Science and computer science

N. F. Stewart [*]
Département d'informatique et de recherche opérationnelle
Université de Montréal

Revised December 12, 1994

1 The nature of computer science

During the few decades that computer science has been identified as a separate discipline, the question of the intellectual nature of the subject has frequently been posed. In particular, the question was discussed recently in the 1993 Turing Award lecture of Juris Hartmanis, "On computational complexity and the nature of computer science" [1].

It is clear that there are components of computer science that could be viewed as subfields of mathematics or engineering. Indeed, competent work in theoretical computer science meets rigorous mathematical standards, and competent work in applied computer science meets the standards of good-quality engineering work (prototypes are built, proof-of-concept projects are conducted, and results are evaluated on the basis of their usefulness in practice). The open question is the extent to which the remaining parts of computer science can be viewed as science. To study this question is not necessarily to engage in sterile debate: it may be, for example, that the field is not yet a science, but that it could become one, and that certain policy changes could accelerate this process.

Hartmanis argues [1, 2] that computer science is different enough from the other sciences to permit different standards in experimental work, and that computer-science "demos" can be viewed as a replacement for the experimentation found in other fields. I do not agree. Computer science is, or has the potential to be, a science similar in character to physics and the other natural sciences. However, its traditions, in the areas of experimentation and formulation of theories, may delay its acceptance and inhibit its development (as a science).

---

[*] This research was supported in part by a grant from the Natural Sciences and Engineering Research Council of Canada. Full address: Département IRO, Université de Montréal, C.P. 6128, Succ. A, Montréal, Québec, Canada H3C 3J7. Electronic mail: stewart@iro.umontreal.ca FAX (514)-343-5834

2 An informal comparison with physics

It is stated in [1] that, firstly, physics and computer science differ due to '... the immense difference in scale of the phenomena computer science deals with", and secondly that in computer science there are no "... new theories developed to reconcile theory with experimental results that reveal unexplained anomalies or new, unexpected phenomena...."

It is not clear in what sense the first of these comparisons is intended, but it seems a difficult thesis to defend. For example, in [3] we are shown photographs of objects of interest in physics, ranging in size from 10^{-16} meters to 10^{25} meters. A ratio of 10^{41} is a very large difference in scale, even for a computer scientist; moreover, physicists do not rule out the study of even larger, or even smaller, objects.

It is also not clear that the second stated difference is fundamental. There are many areas of computer science where we might propose theories that could be decided on the basis of experiment. For example, what we understand as "intelligence" is very poorly understood. It seems reasonable, therefore, to introduce various theories of intelligence, or models of intelligent behaviour, and to test them using the methods and ethics of modern experimental science. (The difficulty here is in defining the problem we want to solve: see Section 4, below.) Similar remarks apply in computer vision. Even in computer graphics, it should be possible, in some contexts, to decide questions of quality by careful definition and careful experimentation [4].

It is also possible, for example, to draw an analogy between theoretical results in computational complexity and theoretical results in thermodynamics: both impose theoretical limits useful on the engineering side of their respective fields.

(Hartmanis also states [1, p. 40] that "the basic, underlying model of digital computing is not seriously challenged by theory or experiments. The ultimate limits of *effective* computing, imposed by the theory of computing, are well understood and accepted." But there was a time when something similar could have been said about Newtonian mechanics, or the Ptolemaic view of the solar system. It is at least *possible*, for example, that so-called "emergent parallelism" could someday show that our currently-accepted models of computing are naively primitive; or that quantum computing [5] or some other as yet unimagined idea could lead to changes in computer science as spectacular as those initiated by Galileo.)

In short, there seem to be many examples suggesting that the usual paradigms of modern science may be appropriate for computer science. It is probably true, as stated in [1], that "in computer science there is no history of critical experiments that decide between the validity of various theories, as there are in the physical sciences." However, this should not be viewed as a justification for continuing as before, but rather, as a possibly fatal flaw to be remedied.

3 What constitutes science?

To decide whether computer science is a science, we must decide what constitutes science. This is an old philosophical question, going back at least to Hume and Kant. Hartmanis refers to the work of Kuhn, who viewed science as a series of periods of "normal science", separated by major revolutions, or paradigm shifts. This view of science is open to debate [6], but clearly it does nothing to detract from the claim that computer science is indeed a science.

One important answer to the question of what constitutes science was given by Popper [8], who referred to the problem as the *problem of demarcation.* He defines this as "... the problem of finding a criterion which would enable us to distinguish between the empirical sciences on the one hand, and mathematics and logic as well as 'metaphysical' systems on the other ..."; his solution was the famous criterion of *falsifiability.* This attempt to define science has had enormous influence, and was used by Popper and others to distinguish between the natural sciences and systems of thought such as Freudian psychology and Marxism [8, 9].

Popper's conditions are considered insufficient, in the field of scientific philosophy, as a definition of science [10], but the idea that scientists should propose "testable hypotheses" is accepted as a basic principle in modern science. In the next section I will outline conditions, suggested by Popper, which seem like obvious *necessary* conditions for a field to be considered a science. I will then discuss to what extent computer science satisfies these conditions.

4 Popper's criteria

According to Popper, "... a scientist, whether theorist or experimenter, puts forward statements, or systems of statements, and tests them step by step...." Science is not a 'body of knowledge', but rather a system of hypotheses, or "... a system of guesses or anticipations which in principle cannot be justified, but with which we work as long as they stand up to tests...." He describes four different approaches to the testing of a theory, the first three logical or theoretical, and the last involving empirical testing [8, pp. 27, 32-33, 317].

The first condition given by Popper is this: even to engage in rational discourse requires that we *state clearly the problem we wish to solve.* Popper writes

> [There is] one method of all *rational discussion*, and therefore of the natural sciences as well as of philosophy ... [namely] ... stating one's problem clearly and ... examining its various proposed solutions *critically.* ... Whenever we propose a solution to a problem, we ought to try as hard as we can to overthrow our solution, rather than defend it. Few of us, unfortunately, practise this precept; but other people, fortunately, will supply the criticism for us if we fail to supply it

> ourselves. Yet criticism will be fruitful only if we state our problem as clearly as we can and put our solution in a sufficiently definite form—a form in which it can be critically discussed....
>
> A severe test of a system presupposes that it is at the time sufficiently definite and final in form to make it impossible for new assumptions to be smuggled in [8, pp. 16, 71].

The second requirement, relating to empirical science, is that experiments should be *repeatable*. This has been a basic principle of empirical science since the seventeenth century [6, p. 147], and Popper takes it almost for granted:

> Only when certain events recur in accordance with rules or regularities, as is the case with repeatable experiments, can our observations be tested—in principle—by anyone. ... No serious physicist would offer for publication, as a scientific discovery, any ... [physical effect] ... for whose reproduction he could give not instructions [8, p. 45].

Indeed, in actual practice, a physicist publishing an experimental result can expect that it will be verified by other physicists, and the verification of an experiment (perhaps with higher accuracy, or by using slightly different methods of measurement) is considered in many areas of physics to be a publishable result.

Finally, a scientific system must, at least in principle, be *falsifiable*. This applies, in particular, to empirical science:

> It must be possible for an empirical scientific system to be refuted by experience [8, p. 41].

This is essentially Popper's definition of science, and he justifies it at length for empirical and non-empirical science; indeed, he applied the same idea to the study of social systems [9]. Harré observes [10] that the idea of privileging negative evidence is an old one, going back to Francis Bacon [6, p. 148]. However, Popper's elaboration of this principle has had great impact, both by direct influence on practising scientists, and indirectly on other philosophers of science, such as Lakatos [11, Ch. 7].

5 Current practice in computer science

For many subfields of computer science, it cannot be said that they meet even Popper's first condition, defining "rational discussion". For example, in computer vision, the problem is frequently left unstated; indeed, there is often no distinction made between the problem to be solved and the algorithm that solves it. An algorithm is proposed, it is applied to real or contrived data, and the results are observed to see whether the algorithm seems to imitate a biological organism in some ill-defined way.

Similar comments can be made about other areas of computer science, such as computer graphics. Of course, there is nothing wrong with doing experimental exploration, especially when a science is in its infancy. (There is evidence that Galileo, in addition to laying the foundations of experimental science by performing experiments to test a hypothesis, also did experiments of this exploratory type [6, p. 141].) But if computer science is to be taken seriously as a science, it will be necessary to move to the next stage (the formulation of well-defined problems, and their examination by theoretical and/or experimental means).

The field is, however, moving in the right direction. In particular, in the two subfields mentioned above, there have been efforts to formalize theories (see for example [12] and [13]). Considering the particular requirement of careful problem definition, the idea that to compare numerical methods in a rigorous way it is essential to define the class of problems to be solved, was advocated over twenty-five years ago [14]. More recently, the introduction of Abstract Data Types can be viewed as a significant step towards putting computer science on a scientific footing, since it directly addresses the question of problem definition. (Indeed, the area of data structures and algorithms is probably the most scientifically well-grounded part of computer science.)

Popper's second condition is that experiments should be repeatable. Here too computer science is far from the best tradition of scientific work. Reading about the results of a putatively scientific experiment in computer science, it is almost certain that there will not be enough information given to permit repetition, and there will therefore be no possibility of refuting the results. One of the reasons for this is the (essentially trivial) difficulty of specifying the exact conditions of the experiment, and of the algorithms implemented. But it may also be due, in part, to a disinclination to make freely available the results of a large implementation.

As with the first condition, there are counterexamples to the remarks just made. For example, Salton's group at Cornell has made the SMART document-retrieval system freely available, so that reported experiments can be verified, and so that other groups can build on what has already been done. Similarly, the fact that the article-evaluation forms for at least two computer graphics conferences include the question "Can an experienced practitioner duplicate this work from the text and references?" is, again, an indication that the field is moving in the right direction.

It would be a good thing if the question just quoted became standard in referee forms in all applied subfields of computer science. Another step towards improving the present situation might be to require, as a matter of course, that published claims about implementation and experimentation be supported by source code available by ftp. A third possibility is to adopt the physics tradition of viewing as a publishable result the confirmation of a reported experiment.

References

[1] Hartmanis, J. On computational complexity and the nature of computer science. *CACM* 37, No. 10, pp. 37-43, October 1994.

[2] Hartmanis, J. Some observations about the nature of computer science. Lecture Notes in Computer Science 761, R. K. Shyamasundar, ed., Springer-Verlag, pp. 1-12, 1993.

[3] Morrison, P. and Morrison, P. Powers of Ten. Scientific American Press, 1994.

[4] Farin, G. and Sapidis, N. Curvature and the fairness of curves and surfaces. *IEEE Computer Graphics and Applications*, pp. 52-57, March 1989.

[5] Brassard, G. Quantum computing: the end of classical cryptography? To appear, Sigact News. Manuscript, Oct. 31, 1994.

[6] Cohen, I. B. Revolution in Science. Harvard University Press, 1985.

[7] Lindberg, D. C. The Beginnings of Western Science. University of Chicago Press, 1992.

[8] Popper, K. J. The Logic of Scientific Discovery. Harper and Row, New York, 1968. (Originally published as: Logik des Forschung, Vienna, 1934.)

[9] Popper, K. J. The Open Society and Its Enemies, Princeton University Press,1966.

[10] Harré, R. Obituary for Professor Sir Karl Popper. The Independent, London, England, September 19, 1994, p. 32.

[11] Davis, P. J. and Hersh, R. The Mathematical Experience. Houghton Mifflin, Boston, 1981.

[12] Tsotsos, J. K. Analyzing vision at the complexity level. *Behavioral and Brain Sciences* (13-3), pp. 423-445, 1990.

[13] Fiume, E. L. The Mathematical Structure of Raster Graphics. Academic Press, 1989.

[14] Hull, T. E. The numerical integration of ordinary differential equations. Proceedings IFIP Congress '68, North-Holland, Amsterdam.