

Laws in Philosophy of Info-Computationalism [†]

Shanshan Fan

School of philosophy, Renmin University of China, Beijing 100872, China; fanshanshan@ruc.edu.cn

[†] The Fourth International Conference on Philosophy of Information, Berkeley, California, USA, 2–6 June 2019.

Published: 14 May 2020

Abstract: What are laws of nature? This issue has always been one of the core issues of philosophy of science. Info-computationalism uses algorithms and information to explain laws of nature, and analyses its nature from the perspective and invariance, which opens up a new path for laws of nature. Therefore, this paper, based on the theory of info-computationalism, compares them to Cartwright's concept of laws of nature. It is found that we can follow the laws in computers to understand the laws of nature, and regard the laws of nature as human, metaphorical, prescriptive and creative products.

Keywords: laws of nature; info-computationalism; abstraction

1. Introduction

In general, the main purpose of science is to reveal the regularity of things and find the laws of nature. In philosophy of science we usually call the laws of nature laws. Many important concepts depend on laws in philosophy. However, in Van Fraassen's view, laws of nature face the problem of inference and the problem of identification (van Fraassen 1989) [1]. In the face of this dilemma, the emerging info-computationalism, especially Timothy Colburn and Cary Shute, published several articles on the concept of laws in Computer Science in recent years, adopted the interpretation of algorithm and information, and analyzed the nature of computer laws from the perspective of normalization and invariance. Their viewpoints can inspire and help to solve the problem of laws in philosophy of science. Therefore, based on the theory of info-computationalism, this paper discusses the concept of laws.

The second part of this paper analyzes Timothy Colburn and Cary Shute in recent years, the study of laws and their philosophical basis in Computer Science—Floridi's information structural theory (Floridi 2008a) [2]; the third part is from the computer law to the discussion of laws of nature. This part points out that once we understand laws of nature as artificial, metaphorical, prescribed, dynamic and nomological machines like computers, it will help to solve many difficulties faced by laws of nature in philosophy. The conclusion part summarizes that we should understand laws of nature as the result of being “man-made”, and this will not make us deny the objectivity and universality.

2. The Computer Law of Timothy Colburn and Gary Shute

Timothy Colburn and Gary Shute are scientists in the field of computer science at the University of Minnesota. At the same time, they also explore logic, models and entities in the philosophy of science based on computers. Since 2007, they have published articles to discuss laws, algorithm structure and other issues in computer science.

As to whether there is a problem of computer law, their answer is “there is”. Colburn and Shute believe that computer science is full of physical metaphors, especially those centered on flow and motion. When people see the operation inside the computer, in fact, the only thing that really moves

is the cooling fan and disk drive (Timothy Colburn and Cary Shute, 2010a) [3]. Although information does not flow like liquid, this metaphorical movement makes them think that there are laws of computer science. Because they are metaphorical, the laws of computer are invented and created by programmers, which is also prescriptive. In addition, laws of computer science have invariance, which can regulate and constrain the programming behavior, so as to achieve the goal more easily. Just as laws of nature admit no exceptions, when programmers make laws for their abstract world, they have to make sure that they admit there is no invariance. As for what is invariance, they provide an example of summing array in their article (Timothy Colburn and Cary Shute, 2010b) [4].

According to Colburn and Shute, if they want to know the role of laws in computer science, it is necessary to further think about the philosophical basis for computer scientists to explore the information world, which is mainly composed of abstract concepts, which can be explained by the information structural realism of Floridi. Therefore, we can see that they want to take Floridi's philosophy of information as the philosophical basis and foundation of their research on the concept of computational laws. In his method of levels of abstractions, Floridi put forward a non-descriptive view of realism. In the view of Colburn and Shute, the core method of Floridi research on information philosophy is to draw lessons from the object-oriented programming method in software engineering, and the object-oriented programming method also provides methodological support for the law of human creation (Timothy Colburn and Cary Shute, 2011) [5].

Does the law of the computer pursue truth like the law of nature? In Colburn's and Shute's view, computer science, as an engineering science, does not seek to reveal the objective truth of nature, but seeks to help control the value of the computing process (Timothy Colburn and Cary Shute, 2010a) [3]. On the question of what is "value", they give examples of bridge theory and object-oriented programming design (OOP). For example, to build a bridge in a given situation depends on different values of its beauty, cost, bearing capacity and environment. According to different situations, engineers need to measure these different values. Here, the basic knowledge of building bridges can be understood as laws. On the premise of following the basic principles of laws, the goal of engineers is to measure between different values to build different types of bridges. Therefore, they believe that computer laws can guide us to build efficient algorithms, establish reasonable values of data structure, establish expressive programming languages, operate reliable and safe systems, and design sophisticated computer architecture (Timothy Colburn and Cary Shute, 2010a) [3].

Through the above analysis, it is not difficult to find that in Colburn's and Shute's view, there are laws of computer science, and its essence is algorithm and information. Its biggest feature is its invariance and prescriptiveness—that is, there are programmers to formulate and standardize the programming process. They vividly answered how the laws of computer science are possible, but they think that although computer laws have many similarities with laws of nature, the former is created by programmers so that their laws are prescriptive; the latter is discovered by scientists, so the laws are descriptive to describe natural phenomena. In addition, they think that laws of nature and the computer laws have different goals. The former aims to discover the truth, while the latter aims to produce values. In other words, the former pursues natural knowledge, while the latter focuses on knowledge about effective values.

3. From Computer Laws to Laws of Nature

On the basis of them, this paper further believes that the laws of computer not only have many things in common with the laws of physics, but also can help us rebuild our understanding of the concept of laws of nature. Moreover, the difference between the laws of nature and the laws of computers that they emphasize is not the case in this paper. In particular, they think that laws of nature do not seek values. In this paper, the bridge theory they exemplified is the same as Lewis's best deductive systems (Lewis 1973, 73) [6] and Mitchell's Dimensions of Scientific Law (Sandra D. Mitchell, 2000) [7]. The bridge should be balanced between aesthetics and bearing capacity, while the law should be balanced between simplicity, strength and fitness, or between stability, strictness and

abstraction. At the same time, the laws of computer science can also be applied to our study of natural science, especially the thinking of laws of nature. Moreover, once we understand laws of nature as artificial, metaphorical, prescriptive, dynamic and nomological mechanisms like computers, it will be helpful to solve many dilemmas faced by laws of nature.

This is consistent with many new theories of the current philosophers of science, especially Cartwright. We can get many new explanations about the problem of laws of nature from the laws of the computer. When we understand laws of nature as computer laws, even this is in line with Woodward's law view based on invariance under intervention (Woodward, 2000) [8]. The so-called intervention is the operation in computer science, and the so-called invariance is the invariance of computer science we mentioned above. In addition, the laws of computer are very similar to those of Cartwright. This can be explained in three aspects. First, the metaphor of computer laws is similar to the "unreal" of Cartwright's physical law [9]. The metaphors of motion and flow of computer laws, although in fact information does not really flow, can help computer scientists to formulate accurate mathematical conditions in information throughput, so as to better explain and realize the calculation process. So, in fact, the law of computer is the same as the law of physics put forward by Cartwright, which is the exchange between reality and explanation. Moreover, from the point of view of computer law, it pays more attention to the explanation of the law, without considering whether or not it is true. With the help of this characteristic of the computer, we can further point out that the nature of physical laws or laws of nature should also be a kind of metaphor to better explain and complete the process of scientific experiments, and to make laws have a more powerful and extensive explanatory force by neglecting its truth. Second, the programmer design of laws of computer science is the same as the artificial creation of Cartwright's laws of nature. Computer laws are created by programmers and are prescriptive. Cartwright thinks that the laws of physics are created or designed by physicists in the laboratory, from which we can reasonably infer that the laws of nature are also prescriptive—that is, scientists formulate and regulate the scientific experiment process. Third, the production process and properties of computer laws and the machines used are similar to those of the laws of nature. Therefore, as mentioned before, to a certain extent, we can understand "computer" as the laws' nomological machine that produces the laws of computer.

Finally, when laws of nature, like those in computer science, are regarded as the result of human creation, will laws' universality and objectivity be denied? This paper reveals that the answer to this is no when we say it is the result of artificial construction, because it is a rational model. In other words, the laws of nature are abstract concepts and rational models, which are universal. However, it needs to be emphasized that when we talk about laws of nature, as an ideal model, it just extracts some aspect of the nature from the object and talks about the nature and commonality, while its other properties and attributes are not separated from itself in any way. Therefore, in the ontological sense, we can say that the laws of nature exist or do not exist. Existence is because it is contained in nature and is one of its many attributes. Non-existence is because we cannot separate from any other attributes in nature to make it exist. Therefore, these laws can also be said to be untenable in ontology. A proposition that can and cannot be discussed in ontology is meaningless. Thus, the laws emphasized in this paper are all epistemological.

4. Summary

Through Colburn and Shute's analysis of the laws of the computer, it further proves that laws of nature are the nature of the artificial picture constructed by our thinking. Therefore, we can reasonably assume that laws of nature are artificial, metaphorical, prescriptive, dynamic and nomological mechanisms.

References

1. van Fraassen, B.C. *Laws and Symmetry*; Oxford University Press: New York, NY, USA, 1989; pp. 38–39.
2. Floridi, L. A Defence of Informational Structural Realism. *Synthese* **2008**, *161*, 219–253.
3. Colburn, T.; Shute, G. Knowledge, truth, and values in computer science. *Int. J. Vallverdu* **2010**, 119–131. doi:10.4018/978-1-61692-014-2.ch008.

4. Colburn, T.; Shute, G. Abstraction, law, and freedom in computer science. *Metaphilosophy* **2010**, *41*, 345–364.
5. Colburn, T.; Shute, G. Decoupling as a Fundamental Value of Computer Science. *Minds Mach.* **2011**, *21*, 241–259.
6. Lewis, D. *Counterfactuals*; Harvard University Press: Cambridge, UK, 1973; p. 73.
7. Mitchell, S. Dimensions of Scientific Law. *Philos. Sci.* **2000**, *67*, 260–263.
8. Woodward, J. Explanation and invariance in the special sciences. *Br. J. Philos. Sci.* **2000**, *51*, 197–254.
9. Cartwright, N. *The Dappled World: A Study of the Boundaries*; Cambridge University Press: Cambridge, UK, 1999; p. 46.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).