Syrian Arab republic

Ministry of Education

National Center for the Distinguished

Complexity

Khaled A.Ismaeel

## Introduction

Complicated, Complex, Simple. Some everyday words that we take for granted. Usually we don’t think about what do they mean, they’re obvious … are they? In fact, if you try to think about these little words a little, if you think outside the dictionary, you’ll find some of the greatest question in human history, Questions that baffled philosophers, physicists, biologists, engineers and mathematicians. All inscribed inside the concept of complexity. So, what is it? Where does it come from? Where does it lead to? Is our world complex? Or is it not?

## The Study of Complexity

It seems a bit strange to talk about the study of complexity before talking about complexity itself. But, as we shall see, the definition of complexity differs from one scientific point of view to another. Neil Johnson admits that: “even among scientists, there’s no unique definition of complexity-- and the scientific notion has traditionally been conveyed using particular examples...”Ultimately, he adopts the definition of ‘complexity science’ as “the study of the phenomena while emerges from a collection of interacting objects”

The definition of complexity often depends on the concept of a system –a set of parts or elements that have different kinds of relationships between each other beside relationships with other elements outside the relational regime-. Many definitions tend to assume that complexity expresses a condition of numerous elements in a system with numerous forms of relationships among them. However, what one sees as complex and one sees as simple remains a matter of perspective. Warren Weaver posited in 1948 two forms of complexity: disorganized complexity, and organized complexity.

Other forms of complexity are studies without understanding the concept of “system” such as computational complexity, algorithmic complexity and programming complexity, as we shall see later.

## Disorganized Complexity -VS- Organized Complexity

One of the problems in defining complexity has been formalizing the conceptual distinction between the large number of variances in relationships extant in random collection of elements, and the smaller number of relationships between elements in systems constraints (related to correlation of otherwise independent elements) simultaneously reduce the variations from element independence and create more-uniform, or correlated, relationships and interactions.

Weaver worked on this problem, and he gave a reasonable solution, by drawing distinction between “disorganized complexity” and “organized complexity”.

In weaver’s view, disorganized complexity results from a system having a very large number of elements, say millions, or even more, and the interactions or relationships between these elements seem to be largely random i.e. you can’t precisely predict what an elements state is going to be at a given time. However, you can predict the behavior of the entire system as a whole, using probability and statistical methods.

A prime example of disorganized complexity is a gas container as a system, and the gas molecules as the elements. Usually, this complexity is compared to the relative ‘simplicity’ of planetary orbits and the solar system, in which the elements state can predicted precisely using Newton’s laws of motion.

In contrast, organized complexity, in Weavers view, comes from non-random or correlated interaction between the elements in a given system.an example of this concept is a city as system, and the habitants as the element. Indeed, the interactions between the people are not random, and therefore this system is very different from the one we talked about in the gas container.

## Computational Complexity

In computational complexity theory, the amount of resources for the execution of an algorithm is studied. Computer engineers are specifically interested in time complexity and space complexity.

An algorithms time complexity is the number of steps the algorithm executes for a given input represented as a function of the size of the input.

While space complexity is the volume of the memory required for the execution of the algorithm represented as a function of the size of the input.

A Beautiful example of time complexity is the popular Merge Sort algorithm. Merge Sort is an algorithm for sorting an array with  numbers (integers or floating points) and it has  complexity i.e. if you entered an array with  numbers, it would take  seconds for the array to be sorted for some constant  (depending on the machine you’re using).

The study of computational complexity and algorithms running times is a very important field in computer science and electrical engineering, because programmers are always looking for the most efficient algorithm there is, to maximize the powers of computation.

Let’s consider the Insertion Sort algorithm, which is another algorithm for sorting arrays of numbers. The Insertion Sort algorithm has  complexity. If we’re sorting a 5-element array, then there’s almost no difference between Merge Sort and Insertion Sort algorithms, however, when we’re working on a much larger array, say a 10,000-element array, then Merge Sort algorithm has  running time for some constant, while Insertion Sort algorithm has  running time for some constant, the difference is quite large. If we increase the number of elements, it’s no longer a matter of which one sorts faster; rather it becomes a matter of which one sorts at all.

## Kolmogorov complexity

Kolmogorov complexity (also called Descriptive complexity) of an object is the measure of the computability resources required to specify that object. In the scope, we shall stick to strings as objects for simplicity.

Kolmogorov complexity of a string is the length of the shortest description of that string using some universal language. If we’re using a programming language (C, Lisp, Java …), we can define Kolmogorov complexity of a string as the length of the shortest program to print out that string.

Take this string for example:

abababababababababababababababab

There’s a very simple English description for this string, simply “ab 16 times” which is 11 characters. If we a look at a more complex string:

PweNJnrot82t93fgf934n982fds

There’s no simple description for this string, the shortest description is to print out the entire string as it is.

Another way for looking at those strings is by using pseudo code; we can wright down a simple program for printing out the first string.

*for i=0 to 16*

*write(ab)*

which is 24 characters long, while the shortest way for printing out the latter is

*write(PweNJnrot82t93fgf934n982fds)*

An interesting aspect of Kolmogorov complexity is the ability to compress strings. Indeed, as we’ve shown above, the first string was 32 characters long, and it was compressed to an 11-character string, while the latter string could not be compressed due to its high complexity.

## Other Types of Complexity

There are many more ways of looking at the concept of complexity. We won’t go through all of them in this scope; instead, we shall give a brief description of some.

### Mathematical Complexity

One field of discrete algebra is Krohn-Rhodes theory, and it’s related to studying finite semi groups and automata.

### Complexity in Physics

In physics, complexity is the measure of probability of the state vector of the system. This should not be confused with entropy; rather it is a distinct mathematical measure, one in which two distinct are never conflated and considered equal.

### Programming Complexity

In software engineering, programming complexity is a measure of the interactions of the various elements of the software. This differs from the computational complexity described above in that it is a measure of the design of the software.

## Conclusion

What we’ve done above is to clarify and explain the concept of complexity in several scientific fields. Even though there’s no definite definition for complexity, it remains a beautiful thing to study in deferent sciences. Studying the concept of complexity is in fact the main goal of Complex Systems Theory, and we gave a brief introduction to this vast scientific field.

## References

* 1. [**Jump up ^**](http://en.wikipedia.org/wiki/Complexity#cite_ref-1) J. M. Zayed, N. Nouvel, U. Rauwald, O. A. Scherman. *Chemical Complexity – supramolecular self-assembly of synthetic and biological building blocks in water*. Chemical Society Reviews, 2010, 39, 2806–2816 <http://pubs.rsc.org/en/Content/ArticleLanding/2010/CS/b922348g>
  2. ^ [Jump up to: ***a***](http://en.wikipedia.org/wiki/Complexity#cite_ref-Neil_Johnson_2-0) [***b***](http://en.wikipedia.org/wiki/Complexity#cite_ref-Neil_Johnson_2-1) Johnson, Neil F. (2009). "Chapter 1: Two's company, three is complexity". [*Simply complexity: A clear guide to complexity theory*](http://www.uvm.edu/rsenr/nr385se/readings/complexity.pdf). Oneworld Publications. p. 3. [ISBN](http://en.wikipedia.org/wiki/International_Standard_Book_Number) [978-1780740492](http://en.wikipedia.org/wiki/Special:BookSources/978-1780740492).
  3. ^ [Jump up to: ***a***](http://en.wikipedia.org/wiki/Complexity#cite_ref-Weaver_3-0) [***b***](http://en.wikipedia.org/wiki/Complexity#cite_ref-Weaver_3-1) Weaver, Warren (1948). ["Science and Complexity"](http://people.physics.anu.edu.au/~tas110/Teaching/Lectures/L1/Material/WEAVER1947.pdf). *American Scientist* **36** (4): 536–44. [PMID](http://en.wikipedia.org/wiki/PubMed_Identifier) [18882675](http://www.ncbi.nlm.nih.gov/pubmed/18882675). Retrieved 2007-11-21.
  4. [**Jump up ^**](http://en.wikipedia.org/wiki/Complexity#cite_ref-4) Johnson, Steven (2001). *Emergence: the connected lives of ants, brains, cities, and software*. New York: Scribner. p. 46. [ISBN](http://en.wikipedia.org/wiki/International_Standard_Book_Number) [0-684-86875-X](http://en.wikipedia.org/wiki/Special:BookSources/0-684-86875-X).
  5. [**Jump up ^**](http://en.wikipedia.org/wiki/Complexity#cite_ref-5) “Sir James Lighthill and Modern Fluid Mechanics”, by Lokenath Debnath, The University of Texas-Pan American, US, Imperial College Press: [ISBN 978-1-84816-113-9](http://en.wikipedia.org/wiki/Special:BookSources/9781848161139): [ISBN 1-84816-113-1](http://en.wikipedia.org/wiki/Special:BookSources/1848161131), Singapore, page 31. Online at <http://cs5594.userapi.com/u11728334/docs/25eb2e1350a5/Lokenath_Debnath_Sir_James_Lighthill_and_mode.pdf>
  6. [**Jump up ^**](http://en.wikipedia.org/wiki/Complexity#cite_ref-6) Jacobs, Jane (1961). *The Death and Life of Great American Cities*. New York: Random House.
  7. [**Jump up ^**](http://en.wikipedia.org/wiki/Complexity#cite_ref-7) Ulanowicz, Robert, "Ecology, the Ascendant Perspective", Columbia, 1997
  8. [**Jump up ^**](http://en.wikipedia.org/wiki/Complexity#cite_ref-8) Lissack, Michael R.; Johan Roos (2000). The Next Common Sense, The e-Manager's Guide to Mastering Complexity. Intercultural Press. [ISBN 978-1-85788-235-3](http://en.wikipedia.org/wiki/Special:BookSources/9781857882353).
  9. [**Jump up ^**](http://en.wikipedia.org/wiki/Complexity#cite_ref-9) Ho, T.K.; Basu, M. (2002). "[Complexity Measures of Supervised Classification Problems](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=990132&tag=1)". IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (3), pp 289-300.
  10. [**Jump up ^**](http://en.wikipedia.org/wiki/Complexity#cite_ref-10) Smith, M.R.; Martinez, T.; Giraud-Carrier, C. (2014). "[An Instance Level Analysis of Data Complexity](http://link.springer.com/article/10.1007%2Fs10994-013-5422-z)". Machine Learning, 95(2): 225-256.
  11. [**Jump up ^**](http://en.wikipedia.org/wiki/Complexity#cite_ref-11) Saez, J.; Luengo, J.; Herrera, F. (2013). "[Predicting Noise Filtering Efficacy with Data Complexity Measures for Nearest Neighbor Classification](http://www.sciencedirect.com/science/article/pii/S0031320312003251)". Pattern Recognition 46 (1) pp 355-364.
  12. [**Jump up ^**](http://en.wikipedia.org/wiki/Complexity#cite_ref-12) Phys. Chem. Chem. Phys., 2011, 13, 10136–10146
  13. [**Jump up ^**](http://en.wikipedia.org/wiki/Complexity#cite_ref-13) [[1]](http://www.constructal.org/en/art/Phil.%20Trans.%20R.%20Soc.%20B%20%282010%29%20365,%201335%961347.pdf) Bejan A., Lorente S., The Constructal Law of Design and Evolution in Nature. *Philosophical Transactions of the Royal Society B, Biological Science*, Vol. 365, 2010, pp. 1335–1347.
  14. [**Jump up ^**](http://en.wikipedia.org/wiki/Complexity#cite_ref-14) Lorente S., Bejan A. (2010). Few Large and Many Small: Hierarchy in Movement on Earth, *International Journal of Design of Nature and Ecodynamics*, Vol. 5, No. 3, pp. 254–267.
  15. [**Jump up ^**](http://en.wikipedia.org/wiki/Complexity#cite_ref-15) Kim S., Lorente S., Bejan A., Milter W., Morse J. (2008) The Emergence of Vascular Design in Three Dimensions, *Journal of Applied Physics*, Vol. 103, 123511.