Journal of Experimental & Theoretical Artificial Intelligence
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/teta20

Algorithmicity and programmability in natural computing with the Game of Life as in silico case study

Hector Zenil[ab]

[a] Unit of Computational Medicine, Karolinska Institutet, Sweden

[b] LABoRES for the Natural and Digital Sciences, Paris, France

Published online: 02 Sep 2014.

Journal of Experimental & Theoretical Artificial Intelligence, 2015
Vol. 27, No. 1, 109–121, http://dx.doi.org/10.1080/0952813X.2014.940686

Algorithmicity and programmability in natural computing with the Game of Life as *in silico* case study

Hector Zenil*

Unit of Computational Medicine, Karolinska Institutet, Sweden; LABoRES for the Natural and Digital Sciences, Paris, France

In a previous article, I suggested a method for testing the *algorithmicity* of a natural/physical process using the concept of Levin's universal distribution. Here, I explain this method in the context of the problem formulated by L. Floridi concerning the testability of pancomputationalism. Then, I will introduce a behavioural battery of *programmability* tests for natural computation, as an example of a *computational philosophy* approach. That is to tackle a simplified version of a complex philosophical question with a computer experiment. I go on to demonstrate the application of this novel approach in a case study featuring Conway's Game of Life. In this context, I also briefly discuss another question raised by Floridi, concerning a grand unified theory of information, which I think is deeply connected to the grand unification of physics.

Keywords: programmability; information theory; algorithmicity; pancomputationalism; natural computation; Conway's Game of Life; nature-like computation; computational philosophy

1. Introduction

In his list of 'Open problems in the philosophy of information (PI)', Floridi (2004) opens up several fronts for discussion. In this article, I address two of them from the standpoint of algorithmic information theory, with a view to either identifying possible directions which the search for answers might take or reformulating the original questions in a bid to generate partial answers to them.

The first is the question of the testability of pancomputationalism. That is, the question of whether the world and the objects in it can be seen as computing processes and computing devices. Traditionally, one proves that a system performs (universal) computation by finding a mapping between the said system's states and symbols, which already assumes that one can have access to and can represent the system's states and symbols. This is of course particularly challenging for natural and physical phenomena, where we can hardly discern either symbols or states, making it impossible to suggest a one-to-one correspondence between a natural system and a (an artificial) computational system.

I address the problem of testability of pancomputationalism in three steps. I first present an *algorithmicity* measure based on a previous method I advanced in Zenil and Delahaye (2010) and discuss in Section 2. I then propose the notion of *programmability* in complementation to *algorithmicity* for testing the computing properties of natural and physical processes. The final

*Email: hzenilc@gmail.com

step consists of a case study focusing on Conway's Game of Life, where I illustrate the potential applications of the notion of programmability. All this culminates in a brief discussion of *computational philosophy* as a new approach to tackle new and old philosophical questions. I discuss specifically the question of a grand unified theory of information, which is addressed in Section 6 from the standpoint of other unifications in physics. The conclusion then highlights the innovative character of this new area of research, both as a new kind of philosophy based on information and computation.

2. The *algorithmicity* of empirical datasets

In Zenil and Delahaye (2010), we introduced a notion that I now refer to as *algorithmicity* (of a dataset), and suggested an approach for statistically testing whether physical and natural phenomena are the result of an algorithmic process by studying the way physical and natural processes distribute patterns. With an objective to find *algorithmic signatures* as predicted by algorithmic probability (Solomonoff, 1964) and formalised by Levin's (1974) universal (semi) probability distribution (the prefix *semi* is due to the fact that the measure does not add 1 for reasons related to the Halting problem) as a test for computationalism or the assumption that natural and physical phenomena are computational in nature.

In Zenil and Delahaye (2010), we took seriously the question of whether physical systems could be regarded as computational devices in which case it would then follow that natural phenomena should follow computational laws. One such law is Levin's (semi)probability distribution that dictates the distribution of patterns, that is, the probability of producing a certain output given a *typical* (random) computer program running on a (prefix-free) universal Turing machine.

The method consisted of looking at empirical data produced by physical phenomena to compare the frequency distributions to an experimental distribution produced by purely algorithmic means [and shown to be robust enough to compare with as demonstrated in Zenil and Delahaye (2010), Delahaye and Zenil (2011), Soler-Toscano, Zenil, Delahaye, and Gauvrit (2014) and Zenil, Soler-Toscano, Delahaye, and Gauvrit (2013)]. We can then proceed with a statistical comparison to test or reject the alternative hypothesis that distributions are correlated to algorithmic distributions (possibly because their source of the same computational nature).

Levin's distribution m provides a mean to define the probability of a string to be produced by a random program (whose instructions are chosen by coin tossing) running on a universal Turing machine. Formally, $m(s) = \sum_{p:U(p)=s} 1/2^{|p|} < 1$, i.e. the sum over all the programs p for which U, a prefix-free universal Turing machine, produces s. The coding theorem describes a connection between $m(s)$ and the Kolmogorov complexity (Kolmogorov, 1965; Chaitin, 1975) of s as follows: $K(s) = -\log_2(m(s)) + O(1)$. This theorem tells us that if a string s is produced by many programs, then there is also a short program that produces s, where $K_U(s) = \min\{|p|, U(p) = s\}$. That is, s is said to be random if it cannot be compressed and not random otherwise.

When observing the world, the outcome of a physical phenomenon f can be seen as the result of a natural process P. We may ask what the probability distribution of a set of processes such as P looks like. In a world of computable processes, $m(s)$ would establish the distribution of patterns. If we wish to know whether the world were algorithmic in nature we would first need to specify what an algorithmic world would look like.

In order to have a distribution to compare with, we calculated an experimental version of m that we later extended in Delahaye and Zenil (2011) and Soler-Toscano et al. (2014) (and that we called D) with another application to the problem of the evaluation of the Kolmogorov complexity K of short strings (a common challenge for compressibility, the traditional method for approximating K).

The Spearman rank correlation coefficient quantifies the strength of the relationship between the ranking order of two distributions, hence providing an objective measure of *algorithmicity* as follows:

> *Algorithmicity*. The greater the similarity between an empirical distribution of and the algorithmic distribution of patterns as predicted by Levin's semimeasure m, the greater the *algorithmicity* of the dataset and more likely to have been produced by an algorithmic (computational) process.

D is simply an approximation of Levin's m calculated using Turing machines (Delahaye & Zenil 2011). In Zenil and Delahaye (2010), however, we showed that the distributions produced with different models of computation (e.g. Turing machines, cellular automata and Post tag systems) were correlated (which accords with our intuitive sense of complexity – remember that m also provides information about the Kolmogorov complexity of the patterns in the distribution obtained through use of the coding theorem)

A large *algorithmicity* suggests that the generating process of a dataset is *algorithmic*, as defined by its degree of *algorithmicity* calculated by these means.

3. *Programmability* as a grading system for natural computation

In Zenil (2010), we showed that the empirical datasets studied had different degrees of correlation, offering a statistical measure for *algorithmicity*, that is, the likelihood that a process is *algorithmic* or *computational*. Another way to overcome the problem of having to define a mapping between systems in order to attribute to them a degree of computation is to approach the problem from a behavioural standpoint, that is, to ascertain whether we can make a system behave in a way that makes it comparable, in terms of its versatility, to a general-purpose computer. Hence, we make the assumption that central to the claim that something computes is the capability of a system to be reprogrammed.

In Zenil (2010), I suggested some measures for classifying the behaviour of Elementary Cellular Automata (ECA) and other systems into Wolfram's (2002) classes, for phase transition detection and for a novel measure of information transmission efficiency and variability that I have more recently proposed (Zenil, 2012a) in connection with a notion of *programmability*.

Its advantage as compared with *algorithmicity* is that it is computable, unlike Levin's m, and hence has a greater range of applicability, particularly to areas of natural computing. Taken together, however, the two approaches (*algorithmicity* and *programmability*) seem to offer sensible ways to approach the question of pancomputationalism that take it beyond the realm of a philosophical discussion, while ultimately providing insights that can enrich and perhaps boost any philosophical discussion we may wish to have.

Instead of trying to draw a crystal clear line between what is and is not a computer (or computation) one defines a measure of (*computedness*) of a natural/physical system. With a definition of programmability we can expect to be able to construct a hierarchy of *computing* devices, with digital computers appearing where they rightly should (at the top of the hierarchy), and objects which may be considered non-computing entities figuring at the bottom of the hierarchy. The idea of programmability put forward in Zenil (2010) is based on the notion of compressibility, which in turn is theoretically based on the concept of algorithmic (Kolmogorov) complexity, given that $K(s)$ is *upper semi-computable*, meaning that there is a sequence of lossless compression algorithms approximating $K(s) : C_1(s) \geq C_2(s) \geq C_3(s) \geq \ldots \geq K(s)$.

$K(s)$ cannot, therefore, be greater than the best compressed version of s. Programmability can then be defined as follows:

> *Programmability* is the ability of a system to change, to react to external stimuli (input) in order to alter its behaviour.

Our mathematical attempt to capture this concept is rather simple (or naive), and the measure can be adjusted (variations seem to capture different properties of the measured system). Assuming a Markov process, we can start by looking at the qualitative differences among different initial configurations as a measure of variability. Let M be a system and i_j an initial configuration for $j \in \{1, 2, 3, \ldots, n\}$. Then, the *variability* of M is given by (for some fixed n and t)

$$c_n^t = \frac{|C(M_t(i_1)) - C(M_t(i_2))| + \cdots + |C(M_t(i_{n-1})) - C(M_t(i_n))|}{t(n-1)}. \quad (1)$$

That is the sum of the absolute values of the differences of the compressed evolutions of a system M for the initial configurations $i_j, j \in \{1, \ldots, n\}$, divided by $t(n-1)$ for the purpose of *normalising* the measure by the system's 'volume' so that we can roughly compare different systems for different n and different t).

Evidently, programmability is related to this variability definition, given that a system with no apparent variability cannot be programmed. Now, let $\mathbb{C}$ be the result of fitting a line with regression analysis to the t data points determined by c_n^t for increasing t and taking the particle derivative of the fitting function $f(c_n^t)$.

$\mathbb{C}$ captures the asymptotic behaviour of M (or its average behaviour) as an indicator of the degree of programmability of M relative to its sensitivity to external stimuli. We provide an example of its application to Conway's Game of Life in Section 4. The larger the derivative, the greater the rate of change. $\mathbb{C}$ provides the programmability efficiency (the rate at which information can be transferred) when compared with other systems for fixed n and t. We can now associate the attribute of computation with natural/physical systems as follows:

> A system U is said to compute (and therefore is a computer) if $\mathbb{C}_n^t(U) > 0$ for long enough n and t. More precisely, U is a $\mathbb{C}_n^t$-computer for its computed $\mathbb{C}_n^t$.

Programmability is therefore a combination of behavioural variability and (external) controllability. We can see that the definition of computational universality is then simply, in these programmability terms, maximum control over the behavioural variability of a system. In Section 4, we see how the programmability of a system can be measured using a lossless compression algorithm.

4. A case study: Conway's Game of Life

An ECA is defined by a local function $f : \{0, 1\}^3 \to \{0, 1\}$. The function f maps the state of a cell and its two immediate neighbours (range = 1) to a new cell state: $f_t : r_{-1}, r_0, r_{+1} \to r_0$. Cells are updated synchronously according to f over the space.

The Game of Life is a cellular automaton devised by Conway (Gardner, 1970; Rendell, 2000). From its inception it has attracted much interest because of the surprising ways in which the patterns can evolve. The Game of Life traditionally unfolds on an infinite two-dimensional grid composed of cells each of which is either 'on/alive' or 'off/dead'. It is the best-known example of a cellular automaton. The rules of the Game of Life were carefully devised by Conway to mimic the behaviour of life (Figure 1; Gardner, 1970; Rendell, 2000).

The Game of Life takes place in discrete time, with the state of each cell at time t determined by its own state and the states of its eight immediate neighbours at $t - 1$ (Moore's

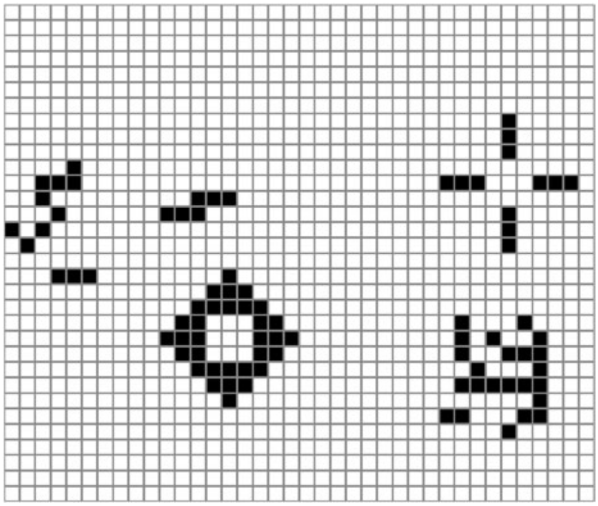

Figure 1. Example of patterns produced by the Game of Life after 100 steps (generations) starting from a random array of equally likely 0s and 1s of size 10×15.

neighbourhood of radius 1). A cell σ is represented by a 1 (black) when alive or a 0 (white) when dead, in an $m \times n$ array of cells. A calculation of the sum of live cells in σ_t's eight-location neighbourhood determines whether σ_{t+1} is alive or dead. These cell interactions can be organised to furnish enough computational versatility to implement logical gates and simple memory counters, thus proving to be capable of Turing computational universality (Berkelamp et al., 1982).

The initial conditions of Conway's Game of Life have been extensively studied from the point of view of the patterns they are able to produce, but little has been done to standardise the enumerations of initial configurations. Evolutions in the Game of Life have been grouped into a few classes depending on the kind of pattern (e.g. static, moving or periodic) that each configuration is able to produce. A pattern is any configuration of alive and dead cells. The LifeWiki (http://www.conwaylife.com/wiki/Main_Page, accessed in July 2012) lists 735 known patterns, including 348 'oscillators', 70 'spaceships' and 132 'still life'. Patterns such as 'gliders', 'puffer trains' and other such wonders arise from the interaction of simple components behaving according to well-defined rules.

In Zenil (2010), I proposed a Gray-code enumeration for one-dimensional cellular automata in order to avoid artificial complexity from making inroads into a system, given that I was investigating a system's sensitivity to initial conditions leading to phase transitions. For a two-dimensional cellular automaton such as the Game of Life, we have greater freedom to choose the way in which we feed the automaton, given that we can play with the relative density of 0s and 1s in the original array in order to feed the system with a smooth stream of different initial configurations. The Shannon entropy of a cellular automaton has been used before as a measure of the fitness (or *interestingness*) of a rule set (e.g. Kazakov & Sweet, 2005). In the same spirit, here we feed the Game of Life with arrays of increasing Shannon entropy. Figure 2 shows the compressed lengths of evolutions starting from four different random initial conditions, demonstrating that the behaviour of the system is robust, in that the compression rate remains about the same over the explored runtime.

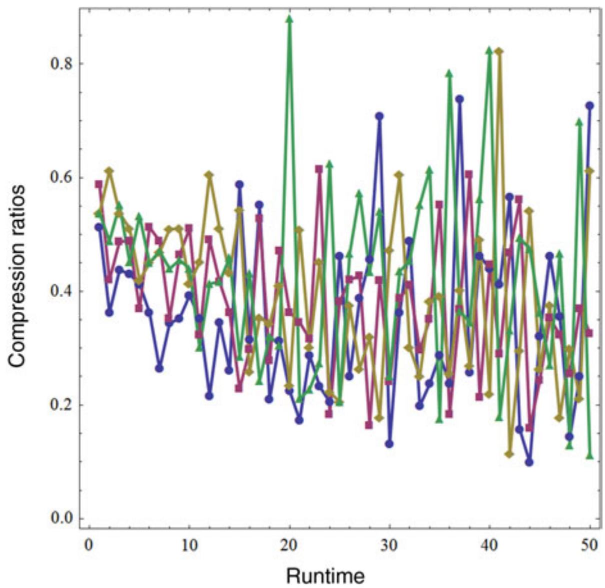

Figure 2. Compression ratios of the evolution of the Game of Life starting from four different random initial conditions of increasing Shannon entropy from 1 to 50 steps (generations) shows that the behaviour of the system remains complex over time and with no clear trend from information content of the initial condition.

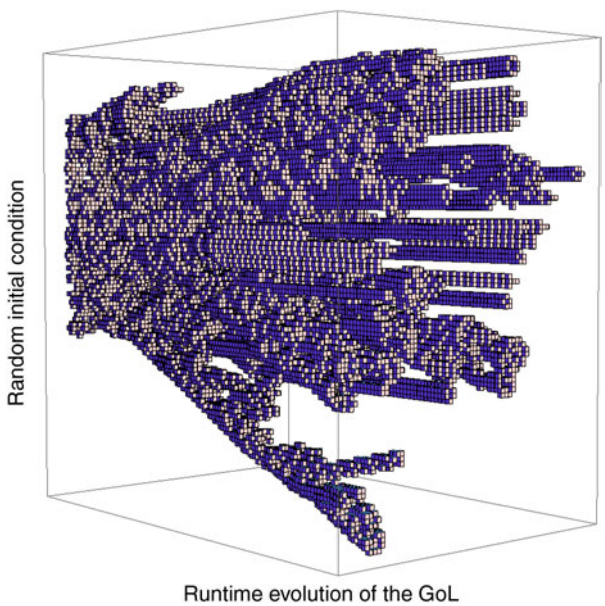

Figure 3. Space–time diagram of the evolution of the Game of Life for 100 steps (generations) starting from a randomly generated array of equally likely 0s and 1s of size 50×50. We notice still and moving patterns, some of them oscillatory, most forming persistent structures capable of transferring information over time.

In Zenil (2010), we calculated $\mathbb{C}$ as introduced in Section 3, capturing the asymptotic behaviour of one-dimensional systems (mainly cellular automata, but also Turing machines). The Game of Life is a two-dimensional cellular automaton, so the application of $\mathbb{C}$ should be over a 3D space as shown in Figure 3.

First, taking the compression rate using DEFLATE's compression algorithm[1] over an evolution of the Game of Life starting from a random initial configuration (a random array of equally likely 0s and 1s), we can see that the compressed length very closely follows the uncompressed length of the evolution of life (both versions being measured in bits). As explained in Zenil (2010), this is already an indication of the apparent complexity of the system, given that the compression algorithm is unable to find any regularities, as none of the patterns of the system died out after the period investigated (in the case of Figure 4 after 1000 generations).

Calculating $\mathbb{C}$ means looking at the compression length differences (see Figure 5) of *consecutive* initial configurations. In this case, consecutive initial configurations are those for which their Shannon entropy was increased by weighting the number of 0s and 1s in the initial array, and increasing the size of the array.

The final evaluation of $\mathbb{C}$ comes from linear fitting the points of c_n^t calculated as described in Section 3 and then taking the derivative (the slope) of the fitted line. The result, as expected, is close to the greatest values found in ECA (one-dimensional cellular automata with neighbourhood range 1), as was also conjectured in Zenil (2010, 2012b) in connection to Turing universality (that is, that systems capable of *efficient* universal computation should have *large* $\mathbb{C}$ values).

The calculation of $\mathbb{C}_{50}^{100} = 0.206$ as defined in Section 3, for the Game of Life, is consistent with the previous findings in Zenil (2010, 2012b), where cellular automata with large phase transitions and those assumed and then proved to be universal had large $\mathbb{C}$ values, reflecting their status as systems sensitive to their inputs and capable of transferring information (Figure 6).

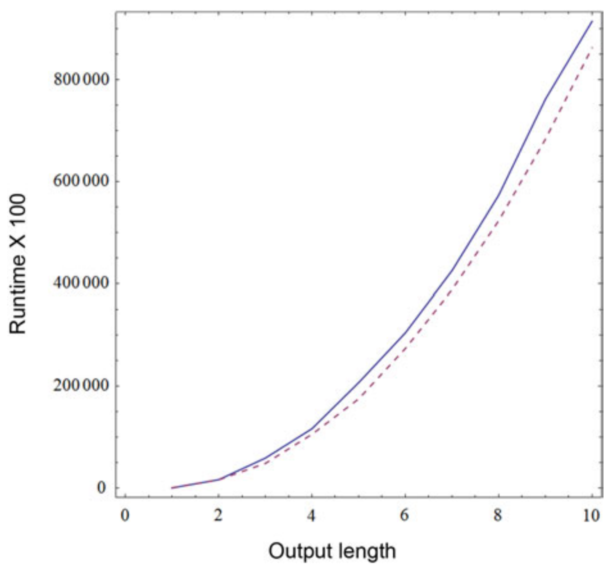

Figure 4. The compressed length of the evolution of the Game of Life (broken line) follows very closely the length of its uncompressed version (continuous line), serving as an indication of the asymptotic behaviour of the system for initial conditions of increasing complexity and longer runtimes.

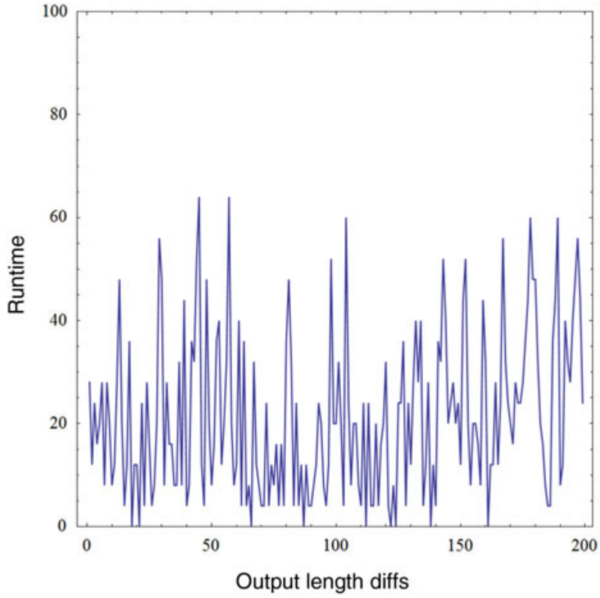

Figure 5. Plot showing the differences between compressed lengths capturing the variability of the evolution of the Game of Life.

This approach may work for abstract systems, but what about real ones? In Terrazas, Zenil, and Krasnogor (2013), we applied some of these programmability ideas to a real-world case in the context of an *in silico* investigation of porphyrin molecules, that is the molecules that

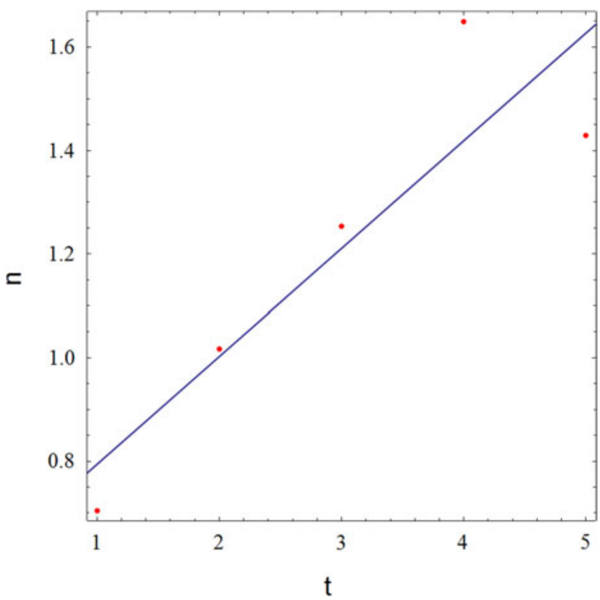

Figure 6. The line fitting the points produced by c_{50}^{100} resulted in a $\mathbb{C}_{50}^{100} = 0.206$ for the Game of Life, calculated as described in Section 3.

give the red colour to the animal's blood. We showed that we were able to simulate and predict some of the shapes in the conformational space of these molecules using these programmability tools based on algorithmic information theory. Moreover, properties studied in synthetic biology, such as orthogonality (the impact of perturbations over other free parameters), could successfully be investigated with the same tools. And in Zenil, Ball, and Tegnér (2013), we also studied the abilities of this programmability measure to spot differences between biological models related to the cell cycle or metabolic pathways with positive results.

5. The computer as a laboratory

In the original article popularising Conway's Game of Life, Gardner wrote:

> … Because of Life's analogies with the rise, fall and alterations of a society of living organisms, it belongs to a growing class of what are called 'simulation games' (games that resemble real life processes)….

In its details, the Game of Life stops far short of looking like a natural system, but it does capture many of the main properties of a phenomenon as complex as life, hence its name. Even though it may be an oversimplification of real life, the Game of Life is complicated enough that Conway originally conjectured that no pattern could grow indefinitely (Gardner, 1970; Rendell, 2000) – i.e. for any initial configuration with a finite number of living cells. He was wrong, and not only do we know today that because of the Turing universality the Game of Life (Berkelamp et al., 1982) is in fact capable of what Conway thought was out of the question, but also that most questions about such a simple system are undecidable. Wolfram (2002) and Dennett (1991) have used Conway's Game of Life and deterministic computational systems in general to illustrate possible complex philosophical formulations, relating to such matters as consciousness and free will, which may be the upshot of simple and deterministic laws governing our universe in the same way that simple and deterministic laws govern the Game of Life. We have used a computer as a cheap lab to tackle, albeit in a very limited way, a philosophical question using a computer experiment.

In the introduction of Bynum and Moor (1998), the authors acknowledge the emergence of a new trend in philosophical practice. The movement began when certain individuals undertook computer experiments with a philosophical motivation, perhaps even before the invention of modern digital computers, though no doubt spurred by them. Today, we can perform experiments to test simplified versions of certain philosophical claims with a view to either verifying or debunking them, in much the same way that thought experiments have been and continue to be used in philosophy. I think the trend has reached the point where it need no longer be confined to isolated computer experiments undertaken by philosophically minded researchers. Rather, a set of tools and methods can now be developed for more systematic use. Computing represents at once a new opportunity and a challenge to traditional philosophy, and will certainly shape the future of how we ask questions about the mind, free will, consciousness and knowledge, among other things. A few of us are pushing in this direction and we have embraced the term *computational philosophy*. I think computational philosophy has great potential for contributing insights into several of Floridi's (2004) questions, especially as they connect areas that are within the purview of computer science, such as artificial intelligence and information. This will not diminish the philosophical content of these subjects but will provide a common platform for discussion.

5.1 Life and nature-like computation

In fact, Floridi (2002) has defined the PI as the elaboration and application of information-theoretic and computational methodologies to philosophical problems, which turns out to be exactly what I think is the motivation of *computational philosophy*. Hence, I see the latter as a tool of priceless value to PI as defined by Floridi (2002).

In Zenil (2013), for example, I have shown that the theory of algorithmic probability can be connected to the question of structure formation in the context of Turing's interest on morphogenesis and symmetry breaking, hence reconnecting Turing machines and biological pattern formation by means of the theory of algorithmic information. I did this by actually exhibiting how patterns emerge from randomness by performing relatively simple, even if computationally expensive, experiments.

6. Natural computation and a Grand unified theory of information

Every time that there has been a unification in physics, what was previously thought to be a fundamental concept was stripped of its meaning. Time and space, for example, were thought to be fundamental and separate concepts of physical reality and their fundamental character is still debated (Wolchover, 2013), but general relativity deprived each of its unique character, unifying the two and making it possible to exchange one for the other as a single geometrical entity. As with general relativity, every unifying theory has transformed a concept that was thought to be physical into something informational. Space–time is now an abstract geometrical entity. There is no notion of absolute time since time has no meaning (or rather its meaning varies from one person to the next), nor is there a distinction between past, present and future in the laws of physics. The best explanation of the nature of space and time available today is a mathematical theory: general relativity. By doing so we have not managed to sort out every detail of a system's behaviour, even though these theories have an extremely impressive predictive power.

Something similar occurred with information and thermodynamics with the works by Szilárd (1929), Landauer (1961), Bennett (1973), Fredkin and Toffoli (1982) and Toffoli (1980), among others. They showed that there was a fundamental connection between information and energy – that one could extract work from information – and moreover that this connection could serve as one of the strongest arguments of essential value for a unification of information and physical properties. Their work provided a framework within which to consider questions such as Maxwell's paradox, and its possible answer, and of connections between computation, information and complexity (Bennett, 1973; Zenil, Gershenson, Marshall, & Rosenblueth, 2012).

As is known, Wheeler thought that it was possible to translate all physical theories into the language of bits. The framework of classical physics is based on a mechanical conception of nature, a conception which is mirrored in the digital model of computation. However, no account of information in relation to physics could be considered complete that does not take into account the possible interpretations of quantum mechanics. Wheeler was not only an information and computation enthusiast, he was also the quantum scientist involved in the identification of the smallest possible lengths in physics, at the quantum scales ($\sim 10^{-33}$ cm and $\sim 10^{-43}$ s) at which general relativity breaks down and should be replaced by, currently debated, laws of 'quantum gravity' (Wheeler is also credited with having coined the terms *Planck time* and *Planck length*, respectively, for these numerical length values). Wheeler (1990) thought that even quantum mechanics would eventually be rooted in the 'language of bits'.

For several reasons, black holes have come to play a vital role in the application of information theory to physics and in the modern quest for a grand unified theory of physics.

Black holes are at the intersection of our physical theory describing the largest objects in the universe, namely general relativity theory, and our physical theory describing the tiniest objects in the universe, namely quantum mechanics. This is because black holes are so massive that they produce what the core of Einstein's relativity theory predicts as being a consequence of gravity (the deformation of space–time due to a massive object), a singularity point with no length, which therefore subsists on a subatomic scale. The quest to unify the theories has therefore focused its attention on black holes.

There is another interesting connection of black holes to information, which involves the relation between the mass of what falls into a black hole and the size of the black hole itself (the surface area of its event horizon). As it turns out, black holes get more massive depending on what falls into them, but what a black hole does is somehow to compress to its minimum possible size whatever falls into it. So black holes are natural data compressors. But this also means that the size of the black hole serves as a physical compression limit of information (one can think of the black hole as providing the Kolmogorov complexity of the information of the objects falling).

It was perhaps not by chance that the same Wheeler who coined the term 'black hole' for the strange solutions that general relativity produced, leading to singularities, also coined the 'it from bit' dictum, suggesting that everything could be written in, and is ultimately composed of, bits of information. According to Misner, Thorne, and Zurek (2009), Wheeler's last blackboard contained the following, among several other ideas: 'We will first understand how simple the universe is when we recognise how strange it is.' Wheeler himself provides examples of the trend from physics to information in his 'it from bit' programme. It seems therefore that a 'grand unified theory of information' will not be that different from the grand unified theory of physics, even in the likely limitations that they may not overcome, unable of complete satisfactory explanations at every level of description and of full predictive power.

7. Concluding remarks

We have had a quick look at possible directions we may take in searching for answers to some of Floridi's open questions from an algorithmic information perspective and following a qualitative and very pragmatic behavioural approach. The measures introduced in Sections 2 and 3 provide a framework for a fruitful pancomputationalism discussion. Given that the ultimate answer may only, if ever, come from physics. I have suggested that one line of research is to approach the question by looking at the *a posteriori* output pattern distribution of a system in order to statistically test whether it resembles the output pattern distribution of a computational process. We can also test the behaviour of the system and treat its ability to react to external stimuli and transfer information as a metric of *programmability*. These two measures provide insight into the question of whether the world looks, at least, algorithmic, if not computational. This means we now have two pragmatic approaches to the question, and a way to determine whether something looks like a computation, and hence the means to differentiate between having and lacking such property to distinguish the things that appear to us to compute from those that do not. To provide a measure of *computedness* is to answer the question of what it would mean for a physical or natural system to be, or more importantly not to be, a computational system (see Floridi, 2008), given that the question has thus far seemed invulnerable to refutation.

We have also indicated how this approach is an example of *computational philosophy* that is the trend favouring the use of computer experiments to address even complex philosophical questions. And finally, we have briefly discussed, in the context of pancomputationalism, how the grand unified theory of information may turn out to go hand to hand with the grand unified theory of physics. That is, one will lead to the other – though again, not all questions will be answered. A grand unified theory of information will not provide all the answers to our questions about information, and especially not to the most keenly discussed questions in Floridi's PI, just as a grand unified theory of physics may not provide the means to unlimited predictability.

Beyond the philosophical questions, actual measures are needed for the kind of computation we are discovering in nature (starting from life's blueprint represented by the DNA code) and the kind of natural computation we are creating ourselves (for example, for reprogramming life), and it is a great opportunity to be able to tackle both the philosophical and the pragmatic issues at the same time and using the same tools.

Supplemental data

Supplemental material and source code available at http://www.complexitycalculator.com/FloridiGoLJETAI.zip

Note

1. DEFLATE is a lossless data compression algorithm that uses a combination of the LZ77 algorithm and Huffman coding. It is the standard algorithm used in many popular file formats such as.gif,.png and.gzip, to mention a few.

References

Bekenstein, J. D. (2003). Information in the holographic universe. *Scientific American*, *289*, 61.

Bennett, C. H. (1973). Logical reversibility of computation. *IBM Journal of Research and Development*, *17*, 525–532.

Berkelamp, E. R., Conway, J. H., & Guy, R. K. (1982). *Winning ways for your mathematical plays* (Vol. 2). New York, NY: Academic Press.

Bynum, T. W., & Moor, J. H. (Eds.). (1998). The Digital Phoenix: How computers are changing philosophy. Oxford: Wiley-Blackwell.

Chaitin, G. J. (1975). A theory of program size formally identical to information theory. *Journal of the Association for Computing Machinery*, *22*, 329–340.

Delahaye, J.-P., & Zenil, H. (2011). Numerical evaluation of the complexity of short strings: A glance into the innermost structure of algorithmic randomness. *Applied Mathematics and Computation*, *218–224*, 63–77 (printed version 2012).

Dennett, D. C. (1991). *Consciousness explained*. Boston, MA: Back Bay Books.

Floridi, L. (2002). What is the philosophy of information? *Metaphilosophy*, *33*, 123–145.

Floridi, L. (2004). Open problems in the philosophy of information. *Metaphilosophy*, *35*, 554–582.

Floridi, L. (2008). Trends in the philosophy of information. In P. Adriaans & J. van Benthem (Eds.), *Handbook of philosophy of information* (pp. 113–132). Amsterdam: Elsevier.

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

Gardner, M. (1970). Mathematical Games: The fantastic combinations of John Conway's new solitaire game "life". *Scientific American*, *223*, 120–123.

Kazakov, D., & Sweet, M. (2005). Evolving the Game of Life. *Adaptive Agents and Multi-Agent Systems II LNCS*, *3394*, 132–146.

Kolmogorov, A. N. (1965). Three approaches to the quantitative definition of information. *Problems of Information and Transmission*, *1*(1), 1–7.

Landauer, R. (1961). Irreversibility and heat generation in the computing process. *IBM Journal of Research and Development*, *5*, 183–191.

Levin, L. (1974). Laws of information conservation (non-growth) and aspects of the foundation of probability theory. *Problems of Information Transmission*, *10*, 206–210.

Misner, C. W., Thorne, K. S., & Zurek, W. H. (2009). John Wheeler, relativity, and quantum information. *Physics Today*, *62*, 40–46.

Rendell, P. (2000). Turing universality of the Game of Life. In A. Adamatzky (Ed.), *Collision based computing* (pp. 513–538). Berlin: Springer-Verlag.

Soler-Toscano, F., Zenil, H., Delahaye, J. -P., & Gauvrit, N. (2014). Calculating Kolmogorov complexity from the frequency output distributions of small turing machines. *PLoS ONE, 9, e96223*.

Solomonoff, R. J. (1964). A formal theory of inductive inference: Parts 1 and 2. *Information and Control*, *7*, 1–22; 224–254.

Szilárd, L. (1929). Über die Entropieverminderung in einem thermodynamischen System bei Eingriffen intelligenter Wesen [On the reduction of entropy in a thermodynamic system by the interference of intelligent beings]. *Zeitschrift für Physik*, *53*, 840–856.

Terrazas, G., Zenil, H., & Krasnogor, N. (2013). Exploring programmable self-assembly in non DNA-based computing. *Natural Computing*, *12*, 499–515. doi:10.1007/s11047-013-9397-2

Toffoli, T. (1980). *Reversible computing. Technical Memo MIT/LCS/TM-151*. Cambridge, MA: MIT Laboratory for Computer Science.

Wheeler, J. A. (1990). Information, physics, quantum: The search for links. In W. Zurek (Ed.), *Complexity, entropy, and the physics of information*. Redwood City, CA: Addison-Wesley.

Wolchover, N. (2013). A jewel at the heart of quantum physics. *Quanta Magazine*. Retrieved from https: www.simonsfoundation.org/quanta/20130917-a-jewel-at-the-heart-of-quantum-physics/

Wolfram, S. (2002). *A new kind of science*. Champaign, IL: Wolfram Media.

Zenil, H. (2010). Compression-based investigation of the dynamical properties of cellular automata and other systems. *Complex Systems*, *19*(1), 1–28.

Zenil, H. (2012a). Nature-like computation and a measure of computability. In G. Dodig-Crnkovic & R. Giovagnoli (Eds.), *Natural computing/unconventional computing and its philosophical significance, Studies in Applied Philosophy, Epistemology and Rational Ethics*. Heidelberg: Springer.

Zenil, H. (2012b). On the dynamic qualitative behavior of universal computation. *Complex Systems*, *20*, 265–278.

Zenil, H. (2013). Turing patterns with turing machines: Emergence and low-level structure formation. *Natural Computing*, *12*, 291–303.

Zenil, H., Ball, G., & Tegnér, J. (2013). Testing biological models for non-linear sensitivity with a programmability test. In P. Liò, O. Miglino, G. Nicosia, S. Nolfi, & M. Pavone (Eds.), *Advances in artificial intelligence, ECAL 2013*. Cambridge, MA: MIT Press.

Zenil, H., & Delahaye, J. P. (2010). On the algorithmic nature of the world. In G. Dodig-Crnkovic & M. Burgin (Eds.), *Information and computation*. Singapore: World Scientific.

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

Zenil, H., Soler-Toscano, F., Delahaye, J.-P., & Gauvrit, N. (2013). Two-dimensional Kolmogorov complexity and validation of the coding theorem method by compressibility. arXiv:1212.6745 [cs.CC].