Complexity Beyond Computation

Liu Yuting

October 14, 2022

1 About Complexity

To appreciate the study of complexity, we may start with what Feynman held as the most important knowledge humans possess: Everything is made of atoms. Atoms form molecules. Molecules form biological cells. Biological cells form organs. Organs form the human body. The human body consists of vast amount of atoms working magically together. The study of complexity is to explain that magic.

So far, no scientific results fully achieve the goal set in the opening paragraph. Although there are claims that some notion solves the entire mystery of complexity, all are baseless and we will refute one such claim in this manuscript.

1.1 Observations

Although the simplicity of modern user interfaces hide great amount of complexities from us, we sometimes still experience the consequences of complexity. Weather can be unpredictable. Financial markets can be volatile. The fate of the world can be uncertain.

Sciences have found ways to cope with complexities. Axiomatic systems derive complex statements from simple axioms. Complex motions may be reduced to a set of physical laws. Darwinian evolution is said to account for the complexities of life. These explanations, though successful, all suffer deficiencies. Gödel proved the limitation of axiomatic systems in deriving all mathematical truths in second order arithmetic. Quantum mechanics and general relativity can not easily reach coherence with each other. The similarity between certain biological forms and automata seems to suggest life has a algorithmic aspect not explored by Darwin.

It's against this unsatisfactory background that modern complexity studies emerged. Compared with scheme theory in algebraic geometry, complexity studies feel like mud. It's not because the subject matter isn't attractive, but we don't have very great tools at hand.

1.2 Modern Complexity Studies

Because there is no consensus how to define complexity, modern complexity studies diverge in subject matter as well as methodology. For some, complexity means randomness. For others, it means logical depth. The lack of agreement on the definition of complexity makes summarizing the literature difficult, so here only a few well-established approaches are described.

1.2.1 Chaos

The study of chaotic dynamics features unpredictability and randomness in solutions to differential equations or iterative maps, that their trajectories are highly sensitive to initial conditions. It's often said that the long-term behavior of chaotic orbits is unpredictable, but there are many possibilities what 'long-term' means. From time to time, it may be 'eventually' or 'exponentially' depending on the dynamical system under consideration.

Although chaotic orbits may seem a mess, hope to gauge their evolution isn't lost. Occasionally, the existence of bounds, topologies, or geometries can be proven to describe the long-term stability of chaotic orbits, that is, they may be unpredictable, but won't fall apart.

A interesting question in chaos theory is when and how a system transitions from order to chaos. Although it's hard to answer such questions, measures of order/disorder can often be found so that we can see a transition indeed occurs.

Probably one of the most astonishing results in chaos theory is Sharkovskii's Theorem, that there is a hierarchy of periods in chaotic maps. Despite being in disorder, there is still structure!

Overall, chaos theory's contribution to complexity study is how unpredictability happens. There is no simple way to accurately pin down the long-term behavior of chaotic orbits.

1.2.2 Computational Complexity

The main measure of computational complexity is logical depth, that a complex procedure requires more logical steps than a simple one. Although the definition may seem elementary, the strength of computational complexity is greatly amplified by the existence of universal Turing machines. Since universal Turing machines can simulate all computations, there is a unified method to generate all complexities in this characterization.

A obvious application of computational complexity is cryptography. A encryption method is good if it requires astronomical logical steps to break it.

But, the computational approach runs much deeper into sciences. Take biology for example. Since genetic materials are discrete data structures, many biological patterns found in nature resemble the result of a algorithm applied to genetic data. In fact, thinking in terms of code has been hugely successful in molecular biology, that DNA codifies proteins.

Measuring in logical depth, the computational approach allows a systematic classification of complexity in terms of *P*, *NP*, *PSPACE*, etc. The class *NP* is particularly interesting as there are a plethora of *NP*-complete problems that if one of them falls in *P*, the entire *NP* is equal to *P*.

1.2.3 PDE

The PDE approach to complexity study is often based on physical laws. Whether it's linear Schrödinger equation, or nonlinear Navier–Stokes equations, PDE gives a concise rule how such systems assemble, interact, and evolve.

Contrary to the computational approach, one significant drawback of the PDE approach is that there is no unified machinery in finding solutions. However, whenever applicable, like in the case of Schrödinger equation, the solution often provides great insights into how complexity arises. Indeed, the whole study of chemistry may be put on a solid foundation of quantum mechanics.

The difficulty in solving PDEs indicates that nature appears to be infinitely subtle, capable of effortlessly generating solutions with high accuracy. Why can nature solve PDEs that beset mathematicians?

1.3 Sense of Complexity

Since there is no agreement what complexity is, it's impossible to satisfactorily categorize the concept. Instead, there are sensible features when complexity is encountered. Here, a few such features are discussed.

1.3.1 Hierarchy of Hardness

There is a hierarchy of hardness in machineries. If a machine A can do whatever another machine B can do and more, then A is more complex than B. A example is the progression from linear bounded automata to Turing machines. A partial order system is naturally formed according to the functionalities of machines.

1.3.2 Nonlinearity

Nonlinearity is a very broad term. Usually, nonlinear systems are complex because the superposition of states isn't valid. These systems can generate complex output from simple input. A particular example is cellular automaton. There are many patterns in nature that resemble cellular automata.

1.3.3 Unpredictability and Randomness

The logistic map exhibits period doubling to chaos. The increase in unpredictability and randomness signals increase in complexity. Shannon entropy and Kolmogorov complexity are useful measures of randomness.

2 The Computational Paradigm

There are many approaches to complexity study. However, one approach, that of computation, finds power to greatly expand into other areas.

2.1 Advantages of Computational Complexity

To understand the success of the computational paradigm, it requires a lot of technical work. Here a brief explanation is provided to clarify how two previously mentioned approaches to complexity study, chaos and PDE, may be covered by computation.

At first glance, the divergent behavior paramount in chaos seems to prohibit all sorts of computational approach. However, with the aid of topological estimates like the shadowing lemma, one may prove that although the true trajectory of a given initial condition may not be computed, a computed trajectory may still be very close to a true trajectory of a very close initial condition. Thus, computation tells something about the system.

As for PDE, the success of computational fluid dynamics not only find applications in vehicle designs, but more importantly, demonstrates the need for more powerful computers to solve more complex CFD problems. There seems to be a correspondence between computational power and complexity of the problem under consideration.

Overall, computation provided concrete solutions to abstract theories. In this way, it gradually became the primary tool for complex study.

2.2 Principle of Computational Equivalence

With the establishment of the computational paradigm, radical ideas began to emerge. One such idea is Stephen Wolfram's Principle of Computational Equivalence.

While traditional sciences view computation as a approximation to a specified model, Wolfram views computation as essential, and traditional models are approximations. The tremendous power of this formulation lies in the fact that there are universal Turing machines that can simulate all computations. Therefore, there is a unified mechanism to generate all complex phenomena.

However, this seems too good to be true, and it's the deficiencies in Wolfram's Principle of Computational Equivalence that lead to considerations beyond the computational paradigm.

3 Beyond the Computational Paradigm

To put it simply, if complexities are computations, then life can be generated on a ink-and-tape Turing machine, which is absurd. Here technical reasons why Principle of Computational Equivalence is wrong are given.

3.1 Deficiencies of the Computational Paradigm

Pretty much of the deficiencies of the computational paradigm can be justified by common sense. Though, it's not clear what's the path forward for the definition of complexity.

3.1.1 Cracks from Within

The computational paradigm highlights the importance of Turing-completeness. However, Turing-completeness alone suffers scale problems. For example, both ink-and-tape Turing machines and sophisticated semiconductor chips can perform identical computations, but semiconductor chips are a lot more complex and much faster. The computational paradigm doesn't address the discrepancy. There are serious implications. If computers were all in the form of ink-and-tape Turing machines, there would be no iPhone.

Quantum computation provides another critique of the computational paradigm. Although both classical and quantum computers can be Turing-complete, there are quantum algorithms that run much faster than classical algorithms. That is, time complexity much depends on architecture.

3.1.2 Cracks from Without

Modern spacecrafts have computer chips. However, it's a fallacy to declare that these spacecrafts are simply computers. Ordinary computers don't go to Mars. Therefore, although ordinary computers and spacecrafts may possess identical computational power, their complexities can be quite different. While ordinary people can perform computations like computers, albeit inefficiently, most people don't have the knowledge of designing complex spacecrafts for Mars.

Specifically, Principle of Computational Equivalence ignores statistical concepts like evidence and geometrical concepts like dimension. It claims that they are irrelevant if complexity is the main concern. Going to Mars is simply as complex as pencil-and-paper computation, which is absurd.

In fact, recent research showed that knots may be put into a partial order system according to some definition of complexity called ribbon concordance. It uses basic tools like Morse theory. It's hard to dismiss such results as being irrelevant.

3.2 A Application

While the computational paradigm provides complexity hierarchy according to computational power, it says nothing about the possible interaction between computers of different capabilities. Such interactions are possible because of computers can be topologically separated. It's these interactions that made the design of Apple's Secure Enclave, and similar constructs, possible. Therefore, going beyond the computational paradigm can be fruitful.

3.3 Conclusion

The status of complexity study is still kind of muddy. No universal definition of complexity can be given, no great tools can be applied, and no approach is satisfactory. But if we are going to understand how the world works as a whole, complexity study must be undertaken.