needs to be explained are emergent properties only [11]. An extreme example of this type of system is provided by the IIM, whose definition is meaningful only within the context of an interaction.

In Section 4 we have seen that when the system boundaries are such as to enclose all processes, neither novelty nor genuine causation can arise. As mentioned, such conditions are highly idealized and make little sense apart from the realm of mathematical abstraction. It is useful to ask what happens at the opposite extreme, that is, when the system boundaries include only one element. Our discussion is based on three types of systems/processes: machines, IIM, and trigger of interaction. From an emergentist perspective, there is a marked difference between them: a machine, as employed in this paper, already involves a number of lower-level processes able, for example, to provide for interaction between a computational process and a tape, as well as for the storage and implementation of a number of instructions. Each of these processes is comparable to the IIM. At this level, according to our discussion, not only is interaction the determinant for behavior, but also behavior itself does not have a meaning outside interaction. At this level, everything is obviously emergent. However, in order to be so, the behavior of each element cannot be predetermined by the specification of the element itself; some sort of flexibility on what type of symbol each element can process (or respond to) must be available in order for complex processes to arise at all. This observation suggests that properties equivalent to interaction openness must be available to most basic elements for interesting novelty to emerge in Nature at all. This suggests the picture of a continuum ranging from a world in which no genuine novelty can arise (when system boundaries include all systems) to a world in which all processes are emergent and causal (when everything but one element is external to the system) and in which the transition from one extreme to the other is determined by the level of interaction openness. We plan to follow this direction in our future research.

6. Conclusions

Interaction is clearly a path to novelty. However, understanding the significance of this novelty involves attention to the way we individuate the components of the interaction. Discussions of novelty and emergence in complex systems science and computer science often take place in a way that obscures the central problem of defining the boundaries of the systems under consideration. We have shown, in a simple form, how the conceptual features of the problem of interaction can be characterized in a straightforward and non-question-begging way. Our initial characterization of interaction openness for systems is intended to focus future studies of interaction on specifiable features of systems that allow the possibility of interaction and emergence. What remains is for philosophers and scientists to shed more light on the problem of the individuation of systems. In the meantime, it is clear that our commitments with respect to system boundaries will determine our commitments with respect to the nature of interaction and the possibility of genuinely novel causal powers for emergent properties.

Acknowledgments

John Symons' work on this paper was supported by the John Templeton Foundation's individual research grants program.

Appendix

A. Statistical Complexity and the Causal State Splitting Reconstruction Algorithm

In the information theory literature, the concept of complexity is closely related to predictability and in particular to the amount of information required (difficulty) to achieve optimal prediction. This idea is captured by Kolmogorov's algorithmic complexity, according to which a fully random time series has maximum complexity.

In the complex systems science tradition, complexity is usually seen as something in between order and randomness. This alternative view is captured by Crutchfield and Young [20] as statistical complexity, defined as the amount of information needed to perform a useful prediction, which is understood as a prediction that captures the statistical properties of a process/data. The approach is summarized as follows.

1. Take the output of a process as a symbolized time series.

2. Use a machine learning algorithm to reconstruct the causal states of the process and their transitions.

3. Define the complexity of the process as the entropy of the causal states; this measures the uncertainty in predicting the next state of the system, given the information on its past behavior, and can be seen as a measure of the amount of memory in the system (in bits) that does a useful job in prediction.

A suitable machine learning algorithm to carry out step 2 is the causal state splitting reconstruction (CSSR) [17], which works as follows.

We take a discrete sequence of N measurements of the process we want to analyze, S_i, $i = 1 \ldots N$. At any time i, we can divide the series S into two half-series, $\overleftarrow{S}$ and $\overrightarrow{S}$, where $\overleftarrow{S} = ..S_{i-2}\, S_{i-2}\, S_i$ represents the "past" and $\overrightarrow{S} = ..S_{i+2}\, S_{i+2}\, S_{i+3} \ldots$ represents the "future." Following the same notation as in [17], we call $\overleftarrow{S}^L$ and $\overrightarrow{S}^L$ histories of length L symbols in the past and in the future, respectively. Also, we call s (and s^L) specific instances of histories belonging to S. Now, suppose we scan the series S, looking for occurrences of the history $\overleftarrow{s}$, and we store the symbol $\overrightarrow{S}^1$ seen as the future in each instance. We can calculate $P\left(\overrightarrow{S}^1 \mid \overleftarrow{s}\right)$, that is, the probabilities of occurrence of any of the k symbols in the alphabet A, given the history s, and we call the vector containing these probabilities the *morph* of $\overleftarrow{s}$. We can then define a *causal state* as the collection of all histories $\overleftarrow{s}$ with the same morph (i.e., histories that share the same probabilistic future). More formally, histories $\overleftarrow{s}_1$ and $\overleftarrow{s}_2$ belong to the same causal state if $P\left(\overrightarrow{S}^1 \mid \overleftarrow{s}_1\right) = P\left(\overrightarrow{S}^1 \mid \overleftarrow{s}_2\right)$.

With the given definition, the purpose of the CSSR algorithm is to reconstruct the set of the causal states of the process and the transition probabilities between the causal states. Following the nomenclature used in [17], the combination of causal states and their transition probabilities is called a ε-machine.

The CSSR algorithm can be divided into the following steps.

1. We start from the null hypothesis that the process is independent and identically distributed. In this case, each of the k symbols $a \in A$ is equally likely at each time step and only one causal state is necessary to model the process: the morph of the state is the k-length vector of components $1/k$.

2. We select a maximum history length *max_L* for our analysis. This is the length of the longest history with which we scan the series S. For histories of length = 1… *max_L*, we scan the series S, storing both the histories found and their futures. Given a history $\overleftarrow{s}$, its morph is trivially obtained by calculating $P\left(a \mid \overleftarrow{s}\right) = \nu\left(a, \overleftarrow{s}\right) / \nu(\overleftarrow{s})$ for each $a \in A$, where $\nu(\overleftarrow{s})$ is the number of occurrences of the history $\overleftarrow{s}$ and $\nu\left(a, \overleftarrow{s}\right)$ is the number of occurrences of the symbol a given the history $\overleftarrow{s}$.

3. We group histories with similar morphs into the same causal states. This involves three steps:

   (a) We need a measure for morph similarity. Real time series are characterized by both the presence of noise and by finite data extent. Consequently, we need to relax the requirement of exactly matching morphs $P\left(\vec{S}^{1} \mid \overleftarrow{s}_1\right) = P\left(\vec{S}^{1} \mid \overleftarrow{s}_2\right)$ to an approximation $P\left(\vec{S}^{1} \mid \overleftarrow{s}_1\right) \approx P\left(\vec{S}^{1} \mid \overleftarrow{s}_2\right)$. In particular, we accept $\left|P\left(\vec{S}^{1} \mid \overleftarrow{s}_1\right) - P\left(\vec{S}^{1} \mid \overleftarrow{s}_2\right)\right| < \varepsilon$, where ε is a user defined parameter.

   (b) We define the morph for a state as the average of the morph of all histories in that state.

   (c) In order to ensure the reconstruction of a minimum number of states, new states are created only when a history is found that cannot match any existent causal state. That is, for each history, we look for an existent state with similar morph and we create a new state only when we cannot find any.

   After these steps, we have a collection of states, grouping all histories found in the time series S according to the similarity between their morphs.

4. As a last step, we want to make sure that transitions between states, on a given symbol, are unique. That is, we want to make sure that, given any history in a state, and a next symbol $a \in A$, the next state is uniquely determined. Notice the difference between the occurrence of the next symbol, which is stochastic and measured by the morph, and the transition to the next state, given a next symbol, which we want to be deterministic. In order to do this, for each state, we store the next state transitions for each history, that is, we store the state a history attained after seeing a certain symbol. This is also represented by a vector of length k, containing, as elements, the next state on each symbol. If a state has two histories whose next state transition vectors are different, we split the state and create a new one.

Once the ε-machine is reconstructed, its entropy is the statistical complexity of the process as proposed by Crutchfield and Young [20].

References

[1] S. Forrest and J. H. Miller, "Emergent Behavior in Classifier Systems," *Physica D*, **42**(1–3), 1990 pp. 213–227. doi:10.1016/0167-2789(90)90075-Z.

[2] D. Hausman, *Causal Asymmetries (Cambridge Studies in Probability, Induction and Decision Theory)*, Cambridge: Cambridge University Press, 1998.

[3] H. Pattee, "Causation, Control, and the Evolution of Complexity," in *Downward Causation* (P. B. Andersen, C. Emmeche, N. O. Finnemann, and P. V. Christiansen, eds.), Århus: University of Århus Press, 1997 pp. 322–348.

[4] J. Pearl, "Statistics and Causal Inference: A Review," *Test Journal*, **12**(281), 2003 pp. 281–345.

[5] F. Boschetti and R. Gray, "Emergence and Computability," *Emergence: Complexity and Organization*, **9**(1–2), 2007 pp. 120–130.

[6] F. Boschetti, D. McDonald, and R. Gray, "Complexity of a Modelling Exercise: A Discussion of the Role of Computer Simulation in Complex System Science," *Complexity*, **13**(6), 2008 pp. 21–28. doi:10.1002/cplx.20215.

[7] A. Baker, "Simulation-Based Definitions of Emergence," *Journal of Artificial Societies and Social Simulation*, **13**(1), 2010. http://jasss.soc.surrey.ac.uk/13/1/9.html.

[8] M. Bedau, "Weak Emergence," in *Philosophical Perspectives* (J. Toberlin, ed.), vol. 11, Oxford: Blackwell Publishers, 1997 pp. 375–399.

[9] J. Symons, "Computational Models of Emergent Properties," *Minds and Machines*, **18**(4), 2008 pp. 475–491. doi:10.1007/s11023-008-9120-8.

[10] F. Boschetti and R. Gray, "A Turing Test for Emergence," in *Advances in Applied Self-Organizing Systems* (M. Prokopenko, ed.), London: Springer-Verlag, 2007 pp. 349–364.

[11] F. Boschetti, "Causality, Emergence, Computation and Unreasonable Expectations," *Synthese*, **181**(3), 2010 pp. 405–412. doi:10.1007/s11229-010-9720-8.

[12] R. Rosen, *Life Itself: A Comprehensive Inquiry into the Nature, Origin, and Fabrication of Life*, New York: Columbia University Press, 2001.

[13] P. Wegner, "Why Interaction Is More Powerful than Algorithms," *Communications of the ACM*, **40**(5), 1997 pp. 80–91. doi:10.1145/253769.253801.

[14] J. Wiedermann and J. Van Leeuwen, "The Emergent Computational Potential of Evolving Artificial Living Systems," *Ai Communications*, **15**(4), 2002 pp. 205–215.

[15] R. Milner, "Elements of Interaction: Turing Award Lecture," *Communications of the ACM*, **36**(1), 1993 pp. 78–89. doi:10.1145/151233.151240.

[16] J. Symons, "Completeness and Metaphysical Fundamentalism," in *Systematic Thinking: Hintikka, Neville and the Future of Philosophy* (R. E. Auxier and D. R. Anderson, eds.), forthcoming.

[17] C. R. Shalizi and K. L. Shalizi, "Blind Construction of Optimal Nonlinear Recursive Predictors for Discrete Sequences," in *Uncertainty in Artificial Intelligence: Proceedings of the Twentieth Conference (2004)*, Banff, Canada (J. Halpern, ed.), Arlington, VA: AUAI Press, 2004 pp. 504–511.

[18] H. Pattee, "The Physics of Symbols and the Evolution of Semiotic Control," in *Workshop on Control Mechanisms for Complex Systems: Issues of Measurement and Semiotic Analysis*, Las Cruces, NM, 1996, Santa Fe Institute Studies in the Sciences of Complexity, Proceedings Volume, Redwood City, CA: Addison-Wesley, 1997.

[19] P. Menzies and H. Price, "Causation as a Secondary Quality," *The British Journal for the Philosophy of Science*, **44**(2), 1993 pp. 187–203. doi:10.1093/bjps/44.2.187.

[20] J. P. Crutchfield and K. Young, "Inferring Statistical Complexity," *Physical Review Letters*, **63**(2), 1989 pp. 105–108. doi:10.1103/PhysRevLett.63.105.