Thus, Nonstandard Analysis provides a 'more computable' framework for calculus and physics. Similar results are available for Bishop's 'constructive' analysis ([1, 6]).

Acknowledgment This research is supported by a grant from the John Templeton Foundation for the project *Philosophical Frontiers in Reverse Mathematics*. The opinions expressed here do not necessarily reflect the views of the John Templeton Foundation.

References

[1] Bishop, E., Bridges, D.S., *Constructive analysis*, Grundlehren der Mathematischen Wissenschaften, volume 279, Springer, Berlin, 1985.

[2] Goldblatt, R. *Lectures on the hyperreals*, Graduate Texts in Mathematics, volume 188, Springer-Verlag, New York, 1998.

[3] Robinson, A., *Non-standard analysis*, North-Holland Publishing Co., Amsterdam, 1966.

[4] Sanders, S., 'ERNA and Friedman's Reverse Mathematics', *J. of Symbolic Logic*, 2011, volume 76.

[5] Sanders, S., 'Reverse mathematics and non-standard analysis; a treasure trove for the philosophy of science', in Mitsuhiro Okado (ed.), *Proceedings of the Ontology and Analytic Metaphysics meeting, Keio University Press*, 2011.

[6] Sanders, S., 'A tale of three Reverse Mathematics', 2011, Submitted.

[7] Soare, Robert I., *Recursively enumerable sets and degrees*, Perspectives in Mathematical Logic, Springer-Verlag, Berlin, 1987.

Is Computer Science Made Scientific by its Experiments?

Viola Schiaffonati
Dipartimento di Elettronica e Informazione
Politecnico di Milano, Italy
schiaffo@elet.polimi.it
Mario Verdicchio
Dipartimento di Ingegneria dell'Informazione
Università degli Studi di Bergamo, Italy
mario.verdicchio@unibg.it

Computer-aided design of bridges and integrated circuits, the long established mapping of computable functions onto recursive functions, and the fundamental role of computer simulations in the verification of models and theories in physics, chemistry, and biology are examples on how ubiquitous the concepts and the instruments of computer science have become in the fields of engineering, mathematics, and natural sciences. Despite such spread and versatility in scientific applications, the scientific status of computer science is still in question.

Computer science has been considered as the science of computers and related phenomena [7], thus supporting the conception of computer science as *artificial science* [9], namely an empirical study of the phenomena connected to computers viewed as artifacts. Contrary to such empirical interpretation, computer science has elsewhere been conceived as the study of theoretical notions, such as information and algorithms [2]. These concepts play a central role also in the work of Peter Denning, who advocates the scientific character of the discipline by shifting the focus from computers to computation and considering the latter as a domain distinguished from and with equal status as physics, society, and biology [3]. As these domains are the subject of physical, social, and life sciences, respectively, so is computation the topic of a specific science, namely, computing. Although agreeing with the author in considering computation as fundamental in his endeavors, we adopt an orthogonal point of view, searching for scientific legitimation not in the content of the discipline, but in its methodology.

If one of the defining criteria of scientific activity is experimental method, according to the empirical sciences' tradition, firstly we need to determine which of the applications of computer science can be legitimately viewed as experimental activity. Undoubtedly, in such analysis we need to take into account the debate on the status and the role of scientific experiments that has been going on in philosophy of science [8]. Then, we intend to understand the contribution of these experiments to the empirical nature of computer science, by using them as means to determine the position of this discipline with respect to the ones traditionally recognized as experimental.

There are two ways to interpret the activities of computer scientists as experiments:

- Programs (e.g. the above-mentioned computer simulations) are used as experiments to provide new data about physical systems, like atoms or galaxies, that are difficult or even impossible to investigate with direct observation [4];
- Iterated computer-based procedures can be used to arbitrate between competing models or hypotheses that are not dealing with a physical or social phenomenon, but with entities that are intrinsically related to computer science, like algorithms and programs.

We consider the former case as out of our scope: it provides an instrumental view of computer science as *infra-science*, according to which the discipline provides new and better instruments for experimentation in the existing sciences. From this perspective, a computer simulation of a galaxy is an up-to-date experiment in astrophysics, but provides no support in analyzing the scientific status of computer science.

Instead, we focus on the latter activities, aimed at working with topics within the scope of computer science, meant as the discipline of automatic computation and information processing. Although in computer science, and computer engineering, the activities called experiments share some common features, there are also impressive differences that make it difficult to have a common definition of experiment. If we consider algorithms, a certain level of rigor in conducting experiments can be evidenced [5], as some principles of experimental methodology, such as *comparison*, *reproducibility*, and *repeatability*, are concretely declined. For instance, in software engineering and software testing in particular, pieces of code are empirically probed following standardized procedures that run them on a set of preselected input values and compare the results with the expected outcome, to search for conditions under which the examined items do not function properly [6]. Still, there exist other subfields of computer science, like autonomous mobile robotics, in which experimentation has not yet reached a comparable level of maturity. Despite the recognized importance of experimental approaches, these ideas have not yet become really part of the current practice, due to the weak awareness of experiments as fundamental elements in the development of a robotic system [1].

The examples above show a certain degree of heterogeneity when speaking of experiments in computer science. Rather than a single experimental method, computer science is characterized by different methodologies for each subfield, where experiments are performed not to confirm a general theory, but to test whether a given system works appropriately. Our working plan is to analyze the production of artifacts in the form of algorithms and programs, and focus on the experimental side of this process, searching for common features in the several subfields of the discipline. We aim at shedding some light on the possibility, despite their significant differences, to individuate general experimental principles in the subfields of computer science in the direction of a more rigorous approach typical of mature scientific disciplines.

Acknowledgement. This work was partially supported by MIUR in the framework of the PRIN Gatecom project.

References

[1] Amigoni, F., Reggiani, M., Schiaffonati, V. (2009). 'An insightful comparison between experiments in mobile robotics and in science', *Autonomous Robots*, Springer, 27(4), pp. 313-325.

[2] Colburn, T. (2004) 'Methodology of Computer Science', in Floridi, L. (ed.), *Philosophy of Computing and Information*, Floridi, L. (ed.), Blackwell, pp. 318-326.

[3] Denning, P. (2009). 'Computing: the Fourth Great Domain of Science', *Communications of the ACM*, 52 (9), pp. 27-29.

[4] Humphreys, P. (2004). *Extending Ourselves. Computational Science, Empiricism, and Scientific Method*, Oxford University Press.

[5] Johnson, D. (2002). 'A Theoretician's Guide to the Experimental Analysis of Algorithms', in Goldwasser, M.H., Johnson, D.S., McGeoch, C.C. (eds.), *Data Structures, Near Neighbor Searches, and Methodology: Fifth and Sixth DIMACS Implementation Challenges*, American Mathematical Society, Providence, pp. 215-250.

[6] Meyers, G. (1979). *The Art of Software Testing*, Wiley & Sons.

[7] Newell, A., Perlis, A., Simon, H. (1967). 'Computer Science', *Science*, 157 (3795), pp. 1373-1374.

[8] Radder, H. (2003). *The Philosophy of Scientific Experimentation*, University of Pittsburgh Press.

[9] Simon, H. (1969). *The Sciences of the Artificial*, MIT Press.

Mechanical brain in the XIX century: Logical machines of Alfred Smee

Valery V. Shilov
MATI - Russian State Technology University
shilov@mati.ru
Vladimir V. Kitov
Institute of History of Natural Sciences and Technics Russian Academy of Science
vladimir.kitov@mail.ru

In the middle of the XIX century the reputed English scientist Alfred Smee (1818-1877) developed the theory, which he called electrobiology. Its purpose was to study the effect of electrical phenomena on the functioning of the human body. Smee was primarily interested in the connection of electrical stimulation of the nervous system and brain work. Later his research in this area became